OpenAI Platform

Introduction

This API reference describes the RESTful, streaming, and realtime APIs you can use to interact with the OpenAI platform. REST APIs are usable via HTTP in any environment that supports HTTP requests. Language-specific SDKs are listed on the libraries page.

Authentication

The OpenAI API uses API keys for authentication. Create, manage, and learn more about API keys in your organization settings.

Remember that your API key is a secret! Do not share it with others or expose it in any client-side code (browsers, apps). API keys should be securely loaded from an environment variable or key management service on the server.

API keys should be provided via HTTP Bearer authentication.

```
Authorization: Bearer OPENAI_API_KEY
```

If you belong to multiple organizations or access projects through a legacy user API key, pass a header to specify which organization and project to use for an API request:

```
curl https://api.openai.com/v1/models \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "OpenAI-Organization: org-j6StrpL3puxPw5BDQ03VRiMw" \
  -H "OpenAI-Project: $PROJECT_ID"
```

Usage from these API requests counts as usage for the specified organization and project.Organization IDs can be found on your organization settings page. Project IDs can be found on your general settings page by selecting the specific project.

Debugging requests

In addition to error codes returned from API responses, you can inspect HTTP response headers containing the unique ID of a particular API request or information about rate limiting applied to your requests. Below is an incomplete list of HTTP headers returned with API responses:

API meta information

- `openai-organization` : The organization associated with the request
- `openai-processing-ms` : Time taken processing your API request
- `openai-version` : REST API version used for this request (currently `2020-10-01`)
- `x-request-id` : Unique identifier for this API request (used in troubleshooting)

Rate limiting information

- `x-ratelimit-limit-requests`
- `x-ratelimit-limit-tokens`
- `x-ratelimit-remaining-requests`
- `x-ratelimit-remaining-tokens`
- `x-ratelimit-reset-requests`
- `x-ratelimit-reset-tokens`

OpenAI recommends logging request IDs in production deployments for more efficient troubleshooting with our support team, should the need arise. Our official SDKs provide a property on top-level response objects containing the value of the `x-request-id` header.

Backward compatibility

OpenAI is committed to providing stability to API users by avoiding breaking changes in major API versions whenever reasonably possible. This includes:

- The REST API (currently `v1`)
- Our first-party SDKs (released SDKs adhere to semantic versioning)
- Model families (like `gpt-4o` or `o4-mini`)

Model prompting behavior between snapshots is subject to change. Model outputs are by their nature variable, so expect changes in prompting and model behavior between snapshots.

For example, if you moved from `gpt-4o-2024-05-13` to `gpt-4o-2024-08-06`, the same `system` or `user` messages could function differently between versions. The best way to ensure consistent prompting behavior and model output is to use pinned model versions, and to implement evals for your applications.

Backwards-compatible API changes:

- Adding new resources (URLs) to the REST API and SDKs
- Adding new optional API parameters
- Adding new properties to JSON response objects or event data
- Changing the order of properties in a JSON response object
- Changing the length or format of opaque strings, like resource identifiers and UUIDs
- Adding new event types (in either streaming or the Realtime API)

See the changelog for a list of backwards-compatible changes and rare breaking changes.

Responses

OpenAI's most advanced interface for generating model responses. Supports text and image inputs, and text outputs. Create stateful interactions with the model, using the output of previous responses as input. Extend the model's capabilities with built-in tools for file search, web search, computer use, and more. Allow the model access to external systems and data using function calling.

Related guides:

- Quickstart
- Text inputs and outputs
- Image inputs
- Structured Outputs
- Function calling
- Conversation state
- Extend the models with tools

Create a model response

`POST https://api.openai.com/v1/responses`

Creates a model response. Provide text or image inputs to generate text or JSON outputs. Have the model call your own custom code or use built-in tools like web search or file search to use your own data as input for the model's response.

Request body

background boolean or null Optional Defaults to false
Whether to run the model response in the background. Learn more.

include array or null Optional
Specify additional output data to include in the model response. Currently supported values are:

- `code_interpreter_call.outputs` : Includes the outputs of python code execution in code interpreter tool call items.
- `computer_call_output.output.image_url` : Include image urls from the computer call output.
- `file_search_call.results` : Include the search results of the file search tool call.
- `message.input_image.image_url` : Include image urls from the input message.
- `message.output_text.logprobs` : Include logprobs with assistant messages.
- `reasoning.encrypted_content` : Includes an encrypted version of reasoning tokens in reasoning item outputs. This enables reasoning items to be used in multi-turn conversations when using the Responses API statelessly (like when the `store` parameter is set to `false` , or when an organization is enrolled in the zero data retention program).

input string or array Optional
Text, image, or file inputs to the model, used to generate a response.

Learn more:

- Text inputs and outputs
- Image inputs
- File inputs
- Conversation state
- Function calling

^ Hide possible types

Text input string
A text input to the model, equivalent to a text input with the `user` role.

Input item list array
A list of one or many input items to the model, containing different content types.

Hide possible types

Input message object
A message input to the model with a role indicating instruction following hierarchy. Instructions given with the `developer` or `system` role take precedence over instructions given with the `user` role. Messages with the `assistant` role are presumed to have been generated by the model in previous interactions.

Hide properties

content string or array Required
Text, image, or audio input to the model, used to generate a response. Can also contain previous assistant responses.

Hide possible types

Text input string
A text input to the model.

Input item content list array
A list of one or many input items to the model, containing different content types.

Hide possible types

Input text object
A text input to the model.

Hide properties

text string Required
The text input to the model.

type string Required
The type of the input item. Always `input_text` .

Input image object
An image input to the model. Learn about image inputs.

Hide properties

detail string Required
The detail level of the image to be sent to the model. One of `high` , `low` , or `auto` .

Defaults to `auto`.

`type` string Required
The type of the input item. Always `input_image`.

`file_id` string Optional
The ID of the file to be sent to the model.

`image_url` string Optional
The URL of the image to be sent to the model. A fully qualified URL or base64 encoded image in a data URL.

Input file object
A file input to the model.

Hide properties

`type` string Required
The type of the input item. Always `input_file`.

`file_data` string Optional
The content of the file to be sent to the model.

`file_id` string Optional
The ID of the file to be sent to the model.

`file_url` string Optional
The URL of the file to be sent to the model.

`filename` string Optional
The name of the file to be sent to the model.

`role` string Required
The role of the message input. One of `user`, `assistant`, `system`, or `developer`.

`type` string Optional
The type of the message input. Always `message`.

Item object
An item representing part of the context for the response to be generated by the model. Can contain text, images, and audio inputs, as well as previous assistant responses and tool call outputs.

Hide possible types

Input message object

A message input to the model with a role indicating instruction following hierarchy. Instructions given with the `developer` or `system` role take precedence over instructions given with the `user` role.

Hide properties

content array Required

A list of one or many input items to the model, containing different content types.

Hide possible types

Input text object

A text input to the model.

Hide properties

text string Required

The text input to the model.

type string Required

The type of the input item. Always `input_text`.

Input image object

An image input to the model. Learn about image inputs.

Hide properties

detail string Required

The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

type string Required

The type of the input item. Always `input_image`.

file_id string Optional

The ID of the file to be sent to the model.

image_url string Optional

The URL of the image to be sent to the model. A fully qualified URL or base64 encoded image in a data URL.

Input file object

A file input to the model.

Hide properties

type string Required
The type of the input item. Always `input_file`.

file_data string Optional
The content of the file to be sent to the model.

file_id string Optional
The ID of the file to be sent to the model.

file_url string Optional
The URL of the file to be sent to the model.

filename string Optional
The name of the file to be sent to the model.

role string Required
The role of the message input. One of `user`, `system`, or `developer`.

status string Optional
The status of item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

type string Optional
The type of the message input. Always set to `message`.

Output message object
An output message from the model.

Hide properties

content array Required
The content of the output message.

Hide possible types

Output text object
A text output from the model.

Hide properties

annotations array Required
The annotations of the text output.

Hide possible types

File citation object

A citation to a file.

Hide properties

file_id string Required

The ID of the file.

filename string Required

The filename of the file cited.

index integer Required

The index of the file in the list of files.

type string Required

The type of the file citation. Always `file_citation` .

URL citation object

A citation for a web resource used to generate a model response.

Hide properties

end_index integer Required

The index of the last character of the URL citation in the message.

start_index integer Required

The index of the first character of the URL citation in the message.

title string Required

The title of the web resource.

type string Required

The type of the URL citation. Always `url_citation` .

url string Required

The URL of the web resource.

Container file citation object

A citation for a container file used to generate a model response.

Hide properties

container_id string Required

The ID of the container file.

end_index integer Required
The index of the last character of the container file citation in the message.

file_id string Required
The ID of the file.

filename string Required
The filename of the container file cited.

start_index integer Required
The index of the first character of the container file citation in the message.

type string Required
The type of the container file citation. Always `container_file_citation`.

File path object
A path to a file.

Hide properties

file_id string Required
The ID of the file.

index integer Required
The index of the file in the list of files.

type string Required
The type of the file path. Always `file_path`.

text string Required
The text output from the model.

type string Required
The type of the output text. Always `output_text`.

logprobs array Optional

Hide properties

bytes array Required

logprob number Required

token string Required

top_logprobs array Required

Hide properties

bytes array Required

logprob number Required

token string Required

Refusal object

A refusal from the model.

Hide properties

refusal string Required

The refusal explanationfrom the model.

type string Required

The type of the refusal. Always `refusal`.

id string Required

The unique ID of the output message.

role string Required

The role of the output message. Always `assistant`.

status string Required

The status of the message input. One of `in_progress`, `completed`, or `incomplete`. Populated when input items are returned via API.

type string Required

The type of the output message. Always `message`.

File search tool call object

The results of a file search tool call. See the file search guide for more information.

Hide properties

id string Required
The unique ID of the file search tool call.

queries array Required
The queries used to search for files.

status string Required
The status of the file search tool call. One of `in_progress`, `searching`, `incomplete` or `failed`,

type string Required
The type of the file search tool call. Always `file_search_call`.

results array or null Optional
The results of the file search tool call.

Hide properties

attributes map Optional
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard. Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters, booleans, or numbers.

file_id string Optional
The unique ID of the file.

filename string Optional
The name of the file.

score number Optional
The relevance score of the file - a value between 0 and 1.

text string Optional
The text that was retrieved from the file.

Computer tool call object
A tool call to a computer use tool. See the computer use guide for more information.

Hide properties

action object Required

Hide possible types

Click object

A click action.

Hide properties

button string Required

Indicates which mouse button was pressed during the click. One of `left`, `right`, `wheel`, `back`, or `forward`.

type string Required

Specifies the event type. For a click action, this property is always set to `click`.

x integer Required

The x-coordinate where the click occurred.

y integer Required

The y-coordinate where the click occurred.

DoubleClick object

A double click action.

Hide properties

type string Required

Specifies the event type. For a double click action, this property is always set to `double_click`.

x integer Required

The x-coordinate where the double click occurred.

y integer Required

The y-coordinate where the double click occurred.

Drag object

A drag action.

Hide properties

path array Required

An array of coordinates representing the path of the drag action. Coordinates will appear as an array of objects, eg

```
[
  { x: 100, y: 200 },
  { x: 200, y: 300 }
]
```

Hide properties

x integer Required
The x-coordinate.

y integer Required
The y-coordinate.

type string Required
Specifies the event type. For a drag action, this property is always set to `drag`.

KeyPress object
A collection of keypresses the model would like to perform.

Hide properties

keys array Required
The combination of keys the model is requesting to be pressed. This is an array of strings, each representing a key.

type string Required
Specifies the event type. For a keypress action, this property is always set to `keypress`.

Move object

A mouse move action.

Hide properties

type string Required

Specifies the event type. For a move action, this property is always set to `move`.

x integer Required

The x-coordinate to move to.

y integer Required

The y-coordinate to move to.

Screenshot object

A screenshot action.

Hide properties

type string Required

Specifies the event type. For a screenshot action, this property is always set to `screenshot`.

Scroll object

A scroll action.

Hide properties

scroll_x integer Required

The horizontal scroll distance.

scroll_y integer Required

The vertical scroll distance.

type string Required

Specifies the event type. For a scroll action, this property is always set to `scroll`.

x integer Required

The x-coordinate where the scroll occurred.

y integer Required

The y-coordinate where the scroll occurred.

Type object

An action to type in text.

Hide properties

text string Required

The text to type.

type string Required

Specifies the event type. For a type action, this property is always set to `type`.

Wait object

A wait action.

Hide properties

type string Required

Specifies the event type. For a wait action, this property is always set to `wait`.

call_id string Required

An identifier used when responding to the tool call with output.

id string Required

The unique ID of the computer call.

pending_safety_checks array Required

The pending safety checks for the computer call.

Hide properties

code string Required

The type of the pending safety check.

id string Required

The ID of the pending safety check.

message string Required

Details about the pending safety check.

status string Required

The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

type string Required
The type of the computer call. Always `computer_call`.

Computer tool call output object
The output of a computer tool call.

Hide properties

call_id string Required
The ID of the computer tool call that produced the output.

output object Required
A computer screenshot image used with the computer use tool.

Hide properties

type string Required
Specifies the event type. For a computer screenshot, this property is always set to `computer_screenshot`.

file_id string Optional
The identifier of an uploaded file that contains the screenshot.

image_url string Optional
The URL of the screenshot image.

type string Required
The type of the computer tool call output. Always `computer_call_output`.

acknowledged_safety_checks array Optional
The safety checks reported by the API that have been acknowledged by the developer.

Hide properties

id string Required
The ID of the pending safety check.

code string Optional
The type of the pending safety check.

message string Optional
Details about the pending safety check.

id string Optional
The ID of the computer tool call output.

status string Optional
The status of the message input. One of `in_progress`, `completed`, or `incomplete`. Populated when input items are returned via API.

Web search tool call object
The results of a web search tool call. See the web search guide for more information.

Hide properties

action object Required
An object describing the specific action taken in this web search call. Includes details on how the model used the web (search, open_page, find).

Hide possible types

Search action object
Action type "search" - Performs a web search query.

Hide properties

query string Required
The search query.

type string Required
The action type.

Open page action object
Action type "open_page" - Opens a specific URL from search results.

Hide properties

type string Required
The action type.

url string Required
The URL opened by the model.

Find action object
Action type "find": Searches for a pattern within a loaded page.

Hide properties

pattern string Required

The pattern or text to search for within the page.

type string Required
The action type.

url string Required
The URL of the page searched for the pattern.

id string Required
The unique ID of the web search tool call.

status string Required
The status of the web search tool call.

type string Required
The type of the web search tool call. Always `web_search_call`.

Function tool call object
A tool call to run a function. See the function calling guide for more information.

Hide properties

arguments string Required
A JSON string of the arguments to pass to the function.

call_id string Required
The unique ID of the function tool call generated by the model.

name string Required
The name of the function to run.

type string Required
The type of the function tool call. Always `function_call`.

id string Optional
The unique ID of the function tool call.

status string Optional
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Function tool call output object

The output of a function tool call.

^ Hide properties

call_id string Required

The unique ID of the function tool call generated by the model.

output string Required

A JSON string of the output of the function tool call.

type string Required

The type of the function tool call output. Always `function_call_output`.

id string Optional

The unique ID of the function tool call output. Populated when this item is returned via API.

status string Optional

The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Reasoning object

A description of the chain of thought used by a reasoning model while generating a response. Be sure to include these items in your `input` to the Responses API for subsequent turns of a conversation if you are manually managing context.

^ Hide properties

id string Required

The unique identifier of the reasoning content.

summary array Required

Reasoning text contents.

^ Hide properties

text string Required

A short summary of the reasoning used by the model when generating the response.

type string Required

The type of the object. Always `summary_text`.

type string Required
The type of the object. Always `reasoning`.

encrypted_content string or null Optional
The encrypted content of the reasoning item - populated when a response is generated with `reasoning.encrypted_content` in the `include` parameter.

status string Optional
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Image generation call object
An image generation request made by the model.

Hide properties

id string Required
The unique ID of the image generation call.

result string or null Required
The generated image encoded in base64.

status string Required
The status of the image generation call.

type string Required
The type of the image generation call. Always `image_generation_call`.

Code interpreter tool call object
A tool call to run code.

Hide properties

code string or null Required
The code to run, or null if not available.

container_id string Required
The ID of the container used to run the code.

id string Required
The unique ID of the code interpreter tool call.

outputs array or null Required

The outputs generated by the code interpreter, such as logs or images. Can be null if no outputs are available.

Hide possible types

Code interpreter output logs object
The logs output from the code interpreter.

Hide properties

logs string Required
The logs output from the code interpreter.

type string Required
The type of the output. Always 'logs'.

Code interpreter output image object
The image output from the code interpreter.

Hide properties

type string Required
The type of the output. Always 'image'.

url string Required
The URL of the image output from the code interpreter.

status string Required
The status of the code interpreter tool call.

type string Required
The type of the code interpreter tool call. Always `code_interpreter_call`.

Local shell call object
A tool call to run a command on the local shell.

Hide properties

action object Required
Execute a shell command on the server.

Hide properties

command array Required
The command to run.

env object Required
Environment variables to set for the command.

type string Required
The type of the local shell action. Always `exec`.

timeout_ms integer or null Optional
Optional timeout in milliseconds for the command.

user string or null Optional
Optional user to run the command as.

working_directory string or null Optional
Optional working directory to run the command in.

call_id string Required
The unique ID of the local shell tool call generated by the model.

id string Required
The unique ID of the local shell call.

status string Required
The status of the local shell call.

type string Required
The type of the local shell call. Always `local_shell_call`.

Local shell call output object
The output of a local shell tool call.

^ Hide properties

id string Required
The unique ID of the local shell tool call generated by the model.

output string Required
A JSON string of the output of the local shell tool call.

type string Required
The type of the local shell tool call output. Always `local_shell_call_output`.

status string or null Optional

The status of the item. One of `in_progress`, `completed`, or `incomplete`.

MCP list tools object

A list of tools available on an MCP server.

Hide properties

id string Required

The unique ID of the list.

server_label string Required

The label of the MCP server.

tools array Required

The tools available on the server.

Hide properties

input_schema object Required

The JSON schema describing the tool's input.

name string Required

The name of the tool.

annotations object or null Optional

Additional annotations about the tool.

description string or null Optional

The description of the tool.

type string Required

The type of the item. Always `mcp_list_tools`.

error string or null Optional

Error message if the server could not list tools.

MCP approval request object

A request for human approval of a tool invocation.

Hide properties

arguments string Required

A JSON string of arguments for the tool.

id string Required
The unique ID of the approval request.

name string Required
The name of the tool to run.

server_label string Required
The label of the MCP server making the request.

type string Required
The type of the item. Always `mcp_approval_request` .

MCP approval response object
A response to an MCP approval request.

Hide properties

approval_request_id string Required
The ID of the approval request being answered.

approve boolean Required
Whether the request was approved.

type string Required
The type of the item. Always `mcp_approval_response` .

id string or null Optional
The unique ID of the approval response

reason string or null Optional
Optional reason for the decision.

MCP tool call object
An invocation of a tool on an MCP server.

Hide properties

arguments string Required
A JSON string of the arguments passed to the tool.

id string Required
The unique ID of the tool call.

name string Required
The name of the tool that was run.

server_label string Required
The label of the MCP server running the tool.

type string Required
The type of the item. Always `mcp_call`.

error string or null Optional
The error from the tool call, if any.

output string or null Optional
The output from the tool call.

Item reference object
An internal identifier for an item to reference.

Hide properties

id string Required
The ID of the item to reference.

type string Optional Defaults to item_reference
The type of item to reference. Always `item_reference`.

instructions string or null Optional
A system (or developer) message inserted into the model's context.

When using along with `previous_response_id`, the instructions from a previous response will not be carried over to the next response. This makes it simple to swap out system (or developer) messages in new responses.

max_output_tokens integer or null Optional
An upper bound for the number of tokens that can be generated for a response, including visible output tokens and reasoning tokens.

max_tool_calls integer or null Optional
The maximum number of total calls to built-in tools that can be processed in a response. This maximum number applies across all built-in tool calls, not per individual tool. Any further attempts to call a tool by the

model will be ignored.

metadata map Optional
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

model string Optional
Model ID used to generate the response, like `gpt-4o` or `o3`. OpenAI offers a wide range of models with different capabilities, performance characteristics, and price points. Refer to the model guide to browse and compare available models.

parallel_tool_calls boolean or null Optional Defaults to true
Whether to allow the model to run tool calls in parallel.

previous_response_id string or null Optional
The unique ID of the previous response to the model. Use this to create multi-turn conversations. Learn more about conversation state.

prompt object or null Optional
Reference to a prompt template and its variables. Learn more.

Hide properties

id string Required
The unique identifier of the prompt template to use.

variables map Optional
Optional map of values to substitute in for variables in your prompt. The substitution values can either be strings, or other Response input types like images or files.

version string or null Optional
Optional version of the prompt template.

reasoning object or null Optional
o-series models only

Configuration options for reasoning models.

Hide properties

effort string or null Optional Defaults to medium

o-series models only

Constrains effort on reasoning for reasoning models. Currently supported values are `low`, `medium`, and `high`. Reducing reasoning effort can result in faster responses and fewer tokens used on reasoning in a response.

`generate_summary` Deprecated string or null Optional
Deprecated: use `summary` instead.

A summary of the reasoning performed by the model. This can be useful for debugging and understanding the model's reasoning process. One of `auto`, `concise`, or `detailed`.

`summary` string or null Optional
A summary of the reasoning performed by the model. This can be useful for debugging and understanding the model's reasoning process. One of `auto`, `concise`, or `detailed`.

`service_tier` string or null Optional Defaults to auto
Specifies the processing type used for serving the request.

> If set to 'auto', then the request will be processed with the service tier configured in the Project settings. Unless otherwise configured, the Project will use 'default'.
>
> If set to 'default', then the requset will be processed with the standard pricing and performance for the selected model.
>
> If set to 'flex' or 'priority', then the request will be processed with the corresponding service tier. Contact sales to learn more about Priority processing.
>
> When not set, the default behavior is 'auto'.

When the `service_tier` parameter is set, the response body will include the `service_tier` value based on the processing mode actually used to serve the request. This response value may be different from the value set in the parameter.

`store` boolean or null Optional Defaults to true
Whether to store the generated model response for later retrieval via API.

`stream` boolean or null Optional Defaults to false
If set to true, the model response data will be streamed to the client as it is generated using server-sent events. See the Streaming section below for more information.

`temperature` number or null Optional Defaults to 1
What sampling temperature to use, between 0 and 2. Higher values like 0.8 will make the output more random, while lower values like 0.2 will make it more focused and deterministic. We generally recommend altering this or `top_p` but not both.

text object Optional

Configuration options for a text response from the model. Can be plain text or structured JSON data. Learn more:

Text inputs and outputs

Structured Outputs

Hide properties

format object Optional

An object specifying the format that the model must output.

Configuring `{ "type": "json_schema" }` enables Structured Outputs, which ensures the model will match your supplied JSON schema. Learn more in the Structured Outputs guide.

The default format is `{ "type": "text" }` with no additional options.

Not recommended for gpt-4o and newer models:

Setting to `{ "type": "json_object" }` enables the older JSON mode, which ensures the message the model generates is valid JSON. Using `json_schema` is preferred for models that support it.

Hide possible types

Text object

Default response format. Used to generate text responses.

Hide properties

type string Required

The type of response format being defined. Always `text`.

JSON schema object

JSON Schema response format. Used to generate structured JSON responses. Learn more about Structured Outputs.

Hide properties

name string Required

The name of the response format. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

schema object Required

The schema for the response format, described as a JSON Schema object. Learn how to build JSON schemas here.

type string Required

The type of response format being defined. Always `json_schema`.

description string Optional
A description of what the response format is for, used by the model to determine how to respond in the format.

strict boolean or null Optional Defaults to false
Whether to enable strict schema adherence when generating the output. If set to true, the model will always follow the exact schema defined in the `schema` field. Only a subset of JSON Schema is supported when `strict` is `true`. To learn more, read the Structured Outputs guide.

JSON object object
JSON object response format. An older method of generating JSON responses. Using `json_schema` is recommended for models that support it. Note that the model will not generate JSON without a system or user message instructing it to do so.

Hide properties

type string Required
The type of response format being defined. Always `json_object`.

tool_choice string or object Optional
How the model should select which tool (or tools) to use when generating a response. See the `tools` parameter to see how to specify which tools the model can call.

Hide possible types

Tool choice mode string
Controls which (if any) tool is called by the model.

`none` means the model will not call any tool and instead generates a message.

`auto` means the model can pick between generating a message or calling one or more tools.

`required` means the model must call one or more tools.

Hosted tool object
Indicates that the model should use a built-in tool to generate a response. Learn more about built-in tools.

Hide properties

type string Required
The type of hosted tool the model should to use. Learn more about built-in tools.

Allowed values are:

`file_search`

`web_search_preview`

`computer_use_preview`

`code_interpreter`

`image_generation`

Function tool object

Use this option to force the model to call a specific function.

Hide properties

name string Required

The name of the function to call.

type string Required

For function calling, the type is always `function`.

MCP tool object

Use this option to force the model to call a specific tool on a remote MCP server.

Hide properties

server_label string Required

The label of the MCP server to use.

type string Required

For MCP tools, the type is always `mcp`.

name string or null Optional

The name of the tool to call on the server.

tools array Optional

An array of tools the model may call while generating a response. You can specify which tool to use by setting the `tool_choice` parameter.

The two categories of tools you can provide the model are:

Built-in tools: Tools that are provided by OpenAI that extend the model's capabilities, like web search or file search. Learn more about built-in tools.

Function calls (custom tools): Functions that are defined by you, enabling the model to call your own code. Learn more about function calling.

Hide possible types

Function object

Defines a function in your own code the model can choose to call. Learn more about function calling.

Hide properties

name string Required

The name of the function to call.

parameters object Required

A JSON schema object describing the parameters of the function.

strict boolean Required

Whether to enforce strict parameter validation. Default `true`.

type string Required

The type of the function tool. Always `function`.

description string Optional

A description of the function. Used by the model to determine whether or not to call the function.

File search object

A tool that searches for relevant content from uploaded files. Learn more about the file search tool.

Hide properties

type string Required

The type of the file search tool. Always `file_search`.

vector_store_ids array Required

The IDs of the vector stores to search.

filters object Optional

A filter to apply.

Hide possible types

Comparison Filter object

A filter used to compare a specified attribute key to a given value using a defined comparison operation.

Hide properties

key string Required

The key to compare against the value.

type string Required

Specifies the comparison operator: `eq`, `ne`, `gt`, `gte`, `lt`, `lte`.

- `eq`: equals
- `ne`: not equal
- `gt`: greater than
- `gte`: greater than or equal
- `lt`: less than
- `lte`: less than or equal

`value` string / number / boolean Required

The value to compare against the attribute key; supports string, number, or boolean types.

Compound Filter object

Combine multiple filters using `and` or `or`.

Hide properties

`filters` array Required

Array of filters to combine. Items can be `ComparisonFilter` or `CompoundFilter`.

Hide possible types

Comparison Filter object

A filter used to compare a specified attribute key to a given value using a defined comparison operation.

Hide properties

`key` string Required

The key to compare against the value.

`type` string Required

Specifies the comparison operator: `eq`, `ne`, `gt`, `gte`, `lt`, `lte`.

- `eq`: equals
- `ne`: not equal
- `gt`: greater than
- `gte`: greater than or equal
- `lt`: less than
- `lte`: less than or equal

value string / number / boolean Required
The value to compare against the attribute key; supports string, number, or boolean types.

type string Required
Type of operation: `and` or `or`.

max_num_results integer Optional
The maximum number of results to return. This number should be between 1 and 50 inclusive.

ranking_options object Optional
Ranking options for search.

Hide properties

ranker string Optional
The ranker to use for the file search.

score_threshold number Optional
The score threshold for the file search, a number between 0 and 1. Numbers closer to 1 will attempt to return only the most relevant results, but may return fewer results.

Web search preview object
This tool searches the web for relevant results to use in a response. Learn more about the web search tool.

Hide properties

type string Required
The type of the web search tool. One of `web_search_preview` or `web_search_preview_2025_03_11`.

search_context_size string Optional
High level guidance for the amount of context window space to use for the search. One of `low`, `medium`, or `high`. `medium` is the default.

user_location object Optional
The user's location.

Hide properties

type string Required
The type of location approximation. Always `approximate`.

city string Optional
Free text input for the city of the user, e.g. `San Francisco`.

country string Optional
The two-letter ISO country code of the user, e.g. `US`.

region string Optional
Free text input for the region of the user, e.g. `California`.

timezone string Optional
The IANA timezone of the user, e.g. `America/Los_Angeles`.

Computer use preview object
A tool that controls a virtual computer. Learn more about the computer tool.

Hide properties

display_height integer Required
The height of the computer display.

display_width integer Required
The width of the computer display.

environment string Required
The type of computer environment to control.

type string Required
The type of the computer use tool. Always `computer_use_preview`.

MCP tool object
Give the model access to additional tools via remote Model Context Protocol (MCP) servers. Learn more about MCP.

Hide properties

server_label string Required
A label for this MCP server, used to identify it in tool calls.

server_url string Required

The URL for the MCP server.

type string Required
The type of the MCP tool. Always `mcp`.

allowed_tools array or object Optional
List of allowed tool names or a filter object.

Hide possible types

MCP allowed tools array
A string array of allowed tool names

MCP allowed tools filter object
A filter object to specify which tools are allowed.

Hide properties

tool_names array Optional
List of allowed tool names.

headers object or null Optional
Optional HTTP headers to send to the MCP server. Use for authentication or other purposes.

require_approval object or string Optional Defaults to always
Specify which of the MCP server's tools require approval.

Hide possible types

MCP tool approval filter object

Hide properties

always object Optional
A list of tools that always require approval.

Hide properties

tool_names array Optional
List of tools that require approval.

never object Optional
A list of tools that never require approval.

Hide properties

tool_names array Optional

List of tools that do not require approval.

MCP tool approval setting string
Specify a single approval policy for all tools. One of `always` or `never`. When set to `always`, all tools will require approval. When set to `never`, all tools will require approval.

`server_description` string Optional
Optional description of the MCP server, used to provide more context.

Code interpreter object
A tool that runs Python code to help generate a response to a prompt.

Hide properties

`container` string or object Required
The code interpreter container. Can be a container ID or an object that specifies uploaded file IDs to make available to your code.

Hide possible types

string
The container ID.

CodeInterpreterContainerAuto object
Configuration for a code interpreter container. Optionally specify the IDs of the files to run the code on.

Hide properties

`type` string Required
Always `auto`.

`file_ids` array Optional
An optional list of uploaded files to make available to your code.

`type` string Required
The type of the code interpreter tool. Always `code_interpreter`.

Image generation tool object
A tool that generates images using a model like `gpt-image-1`.

Hide properties

type string Required
The type of the image generation tool. Always `image_generation`.

background string Optional Defaults to auto
Background type for the generated image. One of `transparent`, `opaque`, or `auto`. Default: `auto`.

input_image_mask object Optional
Optional mask for inpainting. Contains `image_url` (string, optional) and `file_id` (string, optional).

Hide properties

file_id string Optional
File ID for the mask image.

image_url string Optional
Base64-encoded mask image.

model string Optional Defaults to gpt-image-1
The image generation model to use. Default: `gpt-image-1`.

moderation string Optional Defaults to auto
Moderation level for the generated image. Default: `auto`.

output_compression integer Optional Defaults to 100
Compression level for the output image. Default: 100.

output_format string Optional Defaults to png
The output format of the generated image. One of `png`, `webp`, or `jpeg`. Default: `png`.

partial_images integer Optional Defaults to 0
Number of partial images to generate in streaming mode, from 0 (default value) to 3.

quality string Optional Defaults to auto
The quality of the generated image. One of `low`, `medium`, `high`, or `auto`. Default: `auto`.

size string Optional Defaults to auto
The size of the generated image. One of `1024x1024`, `1024x1536`, `1536x1024`, or `auto`. Default: `auto`.

Local shell tool object

A tool that allows the model to execute shell commands in a local environment.

Hide properties

type string Required
The type of the local shell tool. Always `local_shell`.

top_logprobs integer or null Optional
An integer between 0 and 20 specifying the number of most likely tokens to return at each token position, each with an associated log probability.

top_p number or null Optional Defaults to 1
An alternative to sampling with temperature, called nucleus sampling, where the model considers the results of the tokens with top_p probability mass. So 0.1 means only the tokens comprising the top 10% probability mass are considered.

We generally recommend altering this or `temperature` but not both.

truncation string or null Optional Defaults to disabled
The truncation strategy to use for the model response.

`auto`: If the context of this response and previous ones exceeds the model's context window size, the model will truncate the response to fit the context window by dropping input items in the middle of the conversation.

`disabled` (default): If a model response will exceed the context window size for a model, the request will fail with a 400 error.

user string Optional
A stable identifier for your end-users. Used to boost cache hit rates by better bucketing similar requests and to help OpenAI detect and prevent abuse. Learn more.

Returns

Returns a Response object.

Text input | Image input | File input | Web search | File search | Streaming | Functions | R

Example request curl

```
curl https://api.openai.com/v1/responses \
```

```
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -d '{
5     "model": "gpt-4.1",
6     "input": "Tell me a three sentence bedtime story about a unicorn."
7   }'
```

Response

```
1  {
2    "id": "resp_67ccd2bed1ec8190b14f964abc0542670bb6a6b452d3795b",
3    "object": "response",
4    "created_at": 1741476542,
5    "status": "completed",
6    "error": null,
7    "incomplete_details": null,
8    "instructions": null,
9    "max_output_tokens": null,
10   "model": "gpt-4.1-2025-04-14",
11   "output": [
12     {
13       "type": "message",
14       "id": "msg_67ccd2bf17f0819081ff3bb2cf6508e60bb6a6b452d3795b",
15       "status": "completed",
16       "role": "assistant",
17       "content": [
18         {
19           "type": "output_text",
20           "text": "In a peaceful grove beneath a silver moon, a unicorn named Lumina disco
21           "annotations": []
22         }
23       ]
24     }
25   ],
26   "parallel_tool_calls": true,
27   "previous_response_id": null,
28   "reasoning": {
29     "effort": null,
30     "summary": null
31   },
32   "store": true,
33   "temperature": 1.0,
34   "text": {
35     "format": {
36       "type": "text"
37     }
38
```

```
  },
  "tool_choice": "auto",
  "tools": [],
  "top_p": 1.0,
  "truncation": "disabled",
  "usage": {
    "input_tokens": 36,
    "input_tokens_details": {
      "cached_tokens": 0
    },
    "output_tokens": 87,
    "output_tokens_details": {
      "reasoning_tokens": 0
    },
    "total_tokens": 123
  },
  "user": null,
  "metadata": {}
}
```

Get a model response

`GET https://api.openai.com/v1/responses/{response_id}`

Retrieves a model response with the given ID.

Path parameters

`response_id` string Required
The ID of the response to retrieve.

Query parameters

`include` array Optional
Additional fields to include in the response. See the `include` parameter for Response creation above for more information.

`starting_after` integer Optional
The sequence number of the event after which to start streaming.

stream boolean Optional

If set to true, the model response data will be streamed to the client as it is generated using server-sent events. See the Streaming section below for more information.

Returns

The Response object matching the specified ID.

Example request curl

```curl
curl https://api.openai.com/v1/responses/resp_123 \
    -H "Content-Type: application/json" \
    -H "Authorization: Bearer $OPENAI_API_KEY"
```

Response

```
{
  "id": "resp_67cb71b351908190a308f3859487620d06981a8637e6bc44",
  "object": "response",
  "created_at": 1741386163,
  "status": "completed",
  "error": null,
  "incomplete_details": null,
  "instructions": null,
  "max_output_tokens": null,
  "model": "gpt-4o-2024-08-06",
  "output": [
    {
      "type": "message",
      "id": "msg_67cb71b3c2b0819084d481baaaf148f206981a8637e6bc44",
      "status": "completed",
      "role": "assistant",
      "content": [
        {
          "type": "output_text",
          "text": "Silent circuits hum,  \nThoughts emerge in data streams—  \nDigital daw
          "annotations": []
        }
      ]
    }
  ],
  "parallel_tool_calls": true,

```

```
  "previous_response_id": null,
  "reasoning": {
    "effort": null,
    "summary": null
  },
  "store": true,
  "temperature": 1.0,
  "text": {
    "format": {
      "type": "text"
    }
  },
  "tool_choice": "auto",
  "tools": [],
  "top_p": 1.0,
  "truncation": "disabled",
  "usage": {
    "input_tokens": 32,
    "input_tokens_details": {
      "cached_tokens": 0
    },
    "output_tokens": 18,
    "output_tokens_details": {
      "reasoning_tokens": 0
    },
    "total_tokens": 50
  },
  "user": null,
  "metadata": {}
}
```

Delete a model response

```
DELETE https://api.openai.com/v1/responses/{response_id}
```

Deletes a model response with the given ID.

Path parameters

response_id string Required

The ID of the response to delete.

Returns

A success message.

Example request curl

```
curl -X DELETE https://api.openai.com/v1/responses/resp_123 \
    -H "Content-Type: application/json" \
    -H "Authorization: Bearer $OPENAI_API_KEY"
```

Response

```
{
  "id": "resp_6786a1bec27481909a17d673315b29f6",
  "object": "response",
  "deleted": true
}
```

Cancel a response

`POST https://api.openai.com/v1/responses/{response_id}/cancel`

Cancels a model response with the given ID. Only responses created with the `background` parameter set to `true` can be cancelled. Learn more.

Path parameters

`response_id` string Required
The ID of the response to cancel.

Returns

A Response object.

Example request curl

```
curl -X POST https://api.openai.com/v1/responses/resp_123/cancel \
```

```
    -H "Content-Type: application/json" \
    -H "Authorization: Bearer $OPENAI_API_KEY"
```

Response


```
{
  "id": "resp_67cb71b351908190a308f3859487620d06981a8637e6bc44",
  "object": "response",
  "created_at": 1741386163,
  "status": "completed",
  "error": null,
  "incomplete_details": null,
  "instructions": null,
  "max_output_tokens": null,
  "model": "gpt-4o-2024-08-06",
  "output": [
    {
      "type": "message",
      "id": "msg_67cb71b3c2b0819084d481baaaf148f206981a8637e6bc44",
      "status": "completed",
      "role": "assistant",
      "content": [
        {
          "type": "output_text",
          "text": "Silent circuits hum,  \nThoughts emerge in data streams—  \nDigital daw
          "annotations": []
        }
      ]
    }
  ],
  "parallel_tool_calls": true,
  "previous_response_id": null,
  "reasoning": {
    "effort": null,
    "summary": null
  },
  "store": true,
  "temperature": 1.0,
  "text": {
    "format": {
      "type": "text"
    }
  },
  "tool_choice": "auto",
  "tools": [],
  "top_p": 1.0,
```

```
  "truncation": "disabled",
  "usage": {
    "input_tokens": 32,
    "input_tokens_details": {
      "cached_tokens": 0
    },
    "output_tokens": 18,
    "output_tokens_details": {
      "reasoning_tokens": 0
    },
    "total_tokens": 50
  },
  "user": null,
  "metadata": {}
}
```

List input items

`GET https://api.openai.com/v1/responses/{response_id}/input_items`

Returns a list of input items for a given response.

Path parameters

`response_id` string Required
The ID of the response to retrieve input items for.

Query parameters

`after` string Optional
An item ID to list items after, used in pagination.

`before` string Optional
An item ID to list items before, used in pagination.

`include` array Optional
Additional fields to include in the response. See the `include` parameter for Response creation above for more information.

limit integer Optional Defaults to 20

A limit on the number of objects to be returned. Limit can range between 1 and 100, and the default is 20.

order string Optional

The order to return the input items in. Default is `desc`.

- `asc`: Return the input items in ascending order.
- `desc`: Return the input items in descending order.

Returns

A list of input item objects.

Example request curl

```
curl https://api.openai.com/v1/responses/resp_abc123/input_items \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY"
```

Response

```
{
  "object": "list",
  "data": [
    {
      "id": "msg_abc123",
      "type": "message",
      "role": "user",
      "content": [
        {
          "type": "input_text",
          "text": "Tell me a three sentence bedtime story about a unicorn."
        }
      ]
    }
  ],
  "first_id": "msg_abc123",
  "last_id": "msg_abc123",
  "has_more": false
}
```

The response object

background boolean or null

Whether to run the model response in the background. Learn more.

created_at number

Unix timestamp (in seconds) of when this Response was created.

error object or null

An error object returned when the model fails to generate a Response.

Hide properties

code string

The error code for the response.

message string

A human-readable description of the error.

id string

Unique identifier for this Response.

incomplete_details object or null

Details about why the response is incomplete.

Hide properties

reason string

The reason why the response is incomplete.

instructions string or array

A system (or developer) message inserted into the model's context.

When using along with `previous_response_id`, the instructions from a previous response will not be carried over to the next response. This makes it simple to swap out system (or developer) messages in new responses.

Hide possible types

string

A text input to the model, equivalent to a text input with the `developer` role.

Input item list array

A list of one or many input items to the model, containing different content types.

Hide possible types

Input message object

A message input to the model with a role indicating instruction following hierarchy. Instructions given with the `developer` or `system` role take precedence over instructions given with the `user` role. Messages with the `assistant` role are presumed to have been generated by the model in previous interactions.

Hide properties

content string or array

Text, image, or audio input to the model, used to generate a response. Can also contain previous assistant responses.

Hide possible types

Text input string

A text input to the model.

Input item content list array

A list of one or many input items to the model, containing different content types.

Hide possible types

Input text object

A text input to the model.

Hide properties

text string

The text input to the model.

type string

The type of the input item. Always `input_text`.

Input image object

An image input to the model. Learn about image inputs.

Hide properties

detail string

The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

type string

The type of the input item. Always `input_image`.

file_id string
The ID of the file to be sent to the model.

image_url string
The URL of the image to be sent to the model. A fully qualified URL or base64 encoded image in a data URL.

Input file object
A file input to the model.

Hide properties

type string
The type of the input item. Always `input_file`.

file_data string
The content of the file to be sent to the model.

file_id string
The ID of the file to be sent to the model.

file_url string
The URL of the file to be sent to the model.

filename string
The name of the file to be sent to the model.

role string
The role of the message input. One of `user`, `assistant`, `system`, or `developer`.

type string
The type of the message input. Always `message`.

Item object
An item representing part of the context for the response to be generated by the model. Can contain text, images, and audio inputs, as well as previous assistant responses and tool call outputs.

Hide possible types

Input message object
A message input to the model with a role indicating instruction following hierarchy. Instructions

given with the `developer` or `system` role take precedence over instructions given with the `user` role.

Hide properties

`content` array

A list of one or many input items to the model, containing different content types.

Hide possible types

Input text object

A text input to the model.

Hide properties

`text` string

The text input to the model.

`type` string

The type of the input item. Always `input_text`.

Input image object

An image input to the model. Learn about image inputs.

Hide properties

`detail` string

The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

`type` string

The type of the input item. Always `input_image`.

`file_id` string

The ID of the file to be sent to the model.

`image_url` string

The URL of the image to be sent to the model. A fully qualified URL or base64 encoded image in a data URL.

Input file object

A file input to the model.

Hide properties

type string
The type of the input item. Always `input_file`.

file_data string
The content of the file to be sent to the model.

file_id string
The ID of the file to be sent to the model.

file_url string
The URL of the file to be sent to the model.

filename string
The name of the file to be sent to the model.

role string
The role of the message input. One of `user`, `system`, or `developer`.

status string
The status of item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

type string
The type of the message input. Always set to `message`.

Output message object
An output message from the model.

Hide properties

content array
The content of the output message.

Hide possible types

Output text object
A text output from the model.

Hide properties

annotations array
The annotations of the text output.

Hide possible types

File citation object

A citation to a file.

Hide properties

file_id string

The ID of the file.

filename string

The filename of the file cited.

index integer

The index of the file in the list of files.

type string

The type of the file citation. Always `file_citation`.

URL citation object

A citation for a web resource used to generate a model response.

Hide properties

end_index integer

The index of the last character of the URL citation in the message.

start_index integer

The index of the first character of the URL citation in the message.

title string

The title of the web resource.

type string

The type of the URL citation. Always `url_citation`.

url string

The URL of the web resource.

Container file citation object

A citation for a container file used to generate a model response.

Hide properties

container_id string

The ID of the container file.

end_index integer
The index of the last character of the container file citation in the message.

file_id string
The ID of the file.

filename string
The filename of the container file cited.

start_index integer
The index of the first character of the container file citation in the message.

type string
The type of the container file citation. Always `container_file_citation`.

File path object
A path to a file.

Hide properties

file_id string
The ID of the file.

index integer
The index of the file in the list of files.

type string
The type of the file path. Always `file_path`.

text string
The text output from the model.

type string
The type of the output text. Always `output_text`.

logprobs array

Hide properties

bytes array

logprob number

token string

top_logprobs array

Hide properties

bytes array

logprob number

token string

Refusal object

A refusal from the model.

Hide properties

refusal string

The refusal explanationfrom the model.

type string

The type of the refusal. Always `refusal`.

id string

The unique ID of the output message.

role string

The role of the output message. Always `assistant`.

status string

The status of the message input. One of `in_progress`, `completed`, or `incomplete`. Populated when input items are returned via API.

type string

The type of the output message. Always `message`.

File search tool call object

The results of a file search tool call. See the file search guide for more information.

Hide properties

id string

The unique ID of the file search tool call.

queries array
The queries used to search for files.

status string
The status of the file search tool call. One of `in_progress`, `searching`, `incomplete` or `failed`,

type string
The type of the file search tool call. Always `file_search_call`.

results array or null
The results of the file search tool call.

Hide properties

attributes map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard. Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters, booleans, or numbers.

file_id string
The unique ID of the file.

filename string
The name of the file.

score number
The relevance score of the file - a value between 0 and 1.

text string
The text that was retrieved from the file.

Computer tool call object
A tool call to a computer use tool. See the computer use guide for more information.

Hide properties

action object

Hide possible types

Click object

A click action.

Hide properties

button string
Indicates which mouse button was pressed during the click. One of `left`, `right`, `wheel`, `back`, or `forward`.

type string
Specifies the event type. For a click action, this property is always set to `click`.

x integer
The x-coordinate where the click occurred.

y integer
The y-coordinate where the click occurred.

DoubleClick object
A double click action.

Hide properties

type string
Specifies the event type. For a double click action, this property is always set to `double_click`.

x integer
The x-coordinate where the double click occurred.

y integer
The y-coordinate where the double click occurred.

Drag object
A drag action.

Hide properties

path array
An array of coordinates representing the path of the drag action. Coordinates will appear as an array of objects, eg

```
[
  { x: 100, y: 200 },
  { x: 200, y: 300 }
]
```

Hide properties

x integer
The x-coordinate.

y integer
The y-coordinate.

type string
Specifies the event type. For a drag action, this property is always set to `drag`.

KeyPress object
A collection of keypresses the model would like to perform.

Hide properties

keys array
The combination of keys the model is requesting to be pressed. This is an array of strings, each representing a key.

type string
Specifies the event type. For a keypress action, this property is always set to `keypress`.

Move object
A mouse move action.

Hide properties

type string
Specifies the event type. For a move action, this property is always set to `move`.

x integer
The x-coordinate to move to.

y integer
The y-coordinate to move to.

Screenshot object

A screenshot action.

Hide properties

type string

Specifies the event type. For a screenshot action, this property is always set to `screenshot`.

Scroll object

A scroll action.

Hide properties

scroll_x integer

The horizontal scroll distance.

scroll_y integer

The vertical scroll distance.

type string

Specifies the event type. For a scroll action, this property is always set to `scroll`.

x integer

The x-coordinate where the scroll occurred.

y integer

The y-coordinate where the scroll occurred.

Type object

An action to type in text.

Hide properties

text string

The text to type.

type string

Specifies the event type. For a type action, this property is always set to `type`.

Wait object

A wait action.

Hide properties

type string
Specifies the event type. For a wait action, this property is always set to `wait`.

call_id string
An identifier used when responding to the tool call with output.

id string
The unique ID of the computer call.

pending_safety_checks array
The pending safety checks for the computer call.

Hide properties

code string
The type of the pending safety check.

id string
The ID of the pending safety check.

message string
Details about the pending safety check.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

type string
The type of the computer call. Always `computer_call`.

Computer tool call output object
The output of a computer tool call.

Hide properties

call_id string
The ID of the computer tool call that produced the output.

output object
A computer screenshot image used with the computer use tool.

Hide properties

type string
Specifies the event type. For a computer screenshot, this property is always set to `computer_screenshot`.

file_id string
The identifier of an uploaded file that contains the screenshot.

image_url string
The URL of the screenshot image.

type string
The type of the computer tool call output. Always `computer_call_output`.

acknowledged_safety_checks array
The safety checks reported by the API that have been acknowledged by the developer.

Hide properties

id string
The ID of the pending safety check.

code string
The type of the pending safety check.

message string
Details about the pending safety check.

id string
The ID of the computer tool call output.

status string
The status of the message input. One of `in_progress`, `completed`, or `incomplete`. Populated when input items are returned via API.

Web search tool call object
The results of a web search tool call. See the web search guide for more information.

Hide properties

action object
An object describing the specific action taken in this web search call. Includes details on how the model used the web (search, open_page, find).

Hide possible types

Search action object
Action type "search" - Performs a web search query.

Hide properties

query string
The search query.

type string
The action type.

Open page action object
Action type "open_page" - Opens a specific URL from search results.

Hide properties

type string
The action type.

url string
The URL opened by the model.

Find action object
Action type "find": Searches for a pattern within a loaded page.

Hide properties

pattern string
The pattern or text to search for within the page.

type string
The action type.

url string
The URL of the page searched for the pattern.

id string
The unique ID of the web search tool call.

status string
The status of the web search tool call.

type string
The type of the web search tool call. Always `web_search_call`.

Function tool call object
A tool call to run a function. See the function calling guide for more information.

Hide properties

arguments string
A JSON string of the arguments to pass to the function.

call_id string
The unique ID of the function tool call generated by the model.

name string
The name of the function to run.

type string
The type of the function tool call. Always `function_call`.

id string
The unique ID of the function tool call.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Function tool call output object
The output of a function tool call.

Hide properties

call_id string
The unique ID of the function tool call generated by the model.

output string
A JSON string of the output of the function tool call.

type string
The type of the function tool call output. Always `function_call_output`.

id string
The unique ID of the function tool call output. Populated when this item is returned via API.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Reasoning object
A description of the chain of thought used by a reasoning model while generating a response. Be sure to include these items in your `input` to the Responses API for subsequent turns of a conversation if you are manually managing context.

Hide properties

id string
The unique identifier of the reasoning content.

summary array
Reasoning text contents.

Hide properties

text string
A short summary of the reasoning used by the model when generating the response.

type string
The type of the object. Always `summary_text`.

type string
The type of the object. Always `reasoning`.

encrypted_content string or null
The encrypted content of the reasoning item - populated when a response is generated with `reasoning.encrypted_content` in the `include` parameter.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Image generation call object
An image generation request made by the model.

Hide properties

id string
The unique ID of the image generation call.

result string or null
The generated image encoded in base64.

status string
The status of the image generation call.

type string
The type of the image generation call. Always `image_generation_call`.

Code interpreter tool call object
A tool call to run code.

Hide properties

code string or null
The code to run, or null if not available.

container_id string
The ID of the container used to run the code.

id string
The unique ID of the code interpreter tool call.

outputs array or null
The outputs generated by the code interpreter, such as logs or images. Can be null if no outputs are available.

Hide possible types

Code interpreter output logs object
The logs output from the code interpreter.

Hide properties

logs string
The logs output from the code interpreter.

type string
The type of the output. Always 'logs'.

Code interpreter output image object
The image output from the code interpreter.

Hide properties

type string
The type of the output. Always 'image'.

url string
The URL of the image output from the code interpreter.

status string
The status of the code interpreter tool call.

type string
The type of the code interpreter tool call. Always `code_interpreter_call`.

Local shell call object
A tool call to run a command on the local shell.

Hide properties

action object
Execute a shell command on the server.

Hide properties

command array
The command to run.

env object
Environment variables to set for the command.

type string
The type of the local shell action. Always `exec`.

timeout_ms integer or null
Optional timeout in milliseconds for the command.

user string or null
Optional user to run the command as.

working_directory string or null
Optional working directory to run the command in.

call_id string
The unique ID of the local shell tool call generated by the model.

id string
The unique ID of the local shell call.

status string
The status of the local shell call.

type string
The type of the local shell call. Always `local_shell_call`.

Local shell call output object
The output of a local shell tool call.

Hide properties

id string
The unique ID of the local shell tool call generated by the model.

output string
A JSON string of the output of the local shell tool call.

type string
The type of the local shell tool call output. Always `local_shell_call_output`.

status string or null
The status of the item. One of `in_progress`, `completed`, or `incomplete`.

MCP list tools object
A list of tools available on an MCP server.

Hide properties

id string
The unique ID of the list.

server_label string
The label of the MCP server.

tools array
The tools available on the server.

Hide properties

input_schema object
The JSON schema describing the tool's input.

name string
The name of the tool.

annotations object or null
Additional annotations about the tool.

description string or null
The description of the tool.

type string
The type of the item. Always `mcp_list_tools`.

error string or null
Error message if the server could not list tools.

MCP approval request object
A request for human approval of a tool invocation.

Hide properties

arguments string
A JSON string of arguments for the tool.

id string
The unique ID of the approval request.

name string
The name of the tool to run.

server_label string
The label of the MCP server making the request.

type string
The type of the item. Always `mcp_approval_request`.

MCP approval response object
A response to an MCP approval request.

Hide properties

approval_request_id string
The ID of the approval request being answered.

approve boolean
Whether the request was approved.

type string
The type of the item. Always `mcp_approval_response`.

id string or null
The unique ID of the approval response

reason string or null
Optional reason for the decision.

MCP tool call object
An invocation of a tool on an MCP server.

^ Hide properties

arguments string
A JSON string of the arguments passed to the tool.

id string
The unique ID of the tool call.

name string
The name of the tool that was run.

server_label string
The label of the MCP server running the tool.

type string
The type of the item. Always `mcp_call`.

error string or null
The error from the tool call, if any.

output string or null
The output from the tool call.

Item reference object

An internal identifier for an item to reference.

Hide properties

id string

The ID of the item to reference.

type string

The type of item to reference. Always `item_reference`.

max_output_tokens integer or null

An upper bound for the number of tokens that can be generated for a response, including visible output tokens and reasoning tokens.

max_tool_calls integer or null

The maximum number of total calls to built-in tools that can be processed in a response. This maximum number applies across all built-in tool calls, not per individual tool. Any further attempts to call a tool by the model will be ignored.

metadata map

Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

model string

Model ID used to generate the response, like `gpt-4o` or `o3`. OpenAI offers a wide range of models with different capabilities, performance characteristics, and price points. Refer to the model guide to browse and compare available models.

object string

The object type of this resource - always set to `response`.

output array

An array of content items generated by the model.

The length and order of items in the `output` array is dependent on the model's response.

Rather than accessing the first item in the `output` array and assuming it's an `assistant` message with the content generated by the model, you might consider using the `output_text` property where

supported in SDKs.

Hide possible types

Output message object

An output message from the model.

Hide properties

content array

The content of the output message.

Hide possible types

Output text object

A text output from the model.

Hide properties

annotations array

The annotations of the text output.

Hide possible types

File citation object

A citation to a file.

Hide properties

file_id string

The ID of the file.

filename string

The filename of the file cited.

index integer

The index of the file in the list of files.

type string

The type of the file citation. Always `file_citation` .

URL citation object

A citation for a web resource used to generate a model response.

Hide properties

end_index integer

The index of the last character of the URL citation in the message.

start_index integer
The index of the first character of the URL citation in the message.

title string
The title of the web resource.

type string
The type of the URL citation. Always `url_citation`.

url string
The URL of the web resource.

Container file citation object
A citation for a container file used to generate a model response.

Hide properties

container_id string
The ID of the container file.

end_index integer
The index of the last character of the container file citation in the message.

file_id string
The ID of the file.

filename string
The filename of the container file cited.

start_index integer
The index of the first character of the container file citation in the message.

type string
The type of the container file citation. Always `container_file_citation`.

File path object
A path to a file.

Hide properties

file_id string
The ID of the file.

index integer
The index of the file in the list of files.

type string
The type of the file path. Always `file_path` .

text string
The text output from the model.

type string
The type of the output text. Always `output_text` .

logprobs array
Hide properties

bytes array

logprob number

token string

top_logprobs array
Hide properties

bytes array

logprob number

token string

Refusal object
A refusal from the model.
Hide properties

refusal string
The refusal explanationfrom the model.

type string
The type of the refusal. Always `refusal` .

id string
The unique ID of the output message.

role string
The role of the output message. Always `assistant`.

status string
The status of the message input. One of `in_progress`, `completed`, or `incomplete`. Populated when input items are returned via API.

type string
The type of the output message. Always `message`.

File search tool call object
The results of a file search tool call. See the file search guide for more information.

Hide properties

id string
The unique ID of the file search tool call.

queries array
The queries used to search for files.

status string
The status of the file search tool call. One of `in_progress`, `searching`, `incomplete` or `failed`,

type string
The type of the file search tool call. Always `file_search_call`.

results array or null
The results of the file search tool call.

Hide properties

attributes map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard. Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters, booleans, or numbers.

file_id string
The unique ID of the file.

filename string
The name of the file.

score number
The relevance score of the file - a value between 0 and 1.

text string
The text that was retrieved from the file.

Function tool call object
A tool call to run a function. See the function calling guide for more information.

Hide properties

arguments string
A JSON string of the arguments to pass to the function.

call_id string
The unique ID of the function tool call generated by the model.

name string
The name of the function to run.

type string
The type of the function tool call. Always `function_call`.

id string
The unique ID of the function tool call.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Web search tool call object
The results of a web search tool call. See the web search guide for more information.

Hide properties

action object
An object describing the specific action taken in this web search call. Includes details on how the model used the web (search, open_page, find).

Hide possible types

Search action object

Action type "search" - Performs a web search query.

^ Hide properties

query string

The search query.

type string

The action type.

Open page action object

Action type "open_page" - Opens a specific URL from search results.

^ Hide properties

type string

The action type.

url string

The URL opened by the model.

Find action object

Action type "find": Searches for a pattern within a loaded page.

^ Hide properties

pattern string

The pattern or text to search for within the page.

type string

The action type.

url string

The URL of the page searched for the pattern.

id string

The unique ID of the web search tool call.

status string

The status of the web search tool call.

type string

The type of the web search tool call. Always `web_search_call`.

Computer tool call object

A tool call to a computer use tool. See the computer use guide for more information.

Hide properties

action object

Hide possible types

Click object

A click action.

Hide properties

button string

Indicates which mouse button was pressed during the click. One of `left`, `right`, `wheel`, `back`, or `forward`.

type string

Specifies the event type. For a click action, this property is always set to `click`.

x integer

The x-coordinate where the click occurred.

y integer

The y-coordinate where the click occurred.

DoubleClick object

A double click action.

Hide properties

type string

Specifies the event type. For a double click action, this property is always set to `double_click`.

x integer

The x-coordinate where the double click occurred.

y integer

The y-coordinate where the double click occurred.

Drag object

A drag action.

Hide properties

path array

An array of coordinates representing the path of the drag action. Coordinates will appear as an array of objects, eg

```
[
  { x: 100, y: 200 },
  { x: 200, y: 300 }
]
```

Hide properties

x integer

The x-coordinate.

y integer

The y-coordinate.

type string

Specifies the event type. For a drag action, this property is always set to `drag`.

KeyPress object

A collection of keypresses the model would like to perform.

Hide properties

keys array

The combination of keys the model is requesting to be pressed. This is an array of strings, each representing a key.

type string

Specifies the event type. For a keypress action, this property is always set to `keypress`.

Move object

A mouse move action.

Hide properties

type string

Specifies the event type. For a move action, this property is always set to `move`.

x integer
The x-coordinate to move to.

y integer
The y-coordinate to move to.

Screenshot object
A screenshot action.

Hide properties

type string
Specifies the event type. For a screenshot action, this property is always set to `screenshot`.

Scroll object
A scroll action.

Hide properties

scroll_x integer
The horizontal scroll distance.

scroll_y integer
The vertical scroll distance.

type string
Specifies the event type. For a scroll action, this property is always set to `scroll`.

x integer
The x-coordinate where the scroll occurred.

y integer
The y-coordinate where the scroll occurred.

Type object
An action to type in text.

Hide properties

text string
The text to type.

type string
Specifies the event type. For a type action, this property is always set to `type`.

Wait object
A wait action.

Hide properties

type string
Specifies the event type. For a wait action, this property is always set to `wait`.

call_id string
An identifier used when responding to the tool call with output.

id string
The unique ID of the computer call.

pending_safety_checks array
The pending safety checks for the computer call.

Hide properties

code string
The type of the pending safety check.

id string
The ID of the pending safety check.

message string
Details about the pending safety check.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

type string
The type of the computer call. Always `computer_call`.

Reasoning object
A description of the chain of thought used by a reasoning model while generating a response. Be sure to include these items in your `input` to the Responses API for subsequent turns of a conversation if you

are manually managing context.

Hide properties

id string
The unique identifier of the reasoning content.

summary array
Reasoning text contents.

Hide properties

text string
A short summary of the reasoning used by the model when generating the response.

type string
The type of the object. Always `summary_text`.

type string
The type of the object. Always `reasoning`.

encrypted_content string or null
The encrypted content of the reasoning item - populated when a response is generated with `reasoning.encrypted_content` in the `include` parameter.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Image generation call object
An image generation request made by the model.

Hide properties

id string
The unique ID of the image generation call.

result string or null
The generated image encoded in base64.

status string
The status of the image generation call.

type string
The type of the image generation call. Always `image_generation_call`.

Code interpreter tool call object
A tool call to run code.

Hide properties

code string or null
The code to run, or null if not available.

container_id string
The ID of the container used to run the code.

id string
The unique ID of the code interpreter tool call.

outputs array or null
The outputs generated by the code interpreter, such as logs or images. Can be null if no outputs are available.

Hide possible types

Code interpreter output logs object
The logs output from the code interpreter.

Hide properties

logs string
The logs output from the code interpreter.

type string
The type of the output. Always 'logs'.

Code interpreter output image object
The image output from the code interpreter.

Hide properties

type string
The type of the output. Always 'image'.

url string
The URL of the image output from the code interpreter.

status string
The status of the code interpreter tool call.

type string
The type of the code interpreter tool call. Always `code_interpreter_call`.

Local shell call object
A tool call to run a command on the local shell.

Hide properties

action object
Execute a shell command on the server.

Hide properties

command array
The command to run.

env object
Environment variables to set for the command.

type string
The type of the local shell action. Always `exec`.

timeout_ms integer or null
Optional timeout in milliseconds for the command.

user string or null
Optional user to run the command as.

working_directory string or null
Optional working directory to run the command in.

call_id string
The unique ID of the local shell tool call generated by the model.

id string
The unique ID of the local shell call.

status string
The status of the local shell call.

type string
The type of the local shell call. Always `local_shell_call` .

MCP tool call object
An invocation of a tool on an MCP server.

Hide properties

arguments string
A JSON string of the arguments passed to the tool.

id string
The unique ID of the tool call.

name string
The name of the tool that was run.

server_label string
The label of the MCP server running the tool.

type string
The type of the item. Always `mcp_call` .

error string or null
The error from the tool call, if any.

output string or null
The output from the tool call.

MCP list tools object
A list of tools available on an MCP server.

Hide properties

id string
The unique ID of the list.

server_label string
The label of the MCP server.

tools array
The tools available on the server.

Hide properties

input_schema object
The JSON schema describing the tool's input.

name string
The name of the tool.

annotations object or null
Additional annotations about the tool.

description string or null
The description of the tool.

type string
The type of the item. Always `mcp_list_tools`.

error string or null
Error message if the server could not list tools.

MCP approval request object
A request for human approval of a tool invocation.

Hide properties

arguments string
A JSON string of arguments for the tool.

id string
The unique ID of the approval request.

name string
The name of the tool to run.

server_label string
The label of the MCP server making the request.

type string
The type of the item. Always `mcp_approval_request`.

`output_text` string or null SDK Only
SDK-only convenience property that contains the aggregated text output from all `output_text` items in the `output` array, if any are present. Supported in the Python and JavaScript SDKs.

`parallel_tool_calls` boolean
Whether to allow the model to run tool calls in parallel.

`previous_response_id` string or null
The unique ID of the previous response to the model. Use this to create multi-turn conversations. Learn more about conversation state.

`prompt` object or null
Reference to a prompt template and its variables. Learn more.

^ Hide properties

`id` string
The unique identifier of the prompt template to use.

`variables` map
Optional map of values to substitute in for variables in your prompt. The substitution values can either be strings, or other Response input types like images or files.

`version` string or null
Optional version of the prompt template.

`reasoning` object or null
o-series models only

Configuration options for reasoning models.

^ Hide properties

`effort` string or null
o-series models only

Constrains effort on reasoning for reasoning models. Currently supported values are `low`, `medium`, and `high`. Reducing reasoning effort can result in faster responses and fewer tokens used on reasoning in a response.

`generate_summary` Deprecated string or null
Deprecated: use `summary` instead.

A summary of the reasoning performed by the model. This can be useful for debugging and understanding the model's reasoning process. One of `auto`, `concise`, or `detailed`.

`summary` string or null
A summary of the reasoning performed by the model. This can be useful for debugging and understanding the model's reasoning process. One of `auto`, `concise`, or `detailed`.

`service_tier` string or null
Specifies the processing type used for serving the request.

> If set to 'auto', then the request will be processed with the service tier configured in the Project settings. Unless otherwise configured, the Project will use 'default'.
>
> If set to 'default', then the requset will be processed with the standard pricing and performance for the selected model.
>
> If set to 'flex' or 'priority', then the request will be processed with the corresponding service tier. Contact sales to learn more about Priority processing.
>
> When not set, the default behavior is 'auto'.

When the `service_tier` parameter is set, the response body will include the `service_tier` value based on the processing mode actually used to serve the request. This response value may be different from the value set in the parameter.

`status` string
The status of the response generation. One of `completed`, `failed`, `in_progress`, `cancelled`, `queued`, or `incomplete`.

`temperature` number or null
What sampling temperature to use, between 0 and 2. Higher values like 0.8 will make the output more random, while lower values like 0.2 will make it more focused and deterministic. We generally recommend altering this or `top_p` but not both.

`text` object
Configuration options for a text response from the model. Can be plain text or structured JSON data. Learn more:

> Text inputs and outputs
>
> Structured Outputs

^ Hide properties

format object

An object specifying the format that the model must output.

Configuring `{ "type": "json_schema" }` enables Structured Outputs, which ensures the model will match your supplied JSON schema. Learn more in the Structured Outputs guide.

The default format is `{ "type": "text" }` with no additional options.

Not recommended for gpt-4o and newer models:

Setting to `{ "type": "json_object" }` enables the older JSON mode, which ensures the message the model generates is valid JSON. Using `json_schema` is preferred for models that support it.

Hide possible types

Text object

Default response format. Used to generate text responses.

Hide properties

type string

The type of response format being defined. Always `text`.

JSON schema object

JSON Schema response format. Used to generate structured JSON responses. Learn more about Structured Outputs.

Hide properties

name string

The name of the response format. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

schema object

The schema for the response format, described as a JSON Schema object. Learn how to build JSON schemas here.

type string

The type of response format being defined. Always `json_schema`.

description string

A description of what the response format is for, used by the model to determine how to respond in the format.

strict boolean or null

Whether to enable strict schema adherence when generating the output. If set to true, the model will always follow the exact schema defined in the `schema` field. Only a subset of JSON Schema

is supported when `strict` is `true`. To learn more, read the Structured Outputs guide.

JSON object object

JSON object response format. An older method of generating JSON responses. Using `json_schema` is recommended for models that support it. Note that the model will not generate JSON without a system or user message instructing it to do so.

Hide properties

type string

The type of response format being defined. Always `json_object`.

tool_choice string or object

How the model should select which tool (or tools) to use when generating a response. See the `tools` parameter to see how to specify which tools the model can call.

Hide possible types

Tool choice mode string

Controls which (if any) tool is called by the model.

`none` means the model will not call any tool and instead generates a message.

`auto` means the model can pick between generating a message or calling one or more tools.

`required` means the model must call one or more tools.

Hosted tool object

Indicates that the model should use a built-in tool to generate a response. Learn more about built-in tools.

Hide properties

type string

The type of hosted tool the model should to use. Learn more about built-in tools.

Allowed values are:

`file_search`

`web_search_preview`

`computer_use_preview`

`code_interpreter`

`image_generation`

Function tool object

Use this option to force the model to call a specific function.

Hide properties

name string

The name of the function to call.

type string

For function calling, the type is always `function`.

MCP tool object

Use this option to force the model to call a specific tool on a remote MCP server.

Hide properties

server_label string

The label of the MCP server to use.

type string

For MCP tools, the type is always `mcp`.

name string or null

The name of the tool to call on the server.

tools array

An array of tools the model may call while generating a response. You can specify which tool to use by setting the `tool_choice` parameter.

The two categories of tools you can provide the model are:

Built-in tools: Tools that are provided by OpenAI that extend the model's capabilities, like web search or file search. Learn more about built-in tools.

Function calls (custom tools): Functions that are defined by you, enabling the model to call your own code. Learn more about function calling.

Hide possible types

Function object

Defines a function in your own code the model can choose to call. Learn more about function calling.

Hide properties

name string

The name of the function to call.

parameters object
A JSON schema object describing the parameters of the function.

strict boolean
Whether to enforce strict parameter validation. Default `true`.

type string
The type of the function tool. Always `function`.

description string
A description of the function. Used by the model to determine whether or not to call the function.

File search object
A tool that searches for relevant content from uploaded files. Learn more about the file search tool.

Hide properties

type string
The type of the file search tool. Always `file_search`.

vector_store_ids array
The IDs of the vector stores to search.

filters object
A filter to apply.

Hide possible types

Comparison Filter object
A filter used to compare a specified attribute key to a given value using a defined comparison operation.

Hide properties

key string
The key to compare against the value.

type string
Specifies the comparison operator: `eq`, `ne`, `gt`, `gte`, `lt`, `lte`.

- `eq`: equals
- `ne`: not equal
- `gt`: greater than
- `gte`: greater than or equal

`lt` : less than

`lte` : less than or equal

`value` string / number / boolean
The value to compare against the attribute key; supports string, number, or boolean types.

Compound Filter object
Combine multiple filters using `and` or `or` .

Hide properties

`filters` array
Array of filters to combine. Items can be `ComparisonFilter` or `CompoundFilter` .

Hide possible types

Comparison Filter object
A filter used to compare a specified attribute key to a given value using a defined comparison operation.

Hide properties

`key` string
The key to compare against the value.

`type` string
Specifies the comparison operator: `eq` , `ne` , `gt` , `gte` , `lt` , `lte` .

`eq` : equals

`ne` : not equal

`gt` : greater than

`gte` : greater than or equal

`lt` : less than

`lte` : less than or equal

`value` string / number / boolean
The value to compare against the attribute key; supports string, number, or boolean types.

`type` string

Type of operation: `and` or `or`.

`max_num_results` integer
The maximum number of results to return. This number should be between 1 and 50 inclusive.

`ranking_options` object
Ranking options for search.

Hide properties

`ranker` string
The ranker to use for the file search.

`score_threshold` number
The score threshold for the file search, a number between 0 and 1. Numbers closer to 1 will attempt to return only the most relevant results, but may return fewer results.

Web search preview object
This tool searches the web for relevant results to use in a response. Learn more about the web search tool.

Hide properties

`type` string
The type of the web search tool. One of `web_search_preview` or `web_search_preview_2025_03_11`.

`search_context_size` string
High level guidance for the amount of context window space to use for the search. One of `low`, `medium`, or `high`. `medium` is the default.

`user_location` object
The user's location.

Hide properties

`type` string
The type of location approximation. Always `approximate`.

`city` string
Free text input for the city of the user, e.g. `San Francisco`.

country string
The two-letter ISO country code of the user, e.g. `US` .

region string
Free text input for the region of the user, e.g. `California` .

timezone string
The IANA timezone of the user, e.g. `America/Los_Angeles` .

Computer use preview object
A tool that controls a virtual computer. Learn more about the computer tool.

Hide properties

display_height integer
The height of the computer display.

display_width integer
The width of the computer display.

environment string
The type of computer environment to control.

type string
The type of the computer use tool. Always `computer_use_preview` .

MCP tool object
Give the model access to additional tools via remote Model Context Protocol (MCP) servers. Learn more about MCP.

Hide properties

server_label string
A label for this MCP server, used to identify it in tool calls.

server_url string
The URL for the MCP server.

type string
The type of the MCP tool. Always `mcp` .

allowed_tools array or object
List of allowed tool names or a filter object.

Hide possible types

MCP allowed tools array
A string array of allowed tool names

MCP allowed tools filter object
A filter object to specify which tools are allowed.

Hide properties

tool_names array
List of allowed tool names.

headers object or null
Optional HTTP headers to send to the MCP server. Use for authentication or other purposes.

require_approval object or string
Specify which of the MCP server's tools require approval.

Hide possible types

MCP tool approval filter object

Hide properties

always object
A list of tools that always require approval.

Hide properties

tool_names array
List of tools that require approval.

never object
A list of tools that never require approval.

Hide properties

tool_names array
List of tools that do not require approval.

MCP tool approval setting string
Specify a single approval policy for all tools. One of `always` or `never`. When set to `always`, all tools will require approval. When set to `never`, all tools will not require approval.

server_description string
Optional description of the MCP server, used to provide more context.

Code interpreter object
A tool that runs Python code to help generate a response to a prompt.

Hide properties

container string or object
The code interpreter container. Can be a container ID or an object that specifies uploaded file IDs to make available to your code.

Hide possible types

string
The container ID.

CodeInterpreterContainerAuto object
Configuration for a code interpreter container. Optionally specify the IDs of the files to run the code on.

Hide properties

type string
Always `auto`.

file_ids array
An optional list of uploaded files to make available to your code.

type string
The type of the code interpreter tool. Always `code_interpreter`.

Image generation tool object
A tool that generates images using a model like `gpt-image-1`.

Hide properties

type string
The type of the image generation tool. Always `image_generation`.

background string
Background type for the generated image. One of `transparent`, `opaque`, or `auto`. Default: `auto`.

input_image_mask object
Optional mask for inpainting. Contains `image_url` (string, optional) and `file_id` (string, optional).

Hide properties

file_id string
File ID for the mask image.

image_url string
Base64-encoded mask image.

model string
The image generation model to use. Default: `gpt-image-1`.

moderation string
Moderation level for the generated image. Default: `auto`.

output_compression integer
Compression level for the output image. Default: 100.

output_format string
The output format of the generated image. One of `png`, `webp`, or `jpeg`. Default: `png`.

partial_images integer
Number of partial images to generate in streaming mode, from 0 (default value) to 3.

quality string
The quality of the generated image. One of `low`, `medium`, `high`, or `auto`. Default: `auto`.

size string
The size of the generated image. One of `1024x1024`, `1024x1536`, `1536x1024`, or `auto`. Default: `auto`.

Local shell tool object
A tool that allows the model to execute shell commands in a local environment.

Hide properties

type string
The type of the local shell tool. Always `local_shell`.

top_logprobs integer or null
An integer between 0 and 20 specifying the number of most likely tokens to return at each token position, each with an associated log probability.

top_p number or null
An alternative to sampling with temperature, called nucleus sampling, where the model considers the results of the tokens with top_p probability mass. So 0.1 means only the tokens comprising the top 10% probability mass are considered.

We generally recommend altering this or `temperature` but not both.

truncation string or null
The truncation strategy to use for the model response.

- `auto` : If the context of this response and previous ones exceeds the model's context window size, the model will truncate the response to fit the context window by dropping input items in the middle of the conversation.
- `disabled` (default): If a model response will exceed the context window size for a model, the request will fail with a 400 error.

usage object
Represents token usage details including input tokens, output tokens, a breakdown of output tokens, and the total tokens used.

Hide properties

input_tokens integer
The number of input tokens.

input_tokens_details object
A detailed breakdown of the input tokens.

Hide properties

cached_tokens integer
The number of tokens that were retrieved from the cache. More on prompt caching.

output_tokens integer
The number of output tokens.

output_tokens_details object
A detailed breakdown of the output tokens.

Hide properties

reasoning_tokens integer
The number of reasoning tokens.

total_tokens integer
The total number of tokens used.

user string
A stable identifier for your end-users. Used to boost cache hit rates by better bucketing similar requests and to help OpenAI detect and prevent abuse. Learn more.

OBJECT The response object


```
{
  "id": "resp_67ccd3a9da748190baa7f1570fe91ac604becb25c45c1d41",
  "object": "response",
  "created_at": 1741476777,
  "status": "completed",
  "error": null,
  "incomplete_details": null,
  "instructions": null,
  "max_output_tokens": null,
  "model": "gpt-4o-2024-08-06",
  "output": [
    {
      "type": "message",
      "id": "msg_67ccd3acc8d48190a77525dc6de64b4104becb25c45c1d41",
      "status": "completed",
      "role": "assistant",
      "content": [
        {
          "type": "output_text",
          "text": "The image depicts a scenic landscape with a wooden boardwalk or pathway
          "annotations": []
        }
      ]
    }
  ],
  "parallel_tool_calls": true,
  "previous_response_id": null,
  "reasoning": {
    "effort": null,
    "summary": null
  },
```

```
  "store": true,
  "temperature": 1,
  "text": {
    "format": {
      "type": "text"
    }
  },
  "tool_choice": "auto",
  "tools": [],
  "top_p": 1,
  "truncation": "disabled",
  "usage": {
    "input_tokens": 328,
    "input_tokens_details": {
      "cached_tokens": 0
    },
    "output_tokens": 52,
    "output_tokens_details": {
      "reasoning_tokens": 0
    },
    "total_tokens": 380
  },
  "user": null,
  "metadata": {}
}
```

The input item list

A list of Response items.

data array

A list of items used to generate this response.

Hide possible types

Input message object

A message input to the model with a role indicating instruction following hierarchy. Instructions given with the `developer` or `system` role take precedence over instructions given with the `user` role.

Hide properties

content array

A list of one or many input items to the model, containing different content types.

Hide possible types

Input text object

A text input to the model.

Hide properties

text string

The text input to the model.

type string

The type of the input item. Always `input_text`.

Input image object

An image input to the model. Learn about image inputs.

Hide properties

detail string

The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

type string

The type of the input item. Always `input_image`.

file_id string

The ID of the file to be sent to the model.

image_url string

The URL of the image to be sent to the model. A fully qualified URL or base64 encoded image in a data URL.

Input file object

A file input to the model.

Hide properties

type string

The type of the input item. Always `input_file`.

file_data string

The content of the file to be sent to the model.

file_id string

The ID of the file to be sent to the model.

file_url string
The URL of the file to be sent to the model.

filename string
The name of the file to be sent to the model.

id string
The unique ID of the message input.

role string
The role of the message input. One of `user`, `system`, or `developer`.

status string
The status of item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

type string
The type of the message input. Always set to `message`.

Output message object
An output message from the model.

Hide properties

content array
The content of the output message.

Hide possible types

Output text object
A text output from the model.

Hide properties

annotations array
The annotations of the text output.

Hide possible types

File citation object
A citation to a file.

Hide properties

file_id string
The ID of the file.

filename string
The filename of the file cited.

index integer
The index of the file in the list of files.

type string
The type of the file citation. Always `file_citation`.

URL citation object
A citation for a web resource used to generate a model response.

Hide properties

end_index integer
The index of the last character of the URL citation in the message.

start_index integer
The index of the first character of the URL citation in the message.

title string
The title of the web resource.

type string
The type of the URL citation. Always `url_citation`.

url string
The URL of the web resource.

Container file citation object
A citation for a container file used to generate a model response.

Hide properties

container_id string
The ID of the container file.

end_index integer
The index of the last character of the container file citation in the message.

file_id string
The ID of the file.

filename string
The filename of the container file cited.

start_index integer
The index of the first character of the container file citation in the message.

type string
The type of the container file citation. Always `container_file_citation`.

File path object
A path to a file.

Hide properties

file_id string
The ID of the file.

index integer
The index of the file in the list of files.

type string
The type of the file path. Always `file_path`.

text string
The text output from the model.

type string
The type of the output text. Always `output_text`.

logprobs array

Hide properties

bytes array

logprob number

token string

top_logprobs array

Hide properties

bytes array

logprob number

token string

Refusal object

A refusal from the model.

Hide properties

refusal string

The refusal explanationfrom the model.

type string

The type of the refusal. Always `refusal`.

id string

The unique ID of the output message.

role string

The role of the output message. Always `assistant`.

status string

The status of the message input. One of `in_progress`, `completed`, or `incomplete`. Populated when input items are returned via API.

type string

The type of the output message. Always `message`.

File search tool call object

The results of a file search tool call. See the file search guide for more information.

Hide properties

id string

The unique ID of the file search tool call.

queries array
The queries used to search for files.

status string
The status of the file search tool call. One of `in_progress`, `searching`, `incomplete` or `failed`,

type string
The type of the file search tool call. Always `file_search_call`.

results array or null
The results of the file search tool call.

Hide properties

attributes map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard. Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters, booleans, or numbers.

file_id string
The unique ID of the file.

filename string
The name of the file.

score number
The relevance score of the file - a value between 0 and 1.

text string
The text that was retrieved from the file.

Computer tool call object
A tool call to a computer use tool. See the computer use guide for more information.

Hide properties

action object

Hide possible types

Click object
A click action.

Hide properties

button string
Indicates which mouse button was pressed during the click. One of `left`, `right`, `wheel`, `back`, or `forward`.

type string
Specifies the event type. For a click action, this property is always set to `click`.

x integer
The x-coordinate where the click occurred.

y integer
The y-coordinate where the click occurred.

DoubleClick object
A double click action.

Hide properties

type string
Specifies the event type. For a double click action, this property is always set to `double_click`.

x integer
The x-coordinate where the double click occurred.

y integer
The y-coordinate where the double click occurred.

Drag object
A drag action.

Hide properties

path array
An array of coordinates representing the path of the drag action. Coordinates will appear as an array of objects, eg

```
[
  { x: 100, y: 200 },
  { x: 200, y: 300 }
]
```

Hide properties

x integer
The x-coordinate.

y integer
The y-coordinate.

type string
Specifies the event type. For a drag action, this property is always set to `drag`.

KeyPress object
A collection of keypresses the model would like to perform.

Hide properties

keys array
The combination of keys the model is requesting to be pressed. This is an array of strings, each representing a key.

type string
Specifies the event type. For a keypress action, this property is always set to `keypress`.

Move object
A mouse move action.

Hide properties

type string
Specifies the event type. For a move action, this property is always set to `move`.

x integer
The x-coordinate to move to.

y integer
The y-coordinate to move to.

Screenshot object
A screenshot action.

Hide properties

type string
Specifies the event type. For a screenshot action, this property is always set to `screenshot`.

Scroll object
A scroll action.

Hide properties

scroll_x integer
The horizontal scroll distance.

scroll_y integer
The vertical scroll distance.

type string
Specifies the event type. For a scroll action, this property is always set to `scroll`.

x integer
The x-coordinate where the scroll occurred.

y integer
The y-coordinate where the scroll occurred.

Type object
An action to type in text.

Hide properties

text string
The text to type.

type string
Specifies the event type. For a type action, this property is always set to `type`.

Wait object
A wait action.

Hide properties

type string
Specifies the event type. For a wait action, this property is always set to `wait`.

call_id string

An identifier used when responding to the tool call with output.

id string
The unique ID of the computer call.

pending_safety_checks array
The pending safety checks for the computer call.

Hide properties

code string
The type of the pending safety check.

id string
The ID of the pending safety check.

message string
Details about the pending safety check.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

type string
The type of the computer call. Always `computer_call`.

Computer tool call output object
The output of a computer tool call.

Hide properties

call_id string
The ID of the computer tool call that produced the output.

id string
The unique ID of the computer call tool output.

output object
A computer screenshot image used with the computer use tool.

Hide properties

type string
Specifies the event type. For a computer screenshot, this property is always set to `computer_screenshot`.

file_id string
The identifier of an uploaded file that contains the screenshot.

image_url string
The URL of the screenshot image.

type string
The type of the computer tool call output. Always `computer_call_output`.

acknowledged_safety_checks array
The safety checks reported by the API that have been acknowledged by the developer.

Hide properties

code string
The type of the pending safety check.

id string
The ID of the pending safety check.

message string
Details about the pending safety check.

status string
The status of the message input. One of `in_progress`, `completed`, or `incomplete`. Populated when input items are returned via API.

Web search tool call object
The results of a web search tool call. See the web search guide for more information.

Hide properties

action object
An object describing the specific action taken in this web search call. Includes details on how the model used the web (search, open_page, find).

Hide possible types

Search action object
Action type "search" - Performs a web search query.

Hide properties

query string
The search query.

type string
The action type.

Open page action object
Action type "open_page" - Opens a specific URL from search results.

Hide properties

type string
The action type.

url string
The URL opened by the model.

Find action object
Action type "find": Searches for a pattern within a loaded page.

Hide properties

pattern string
The pattern or text to search for within the page.

type string
The action type.

url string
The URL of the page searched for the pattern.

id string
The unique ID of the web search tool call.

status string
The status of the web search tool call.

type string
The type of the web search tool call. Always `web_search_call` .

Function tool call object

A tool call to run a function. See the function calling guide for more information.

Hide properties

arguments string

A JSON string of the arguments to pass to the function.

call_id string

The unique ID of the function tool call generated by the model.

id string

The unique ID of the function tool call.

name string

The name of the function to run.

type string

The type of the function tool call. Always `function_call`.

status string

The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Function tool call output object

The output of a function tool call.

Hide properties

call_id string

The unique ID of the function tool call generated by the model.

id string

The unique ID of the function call tool output.

output string

A JSON string of the output of the function tool call.

type string

The type of the function tool call output. Always `function_call_output`.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Image generation call object
An image generation request made by the model.

Hide properties

id string
The unique ID of the image generation call.

result string or null
The generated image encoded in base64.

status string
The status of the image generation call.

type string
The type of the image generation call. Always `image_generation_call`.

Code interpreter tool call object
A tool call to run code.

Hide properties

code string or null
The code to run, or null if not available.

container_id string
The ID of the container used to run the code.

id string
The unique ID of the code interpreter tool call.

outputs array or null
The outputs generated by the code interpreter, such as logs or images. Can be null if no outputs are available.

Hide possible types

Code interpreter output logs object
The logs output from the code interpreter.

Hide properties

logs string
The logs output from the code interpreter.

type string
The type of the output. Always 'logs'.

Code interpreter output image object
The image output from the code interpreter.

Hide properties

type string
The type of the output. Always 'image'.

url string
The URL of the image output from the code interpreter.

status string
The status of the code interpreter tool call.

type string
The type of the code interpreter tool call. Always `code_interpreter_call`.

Local shell call object
A tool call to run a command on the local shell.

Hide properties

action object
Execute a shell command on the server.

Hide properties

command array
The command to run.

env object
Environment variables to set for the command.

type string
The type of the local shell action. Always `exec`.

timeout_ms integer or null

Optional timeout in milliseconds for the command.

user string or null
Optional user to run the command as.

working_directory string or null
Optional working directory to run the command in.

call_id string
The unique ID of the local shell tool call generated by the model.

id string
The unique ID of the local shell call.

status string
The status of the local shell call.

type string
The type of the local shell call. Always `local_shell_call`.

Local shell call output object
The output of a local shell tool call.

Hide properties

id string
The unique ID of the local shell tool call generated by the model.

output string
A JSON string of the output of the local shell tool call.

type string
The type of the local shell tool call output. Always `local_shell_call_output`.

status string or null
The status of the item. One of `in_progress`, `completed`, or `incomplete`.

MCP list tools object
A list of tools available on an MCP server.

Hide properties

id string

The unique ID of the list.

server_label string
The label of the MCP server.

tools array
The tools available on the server.

Hide properties

input_schema object
The JSON schema describing the tool's input.

name string
The name of the tool.

annotations object or null
Additional annotations about the tool.

description string or null
The description of the tool.

type string
The type of the item. Always `mcp_list_tools`.

error string or null
Error message if the server could not list tools.

MCP approval request object
A request for human approval of a tool invocation.

Hide properties

arguments string
A JSON string of arguments for the tool.

id string
The unique ID of the approval request.

name string
The name of the tool to run.

server_label string
The label of the MCP server making the request.

type string
The type of the item. Always `mcp_approval_request`.

MCP approval response object
A response to an MCP approval request.

Hide properties

approval_request_id string
The ID of the approval request being answered.

approve boolean
Whether the request was approved.

id string
The unique ID of the approval response

type string
The type of the item. Always `mcp_approval_response`.

reason string or null
Optional reason for the decision.

MCP tool call object
An invocation of a tool on an MCP server.

Hide properties

arguments string
A JSON string of the arguments passed to the tool.

id string
The unique ID of the tool call.

name string
The name of the tool that was run.

server_label string
The label of the MCP server running the tool.

type string
The type of the item. Always `mcp_call`.

error string or null
The error from the tool call, if any.

output string or null
The output from the tool call.

first_id string
The ID of the first item in the list.

has_more boolean
Whether there are more items available.

last_id string
The ID of the last item in the list.

object string
The type of object returned, must be `list`.

```
OBJECT The input item list
```

```
{
  "object": "list",
  "data": [
    {
      "id": "msg_abc123",
      "type": "message",
      "role": "user",
      "content": [
        {
          "type": "input_text",
          "text": "Tell me a three sentence bedtime story about a unicorn."
        }
      ]
    }
  ],
  "first_id": "msg_abc123",
  "last_id": "msg_abc123",
  "has_more": false
}
```

Streaming

When you create a Response with `stream` set to `true`, the server will emit server-sent events to the client as the Response is generated. This section contains the events that are emitted by the server.

Learn more about streaming responses.

response.created

An event that is emitted when a response is created.

response object
The response that was created.

Hide properties

created_at number

Unix timestamp (in seconds) of when this Response was created.

error object or null

An error object returned when the model fails to generate a Response.

Hide properties

code string

The error code for the response.

message string

A human-readable description of the error.

id string

Unique identifier for this Response.

incomplete_details object or null

Details about why the response is incomplete.

Hide properties

reason string

The reason why the response is incomplete.

instructions string or array

A system (or developer) message inserted into the model's context.

When using along with `previous_response_id`, the instructions from a previous response will not be carried over to the next response. This makes it simple to swap out system (or developer) messages in new responses.

Hide possible types

string

A text input to the model, equivalent to a text input with the `developer` role.

Input item list array

A list of one or many input items to the model, containing different content types.

Hide possible types

Input message object

A message input to the model with a role indicating instruction following hierarchy. Instructions given with the `developer` or `system` role take precedence over instructions given with the `user` role. Messages with the `assistant` role are presumed to have been generated by the model in previous interactions.

Hide properties

content string or array

Text, image, or audio input to the model, used to generate a response. Can also contain previous assistant responses.

Hide possible types

Text input string

A text input to the model.

Input item content list array

A list of one or many input items to the model, containing different content types.

Hide possible types

Input text object

A text input to the model.

Hide properties

text string

The text input to the model.

type string

The type of the input item. Always `input_text`.

Input image object

An image input to the model. Learn about image inputs.

Hide properties

detail string

The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

type string

The type of the input item. Always `input_image`.

file_id string

The ID of the file to be sent to the model.

image_url string

The URL of the image to be sent to the model. A fully qualified URL or base64 encoded image in a data URL.

Input file object

A file input to the model.

Hide properties

type string

The type of the input item. Always `input_file`.

file_data string

The content of the file to be sent to the model.

file_id string

The ID of the file to be sent to the model.

file_url string

The URL of the file to be sent to the model.

filename string

The name of the file to be sent to the model.

role string

The role of the message input. One of `user`, `assistant`, `system`, or `developer`.

type string

The type of the message input. Always `message`.

Item object

An item representing part of the context for the response to be generated by the model. Can contain text, images, and audio inputs, as well as previous assistant responses and tool call outputs.

Hide possible types

Input message object

A message input to the model with a role indicating instruction following hierarchy. Instructions given with the `developer` or `system` role take precedence over instructions given with the `user` role.

Hide properties

content array

A list of one or many input items to the model, containing different content types.

Hide possible types

Input text object

A text input to the model.

Hide properties

text string

The text input to the model.

type string

The type of the input item. Always `input_text`.

Input image object

An image input to the model. Learn about image inputs.

Hide properties

detail string

The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

type string

The type of the input item. Always `input_image`.

file_id string

The ID of the file to be sent to the model.

image_url string

The URL of the image to be sent to the model. A fully qualified URL or base64 encoded image in a data URL.

Input file object

A file input to the model.

Hide properties

type string

The type of the input item. Always `input_file`.

file_data string

The content of the file to be sent to the model.

file_id string

The ID of the file to be sent to the model.

file_url string
The URL of the file to be sent to the model.

filename string
The name of the file to be sent to the model.

role string
The role of the message input. One of `user`, `system`, or `developer`.

status string
The status of item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

type string
The type of the message input. Always set to `message`.

Output message object
An output message from the model.

Hide properties

content array
The content of the output message.

Hide possible types

Output text object
A text output from the model.

Hide properties

annotations array
The annotations of the text output.

Hide possible types

File citation object
A citation to a file.

Hide properties

file_id string
The ID of the file.

filename string
The filename of the file cited.

index integer
The index of the file in the list of files.

type string
The type of the file citation. Always `file_citation` .

URL citation object
A citation for a web resource used to generate a model response.

Hide properties

end_index integer
The index of the last character of the URL citation in the message.

start_index integer
The index of the first character of the URL citation in the message.

title string
The title of the web resource.

type string
The type of the URL citation. Always `url_citation` .

url string
The URL of the web resource.

Container file citation object
A citation for a container file used to generate a model response.

Hide properties

container_id string
The ID of the container file.

end_index integer
The index of the last character of the container file citation in the message.

file_id string
The ID of the file.

filename string
The filename of the container file cited.

start_index integer
The index of the first character of the container file citation in the message.

type string
The type of the container file citation. Always `container_file_citation`.

File path object
A path to a file.

Hide properties

file_id string
The ID of the file.

index integer
The index of the file in the list of files.

type string
The type of the file path. Always `file_path`.

text string
The text output from the model.

type string
The type of the output text. Always `output_text`.

logprobs array

Hide properties

bytes array

logprob number

token string

top_logprobs array

Hide properties

bytes array

logprob number

token string

Refusal object

A refusal from the model.

Hide properties

refusal string

The refusal explanationfrom the model.

type string

The type of the refusal. Always `refusal`.

id string

The unique ID of the output message.

role string

The role of the output message. Always `assistant`.

status string

The status of the message input. One of `in_progress`, `completed`, or `incomplete`. Populated when input items are returned via API.

type string

The type of the output message. Always `message`.

File search tool call object

The results of a file search tool call. See the file search guide for more information.

Hide properties

id string

The unique ID of the file search tool call.

queries array
The queries used to search for files.

status string
The status of the file search tool call. One of `in_progress`, `searching`, `incomplete` or `failed`,

type string
The type of the file search tool call. Always `file_search_call`.

results array or null
The results of the file search tool call.

Hide properties

attributes map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard. Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters, booleans, or numbers.

file_id string
The unique ID of the file.

filename string
The name of the file.

score number
The relevance score of the file - a value between 0 and 1.

text string
The text that was retrieved from the file.

Computer tool call object
A tool call to a computer use tool. See the computer use guide for more information.

Hide properties

action object

Hide possible types

Click object

A click action.

Hide properties

button string
Indicates which mouse button was pressed during the click. One of `left`, `right`, `wheel`, `back`, or `forward`.

type string
Specifies the event type. For a click action, this property is always set to `click`.

x integer
The x-coordinate where the click occurred.

y integer
The y-coordinate where the click occurred.

DoubleClick object
A double click action.

Hide properties

type string
Specifies the event type. For a double click action, this property is always set to `double_click`.

x integer
The x-coordinate where the double click occurred.

y integer
The y-coordinate where the double click occurred.

Drag object
A drag action.

Hide properties

path array
An array of coordinates representing the path of the drag action. Coordinates will appear as an array of objects, eg

```
[
  { x: 100, y: 200 },
  { x: 200, y: 300 }
]
```

Hide properties

x integer
The x-coordinate.

y integer
The y-coordinate.

type string
Specifies the event type. For a drag action, this property is always set to `drag`.

KeyPress object
A collection of keypresses the model would like to perform.

Hide properties

keys array
The combination of keys the model is requesting to be pressed. This is an array of strings, each representing a key.

type string
Specifies the event type. For a keypress action, this property is always set to `keypress`.

Move object
A mouse move action.

Hide properties

type string
Specifies the event type. For a move action, this property is always set to `move`.

x integer
The x-coordinate to move to.

y integer
The y-coordinate to move to.

Screenshot object

A screenshot action.

Hide properties

type string

Specifies the event type. For a screenshot action, this property is always set to `screenshot`.

Scroll object

A scroll action.

Hide properties

scroll_x integer

The horizontal scroll distance.

scroll_y integer

The vertical scroll distance.

type string

Specifies the event type. For a scroll action, this property is always set to `scroll`.

x integer

The x-coordinate where the scroll occurred.

y integer

The y-coordinate where the scroll occurred.

Type object

An action to type in text.

Hide properties

text string

The text to type.

type string

Specifies the event type. For a type action, this property is always set to `type`.

Wait object

A wait action.

Hide properties

type string
Specifies the event type. For a wait action, this property is always set to `wait`.

call_id string
An identifier used when responding to the tool call with output.

id string
The unique ID of the computer call.

pending_safety_checks array
The pending safety checks for the computer call.

Hide properties

code string
The type of the pending safety check.

id string
The ID of the pending safety check.

message string
Details about the pending safety check.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

type string
The type of the computer call. Always `computer_call`.

Computer tool call output object
The output of a computer tool call.

Hide properties

call_id string
The ID of the computer tool call that produced the output.

output object
A computer screenshot image used with the computer use tool.

Hide properties

type string
Specifies the event type. For a computer screenshot, this property is always set to `computer_screenshot`.

file_id string
The identifier of an uploaded file that contains the screenshot.

image_url string
The URL of the screenshot image.

type string
The type of the computer tool call output. Always `computer_call_output`.

acknowledged_safety_checks array
The safety checks reported by the API that have been acknowledged by the developer.

Hide properties

id string
The ID of the pending safety check.

code string
The type of the pending safety check.

message string
Details about the pending safety check.

id string
The ID of the computer tool call output.

status string
The status of the message input. One of `in_progress`, `completed`, or `incomplete`. Populated when input items are returned via API.

Web search tool call object
The results of a web search tool call. See the web search guide for more information.

Hide properties

action object
An object describing the specific action taken in this web search call. Includes details on how the model used the web (search, open_page, find).

Hide possible types

Search action object

Action type "search" - Performs a web search query.

Hide properties

query string

The search query.

type string

The action type.

Open page action object

Action type "open_page" - Opens a specific URL from search results.

Hide properties

type string

The action type.

url string

The URL opened by the model.

Find action object

Action type "find": Searches for a pattern within a loaded page.

Hide properties

pattern string

The pattern or text to search for within the page.

type string

The action type.

url string

The URL of the page searched for the pattern.

id string

The unique ID of the web search tool call.

status string

The status of the web search tool call.

type string
The type of the web search tool call. Always `web_search_call`.

Function tool call object
A tool call to run a function. See the function calling guide for more information.

Hide properties

arguments string
A JSON string of the arguments to pass to the function.

call_id string
The unique ID of the function tool call generated by the model.

name string
The name of the function to run.

type string
The type of the function tool call. Always `function_call`.

id string
The unique ID of the function tool call.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Function tool call output object
The output of a function tool call.

Hide properties

call_id string
The unique ID of the function tool call generated by the model.

output string
A JSON string of the output of the function tool call.

type string
The type of the function tool call output. Always `function_call_output`.

id string
The unique ID of the function tool call output. Populated when this item is returned via

API.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Reasoning object
A description of the chain of thought used by a reasoning model while generating a response. Be sure to include these items in your `input` to the Responses API for subsequent turns of a conversation if you are manually managing context.

Hide properties

id string
The unique identifier of the reasoning content.

summary array
Reasoning text contents.

Hide properties

text string
A short summary of the reasoning used by the model when generating the response.

type string
The type of the object. Always `summary_text`.

type string
The type of the object. Always `reasoning`.

encrypted_content string or null
The encrypted content of the reasoning item - populated when a response is generated with `reasoning.encrypted_content` in the `include` parameter.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Image generation call object
An image generation request made by the model.

Hide properties

id string

The unique ID of the image generation call.

result string or null
The generated image encoded in base64.

status string
The status of the image generation call.

type string
The type of the image generation call. Always `image_generation_call`.

Code interpreter tool call object
A tool call to run code.

Hide properties

code string or null
The code to run, or null if not available.

container_id string
The ID of the container used to run the code.

id string
The unique ID of the code interpreter tool call.

outputs array or null
The outputs generated by the code interpreter, such as logs or images. Can be null if no outputs are available.

Hide possible types

Code interpreter output logs object
The logs output from the code interpreter.

Hide properties

logs string
The logs output from the code interpreter.

type string
The type of the output. Always 'logs'.

Code interpreter output image object
The image output from the code interpreter.

Hide properties

type string
The type of the output. Always 'image'.

url string
The URL of the image output from the code interpreter.

status string
The status of the code interpreter tool call.

type string
The type of the code interpreter tool call. Always `code_interpreter_call`.

Local shell call object
A tool call to run a command on the local shell.

Hide properties

action object
Execute a shell command on the server.

Hide properties

command array
The command to run.

env object
Environment variables to set for the command.

type string
The type of the local shell action. Always `exec`.

timeout_ms integer or null
Optional timeout in milliseconds for the command.

user string or null
Optional user to run the command as.

working_directory string or null
Optional working directory to run the command in.

call_id string

The unique ID of the local shell tool call generated by the model.

id string
The unique ID of the local shell call.

status string
The status of the local shell call.

type string
The type of the local shell call. Always `local_shell_call`.

Local shell call output object
The output of a local shell tool call.

Hide properties

id string
The unique ID of the local shell tool call generated by the model.

output string
A JSON string of the output of the local shell tool call.

type string
The type of the local shell tool call output. Always `local_shell_call_output`.

status string or null
The status of the item. One of `in_progress`, `completed`, or `incomplete`.

MCP list tools object
A list of tools available on an MCP server.

Hide properties

id string
The unique ID of the list.

server_label string
The label of the MCP server.

tools array
The tools available on the server.

Hide properties

input_schema object

The JSON schema describing the tool's input.

name string
The name of the tool.

annotations object or null
Additional annotations about the tool.

description string or null
The description of the tool.

type string
The type of the item. Always `mcp_list_tools`.

error string or null
Error message if the server could not list tools.

MCP approval request object
A request for human approval of a tool invocation.

Hide properties

arguments string
A JSON string of arguments for the tool.

id string
The unique ID of the approval request.

name string
The name of the tool to run.

server_label string
The label of the MCP server making the request.

type string
The type of the item. Always `mcp_approval_request`.

MCP approval response object
A response to an MCP approval request.

Hide properties

approval_request_id string

The ID of the approval request being answered.

approve boolean
Whether the request was approved.

type string
The type of the item. Always `mcp_approval_response`.

id string or null
The unique ID of the approval response

reason string or null
Optional reason for the decision.

MCP tool call object
An invocation of a tool on an MCP server.

^ Hide properties

arguments string
A JSON string of the arguments passed to the tool.

id string
The unique ID of the tool call.

name string
The name of the tool that was run.

server_label string
The label of the MCP server running the tool.

type string
The type of the item. Always `mcp_call`.

error string or null
The error from the tool call, if any.

output string or null
The output from the tool call.

Item reference object
An internal identifier for an item to reference.

Hide properties

id string
The ID of the item to reference.

type string
The type of item to reference. Always `item_reference`.

metadata map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

model string
Model ID used to generate the response, like `gpt-4o` or `o3`. OpenAI offers a wide range of models with different capabilities, performance characteristics, and price points. Refer to the model guide to browse and compare available models.

object string
The object type of this resource - always set to `response`.

output array
An array of content items generated by the model.

The length and order of items in the `output` array is dependent on the model's response.

Rather than accessing the first item in the `output` array and assuming it's an `assistant` message with the content generated by the model, you might consider using the `output_text` property where supported in SDKs.

Hide possible types

Output message object
An output message from the model.

Hide properties

content array
The content of the output message.

Hide possible types

Output text object
A text output from the model.

Hide properties

annotations array

The annotations of the text output.

Hide possible types

File citation object

A citation to a file.

Hide properties

file_id string

The ID of the file.

filename string

The filename of the file cited.

index integer

The index of the file in the list of files.

type string

The type of the file citation. Always `file_citation` .

URL citation object

A citation for a web resource used to generate a model response.

Hide properties

end_index integer

The index of the last character of the URL citation in the message.

start_index integer

The index of the first character of the URL citation in the message.

title string

The title of the web resource.

type string

The type of the URL citation. Always `url_citation` .

url string

The URL of the web resource.

Container file citation object

A citation for a container file used to generate a model response.

Hide properties

container_id string
The ID of the container file.

end_index integer
The index of the last character of the container file citation in the message.

file_id string
The ID of the file.

filename string
The filename of the container file cited.

start_index integer
The index of the first character of the container file citation in the message.

type string
The type of the container file citation. Always `container_file_citation`.

File path object
A path to a file.

Hide properties

file_id string
The ID of the file.

index integer
The index of the file in the list of files.

type string
The type of the file path. Always `file_path`.

text string
The text output from the model.

type string
The type of the output text. Always `output_text`.

logprobs array

Hide properties

bytes array

logprob number

token string

top_logprobs array

Hide properties

bytes array

logprob number

token string

Refusal object

A refusal from the model.

Hide properties

refusal string

The refusal explanationfrom the model.

type string

The type of the refusal. Always `refusal`.

id string

The unique ID of the output message.

role string

The role of the output message. Always `assistant`.

status string

The status of the message input. One of `in_progress`, `completed`, or `incomplete`. Populated when input items are returned via API.

type string

The type of the output message. Always `message`.

File search tool call object

The results of a file search tool call. See the file search guide for more information.

Hide properties

id string

The unique ID of the file search tool call.

queries array

The queries used to search for files.

status string

The status of the file search tool call. One of `in_progress`, `searching`, `incomplete` or `failed`,

type string

The type of the file search tool call. Always `file_search_call`.

results array or null

The results of the file search tool call.

Hide properties

attributes map

Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard. Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters, booleans, or numbers.

file_id string

The unique ID of the file.

filename string

The name of the file.

score number

The relevance score of the file - a value between 0 and 1.

text string

The text that was retrieved from the file.

Function tool call object

A tool call to run a function. See the function calling guide for more information.

Hide properties

arguments string
A JSON string of the arguments to pass to the function.

call_id string
The unique ID of the function tool call generated by the model.

name string
The name of the function to run.

type string
The type of the function tool call. Always `function_call`.

id string
The unique ID of the function tool call.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Web search tool call object
The results of a web search tool call. See the web search guide for more information.

Hide properties

action object
An object describing the specific action taken in this web search call. Includes details on how the model used the web (search, open_page, find).

Hide possible types

Search action object
Action type "search" - Performs a web search query.

Hide properties

query string
The search query.

type string
The action type.

Open page action object
Action type "open_page" - Opens a specific URL from search results.

Hide properties

type string
The action type.

url string
The URL opened by the model.

Find action object
Action type "find": Searches for a pattern within a loaded page.

Hide properties

pattern string
The pattern or text to search for within the page.

type string
The action type.

url string
The URL of the page searched for the pattern.

id string
The unique ID of the web search tool call.

status string
The status of the web search tool call.

type string
The type of the web search tool call. Always `web_search_call`.

Computer tool call object
A tool call to a computer use tool. See the computer use guide for more information.

Hide properties

action object

Hide possible types

Click object
A click action.

Hide properties

button string
Indicates which mouse button was pressed during the click. One of `left`, `right`, `wheel`, `back`, or `forward`.

type string
Specifies the event type. For a click action, this property is always set to `click`.

x integer
The x-coordinate where the click occurred.

y integer
The y-coordinate where the click occurred.

DoubleClick object
A double click action.

Hide properties

type string
Specifies the event type. For a double click action, this property is always set to `double_click`.

x integer
The x-coordinate where the double click occurred.

y integer
The y-coordinate where the double click occurred.

Drag object
A drag action.

Hide properties

path array
An array of coordinates representing the path of the drag action. Coordinates will appear as an array of objects, eg

```
[
  { x: 100, y: 200 },
  { x: 200, y: 300 }
]
```

Hide properties

x integer
The x-coordinate.

y integer
The y-coordinate.

type string
Specifies the event type. For a drag action, this property is always set to `drag`.

KeyPress object
A collection of keypresses the model would like to perform.

Hide properties

keys array
The combination of keys the model is requesting to be pressed. This is an array of strings, each representing a key.

type string
Specifies the event type. For a keypress action, this property is always set to `keypress`.

Move object
A mouse move action.

Hide properties

type string
Specifies the event type. For a move action, this property is always set to `move`.

x integer
The x-coordinate to move to.

y integer
The y-coordinate to move to.

Screenshot object
A screenshot action.

Hide properties

type string
Specifies the event type. For a screenshot action, this property is always set to

`screenshot`.

Scroll object

A scroll action.

Hide properties

scroll_x integer

The horizontal scroll distance.

scroll_y integer

The vertical scroll distance.

type string

Specifies the event type. For a scroll action, this property is always set to `scroll`.

x integer

The x-coordinate where the scroll occurred.

y integer

The y-coordinate where the scroll occurred.

Type object

An action to type in text.

Hide properties

text string

The text to type.

type string

Specifies the event type. For a type action, this property is always set to `type`.

Wait object

A wait action.

Hide properties

type string

Specifies the event type. For a wait action, this property is always set to `wait`.

call_id string
An identifier used when responding to the tool call with output.

id string
The unique ID of the computer call.

pending_safety_checks array
The pending safety checks for the computer call.

Hide properties

code string
The type of the pending safety check.

id string
The ID of the pending safety check.

message string
Details about the pending safety check.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

type string
The type of the computer call. Always `computer_call`.

Reasoning object
A description of the chain of thought used by a reasoning model while generating a response. Be sure to include these items in your `input` to the Responses API for subsequent turns of a conversation if you are manually managing context.

Hide properties

id string
The unique identifier of the reasoning content.

summary array
Reasoning text contents.

Hide properties

text string
A short summary of the reasoning used by the model when generating the response.

type string
The type of the object. Always `summary_text`.

type string
The type of the object. Always `reasoning`.

encrypted_content string or null
The encrypted content of the reasoning item - populated when a response is generated with `reasoning.encrypted_content` in the `include` parameter.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Image generation call object
An image generation request made by the model.

Hide properties

id string
The unique ID of the image generation call.

result string or null
The generated image encoded in base64.

status string
The status of the image generation call.

type string
The type of the image generation call. Always `image_generation_call`.

Code interpreter tool call object
A tool call to run code.

Hide properties

code string or null
The code to run, or null if not available.

container_id string
The ID of the container used to run the code.

id string

The unique ID of the code interpreter tool call.

outputs array or null

The outputs generated by the code interpreter, such as logs or images. Can be null if no outputs are available.

Hide possible types

Code interpreter output logs object

The logs output from the code interpreter.

Hide properties

logs string

The logs output from the code interpreter.

type string

The type of the output. Always 'logs'.

Code interpreter output image object

The image output from the code interpreter.

Hide properties

type string

The type of the output. Always 'image'.

url string

The URL of the image output from the code interpreter.

status string

The status of the code interpreter tool call.

type string

The type of the code interpreter tool call. Always `code_interpreter_call`.

Local shell call object

A tool call to run a command on the local shell.

Hide properties

action object

Execute a shell command on the server.

Hide properties

command array
The command to run.

env object
Environment variables to set for the command.

type string
The type of the local shell action. Always `exec`.

timeout_ms integer or null
Optional timeout in milliseconds for the command.

user string or null
Optional user to run the command as.

working_directory string or null
Optional working directory to run the command in.

call_id string
The unique ID of the local shell tool call generated by the model.

id string
The unique ID of the local shell call.

status string
The status of the local shell call.

type string
The type of the local shell call. Always `local_shell_call`.

MCP tool call object
An invocation of a tool on an MCP server.

Hide properties

arguments string
A JSON string of the arguments passed to the tool.

id string
The unique ID of the tool call.

name string

The name of the tool that was run.

server_label string
The label of the MCP server running the tool.

type string
The type of the item. Always `mcp_call`.

error string or null
The error from the tool call, if any.

output string or null
The output from the tool call.

MCP list tools object
A list of tools available on an MCP server.

Hide properties

id string
The unique ID of the list.

server_label string
The label of the MCP server.

tools array
The tools available on the server.

Hide properties

input_schema object
The JSON schema describing the tool's input.

name string
The name of the tool.

annotations object or null
Additional annotations about the tool.

description string or null
The description of the tool.

type string
The type of the item. Always `mcp_list_tools`.

error string or null
Error message if the server could not list tools.

MCP approval request object
A request for human approval of a tool invocation.

Hide properties

arguments string
A JSON string of arguments for the tool.

id string
The unique ID of the approval request.

name string
The name of the tool to run.

server_label string
The label of the MCP server making the request.

type string
The type of the item. Always `mcp_approval_request`.

parallel_tool_calls boolean
Whether to allow the model to run tool calls in parallel.

temperature number or null
What sampling temperature to use, between 0 and 2. Higher values like 0.8 will make the output more random, while lower values like 0.2 will make it more focused and deterministic. We generally recommend altering this or `top_p` but not both.

tool_choice string or object
How the model should select which tool (or tools) to use when generating a response. See the `tools` parameter to see how to specify which tools the model can call.

Hide possible types

Tool choice mode string
Controls which (if any) tool is called by the model.

`none` means the model will not call any tool and instead generates a message.

`auto` means the model can pick between generating a message or calling one or more tools.

`required` means the model must call one or more tools.

Hosted tool object

Indicates that the model should use a built-in tool to generate a response. Learn more about built-in tools.

Hide properties

type string

The type of hosted tool the model should to use. Learn more about built-in tools.

Allowed values are:

- `file_search`
- `web_search_preview`
- `computer_use_preview`
- `code_interpreter`
- `image_generation`

Function tool object

Use this option to force the model to call a specific function.

Hide properties

name string

The name of the function to call.

type string

For function calling, the type is always `function`.

MCP tool object

Use this option to force the model to call a specific tool on a remote MCP server.

Hide properties

server_label string

The label of the MCP server to use.

type string

For MCP tools, the type is always `mcp`.

name string or null
The name of the tool to call on the server.

tools array
An array of tools the model may call while generating a response. You can specify which tool to use by setting the `tool_choice` parameter.

The two categories of tools you can provide the model are:

Built-in tools: Tools that are provided by OpenAI that extend the model's capabilities, like web search or file search. Learn more about built-in tools.

Function calls (custom tools): Functions that are defined by you, enabling the model to call your own code. Learn more about function calling.

Hide possible types

Function object
Defines a function in your own code the model can choose to call. Learn more about function calling.

Hide properties

name string
The name of the function to call.

parameters object
A JSON schema object describing the parameters of the function.

strict boolean
Whether to enforce strict parameter validation. Default `true`.

type string
The type of the function tool. Always `function`.

description string
A description of the function. Used by the model to determine whether or not to call the function.

File search object
A tool that searches for relevant content from uploaded files. Learn more about the file search tool.

Hide properties

type string
The type of the file search tool. Always `file_search`.

vector_store_ids array
The IDs of the vector stores to search.

filters object
A filter to apply.

Hide possible types

Comparison Filter object
A filter used to compare a specified attribute key to a given value using a defined comparison operation.

Hide properties

key string
The key to compare against the value.

type string
Specifies the comparison operator: `eq`, `ne`, `gt`, `gte`, `lt`, `lte`.

- `eq`: equals
- `ne`: not equal
- `gt`: greater than
- `gte`: greater than or equal
- `lt`: less than
- `lte`: less than or equal

value string / number / boolean
The value to compare against the attribute key; supports string, number, or boolean types.

Compound Filter object
Combine multiple filters using `and` or `or`.

Hide properties

filters array
Array of filters to combine. Items can be `ComparisonFilter` or `CompoundFilter`.

Hide possible types

Comparison Filter object
A filter used to compare a specified attribute key to a given value using a defined comparison operation.

Hide properties

key string
The key to compare against the value.

type string
Specifies the comparison operator: `eq`, `ne`, `gt`, `gte`, `lt`, `lte`.

- `eq`: equals
- `ne`: not equal
- `gt`: greater than
- `gte`: greater than or equal
- `lt`: less than
- `lte`: less than or equal

value string / number / boolean
The value to compare against the attribute key; supports string, number, or boolean types.

type string
Type of operation: `and` or `or`.

max_num_results integer
The maximum number of results to return. This number should be between 1 and 50 inclusive.

ranking_options object
Ranking options for search.

Hide properties

ranker string
The ranker to use for the file search.

score_threshold number
The score threshold for the file search, a number between 0 and 1. Numbers closer to 1 will attempt to return only the most relevant results, but may return fewer results.

Web search preview object

This tool searches the web for relevant results to use in a response. Learn more about the web search tool.

Hide properties

type string
The type of the web search tool. One of `web_search_preview` or `web_search_preview_2025_03_11`.

search_context_size string
High level guidance for the amount of context window space to use for the search. One of `low`, `medium`, or `high`. `medium` is the default.

user_location object
The user's location.

Hide properties

type string
The type of location approximation. Always `approximate`.

city string
Free text input for the city of the user, e.g. `San Francisco`.

country string
The two-letter ISO country code of the user, e.g. `US`.

region string
Free text input for the region of the user, e.g. `California`.

timezone string
The IANA timezone of the user, e.g. `America/Los_Angeles`.

Computer use preview object
A tool that controls a virtual computer. Learn more about the computer tool.

Hide properties

display_height integer
The height of the computer display.

display_width integer
The width of the computer display.

environment string

The type of computer environment to control.

type string
The type of the computer use tool. Always `computer_use_preview`.

MCP tool object
Give the model access to additional tools via remote Model Context Protocol (MCP) servers. Learn more about MCP.

Hide properties

server_label string
A label for this MCP server, used to identify it in tool calls.

server_url string
The URL for the MCP server.

type string
The type of the MCP tool. Always `mcp`.

allowed_tools array or object
List of allowed tool names or a filter object.

Hide possible types

MCP allowed tools array
A string array of allowed tool names

MCP allowed tools filter object
A filter object to specify which tools are allowed.

Hide properties

tool_names array
List of allowed tool names.

headers object or null
Optional HTTP headers to send to the MCP server. Use for authentication or other purposes.

require_approval object or string
Specify which of the MCP server's tools require approval.

Hide possible types

MCP tool approval filter object

Hide properties

always object
A list of tools that always require approval.

Hide properties

tool_names array
List of tools that require approval.

never object
A list of tools that never require approval.

Hide properties

tool_names array
List of tools that do not require approval.

MCP tool approval setting string
Specify a single approval policy for all tools. One of `always` or `never`. When set to `always`, all tools will require approval. When set to `never`, all tools will not require approval.

server_description string
Optional description of the MCP server, used to provide more context.

Code interpreter object
A tool that runs Python code to help generate a response to a prompt.

Hide properties

container string or object
The code interpreter container. Can be a container ID or an object that specifies uploaded file IDs to make available to your code.

Hide possible types

string
The container ID.

CodeInterpreterContainerAuto object
Configuration for a code interpreter container. Optionally specify the IDs of the files to run the code on.

Hide properties

type string
Always `auto`.

file_ids array
An optional list of uploaded files to make available to your code.

type string
The type of the code interpreter tool. Always `code_interpreter`.

Image generation tool object
A tool that generates images using a model like `gpt-image-1`.

Hide properties

type string
The type of the image generation tool. Always `image_generation`.

background string
Background type for the generated image. One of `transparent`, `opaque`, or `auto`. Default: `auto`.

input_image_mask object
Optional mask for inpainting. Contains `image_url` (string, optional) and `file_id` (string, optional).

Hide properties

file_id string
File ID for the mask image.

image_url string
Base64-encoded mask image.

model string
The image generation model to use. Default: `gpt-image-1`.

moderation string
Moderation level for the generated image. Default: `auto`.

output_compression integer
Compression level for the output image. Default: 100.

output_format string
The output format of the generated image. One of `png`, `webp`, or `jpeg`. Default: `png`.

partial_images integer
Number of partial images to generate in streaming mode, from 0 (default value) to 3.

quality string
The quality of the generated image. One of `low`, `medium`, `high`, or `auto`. Default: `auto`.

size string
The size of the generated image. One of `1024x1024`, `1024x1536`, `1536x1024`, or `auto`. Default: `auto`.

Local shell tool object
A tool that allows the model to execute shell commands in a local environment.

Hide properties

type string
The type of the local shell tool. Always `local_shell`.

top_p number or null
An alternative to sampling with temperature, called nucleus sampling, where the model considers the results of the tokens with top_p probability mass. So 0.1 means only the tokens comprising the top 10% probability mass are considered.

We generally recommend altering this or `temperature` but not both.

background boolean or null
Whether to run the model response in the background. Learn more.

max_output_tokens integer or null
An upper bound for the number of tokens that can be generated for a response, including visible output tokens and reasoning tokens.

max_tool_calls integer or null
The maximum number of total calls to built-in tools that can be processed in a response. This maximum number applies across all built-in tool calls, not per individual tool. Any further attempts to call a tool by the model will be ignored.

output_text string or null SDK Only
SDK-only convenience property that contains the aggregated text output from all `output_text` items

in the `output` array, if any are present. Supported in the Python and JavaScript SDKs.

`previous_response_id` string or null
The unique ID of the previous response to the model. Use this to create multi-turn conversations. Learn more about conversation state.

`prompt` object or null
Reference to a prompt template and its variables. Learn more.

Hide properties

`id` string
The unique identifier of the prompt template to use.

`variables` map
Optional map of values to substitute in for variables in your prompt. The substitution values can either be strings, or other Response input types like images or files.

`version` string or null
Optional version of the prompt template.

`reasoning` object or null
o-series models only

Configuration options for reasoning models.

Hide properties

`effort` string or null
o-series models only

Constrains effort on reasoning for reasoning models. Currently supported values are `low`, `medium`, and `high`. Reducing reasoning effort can result in faster responses and fewer tokens used on reasoning in a response.

`generate_summary` Deprecated string or null
Deprecated: use `summary` instead.

A summary of the reasoning performed by the model. This can be useful for debugging and understanding the model's reasoning process. One of `auto`, `concise`, or `detailed`.

`summary` string or null
A summary of the reasoning performed by the model. This can be useful for debugging and understanding the model's reasoning process. One of `auto`, `concise`, or `detailed`.

service_tier string or null

Specifies the processing type used for serving the request.

If set to 'auto', then the request will be processed with the service tier configured in the Project settings. Unless otherwise configured, the Project will use 'default'.

If set to 'default', then the requset will be processed with the standard pricing and performance for the selected model.

If set to 'flex' or 'priority', then the request will be processed with the corresponding service tier. Contact sales to learn more about Priority processing.

When not set, the default behavior is 'auto'.

When the `service_tier` parameter is set, the response body will include the `service_tier` value based on the processing mode actually used to serve the request. This response value may be different from the value set in the parameter.

status string

The status of the response generation. One of `completed`, `failed`, `in_progress`, `cancelled`, `queued`, or `incomplete`.

text object

Configuration options for a text response from the model. Can be plain text or structured JSON data. Learn more:

Text inputs and outputs

Structured Outputs

Hide properties

format object

An object specifying the format that the model must output.

Configuring `{ "type": "json_schema" }` enables Structured Outputs, which ensures the model will match your supplied JSON schema. Learn more in the Structured Outputs guide.

The default format is `{ "type": "text" }` with no additional options.

Not recommended for gpt-4o and newer models:

Setting to `{ "type": "json_object" }` enables the older JSON mode, which ensures the message the model generates is valid JSON. Using `json_schema` is preferred for models that support it.

Hide possible types

Text object

Default response format. Used to generate text responses.

Hide properties

type string
The type of response format being defined. Always `text`.

JSON schema object
JSON Schema response format. Used to generate structured JSON responses. Learn more about Structured Outputs.

Hide properties

name string
The name of the response format. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

schema object
The schema for the response format, described as a JSON Schema object. Learn how to build JSON schemas here.

type string
The type of response format being defined. Always `json_schema`.

description string
A description of what the response format is for, used by the model to determine how to respond in the format.

strict boolean or null
Whether to enable strict schema adherence when generating the output. If set to true, the model will always follow the exact schema defined in the `schema` field. Only a subset of JSON Schema is supported when `strict` is `true`. To learn more, read the Structured Outputs guide.

JSON object object
JSON object response format. An older method of generating JSON responses. Using `json_schema` is recommended for models that support it. Note that the model will not generate JSON without a system or user message instructing it to do so.

Hide properties

type string
The type of response format being defined. Always `json_object`.

top_logprobs integer or null

An integer between 0 and 20 specifying the number of most likely tokens to return at each token position, each with an associated log probability.

`truncation` string or null

The truncation strategy to use for the model response.

`auto` : If the context of this response and previous ones exceeds the model's context window size, the model will truncate the response to fit the context window by dropping input items in the middle of the conversation.

`disabled` (default): If a model response will exceed the context window size for a model, the request will fail with a 400 error.

`usage` object

Represents token usage details including input tokens, output tokens, a breakdown of output tokens, and the total tokens used.

Hide properties

`input_tokens` integer

The number of input tokens.

`input_tokens_details` object

A detailed breakdown of the input tokens.

Hide properties

`cached_tokens` integer

The number of tokens that were retrieved from the cache. More on prompt caching.

`output_tokens` integer

The number of output tokens.

`output_tokens_details` object

A detailed breakdown of the output tokens.

Hide properties

`reasoning_tokens` integer

The number of reasoning tokens.

`total_tokens` integer

The total number of tokens used.

`user` string

A stable identifier for your end-users. Used to boost cache hit rates by better bucketing similar requests and to help OpenAI detect and prevent abuse. Learn more.

sequence_number integer

The sequence number for this event.

type string

The type of the event. Always `response.created`.

OBJECT response.created

```
{
  "type": "response.created",
  "response": {
    "id": "resp_67ccfcdd16748190a91872c75d38539e09e4d4aac714747c",
    "object": "response",
    "created_at": 1741487325,
    "status": "in_progress",
    "error": null,
    "incomplete_details": null,
    "instructions": null,
    "max_output_tokens": null,
    "model": "gpt-4o-2024-08-06",
    "output": [],
    "parallel_tool_calls": true,
    "previous_response_id": null,
    "reasoning": {
      "effort": null,
      "summary": null
    },
    "store": true,
    "temperature": 1,
    "text": {
      "format": {
        "type": "text"
      }
    },
    "tool_choice": "auto",
    "tools": [],
    "top_p": 1,
    "truncation": "disabled",
    "usage": null,
    "user": null,
    "metadata": {}
```

```
35    },
36    "sequence_number": 1
}
```

response.in_progress

Emitted when the response is in progress.

response object
The response that is in progress.

Hide properties

created_at number
Unix timestamp (in seconds) of when this Response was created.

error object or null
An error object returned when the model fails to generate a Response.

Hide properties

code string
The error code for the response.

message string
A human-readable description of the error.

id string
Unique identifier for this Response.

incomplete_details object or null
Details about why the response is incomplete.

Hide properties

reason string
The reason why the response is incomplete.

instructions string or array
A system (or developer) message inserted into the model's context.

When using along with `previous_response_id`, the instructions from a previous response will not be carried over to the next response. This makes it simple to swap out system (or developer) messages in

new responses.

Hide possible types

string
A text input to the model, equivalent to a text input with the `developer` role.

Input item list array
A list of one or many input items to the model, containing different content types.

Hide possible types

Input message object
A message input to the model with a role indicating instruction following hierarchy. Instructions given with the `developer` or `system` role take precedence over instructions given with the `user` role. Messages with the `assistant` role are presumed to have been generated by the model in previous interactions.

Hide properties

content string or array
Text, image, or audio input to the model, used to generate a response. Can also contain previous assistant responses.

Hide possible types

Text input string
A text input to the model.

Input item content list array
A list of one or many input items to the model, containing different content types.

Hide possible types

Input text object
A text input to the model.

Hide properties

text string
The text input to the model.

type string
The type of the input item. Always `input_text`.

Input image object
An image input to the model. Learn about image inputs.

Hide properties

detail string
The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

type string
The type of the input item. Always `input_image`.

file_id string
The ID of the file to be sent to the model.

image_url string
The URL of the image to be sent to the model. A fully qualified URL or base64 encoded image in a data URL.

Input file object
A file input to the model.

Hide properties

type string
The type of the input item. Always `input_file`.

file_data string
The content of the file to be sent to the model.

file_id string
The ID of the file to be sent to the model.

file_url string
The URL of the file to be sent to the model.

filename string
The name of the file to be sent to the model.

role string
The role of the message input. One of `user`, `assistant`, `system`, or `developer`.

type string
The type of the message input. Always `message`.

Item object

An item representing part of the context for the response to be generated by the model. Can contain text, images, and audio inputs, as well as previous assistant responses and tool call outputs.

Hide possible types

Input message object

A message input to the model with a role indicating instruction following hierarchy. Instructions given with the `developer` or `system` role take precedence over instructions given with the `user` role.

Hide properties

content array

A list of one or many input items to the model, containing different content types.

Hide possible types

Input text object

A text input to the model.

Hide properties

text string

The text input to the model.

type string

The type of the input item. Always `input_text`.

Input image object

An image input to the model. Learn about image inputs.

Hide properties

detail string

The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

type string

The type of the input item. Always `input_image`.

file_id string

The ID of the file to be sent to the model.

image_url string

The URL of the image to be sent to the model. A fully qualified URL or base64

encoded image in a data URL.

Input file object
A file input to the model.

Hide properties

type string
The type of the input item. Always `input_file`.

file_data string
The content of the file to be sent to the model.

file_id string
The ID of the file to be sent to the model.

file_url string
The URL of the file to be sent to the model.

filename string
The name of the file to be sent to the model.

role string
The role of the message input. One of `user`, `system`, or `developer`.

status string
The status of item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

type string
The type of the message input. Always set to `message`.

Output message object
An output message from the model.

Hide properties

content array
The content of the output message.

Hide possible types

Output text object
A text output from the model.

Hide properties

annotations array

The annotations of the text output.

Hide possible types

File citation object

A citation to a file.

Hide properties

file_id string

The ID of the file.

filename string

The filename of the file cited.

index integer

The index of the file in the list of files.

type string

The type of the file citation. Always `file_citation`.

URL citation object

A citation for a web resource used to generate a model response.

Hide properties

end_index integer

The index of the last character of the URL citation in the message.

start_index integer

The index of the first character of the URL citation in the message.

title string

The title of the web resource.

type string

The type of the URL citation. Always `url_citation`.

url string

The URL of the web resource.

Container file citation object

A citation for a container file used to generate a model response.

Hide properties

container_id string
The ID of the container file.

end_index integer
The index of the last character of the container file citation in the message.

file_id string
The ID of the file.

filename string
The filename of the container file cited.

start_index integer
The index of the first character of the container file citation in the message.

type string
The type of the container file citation. Always `container_file_citation`.

File path object
A path to a file.

Hide properties

file_id string
The ID of the file.

index integer
The index of the file in the list of files.

type string
The type of the file path. Always `file_path`.

text string
The text output from the model.

type string

The type of the output text. Always `output_text`.

logprobs array

Hide properties

bytes array

logprob number

token string

top_logprobs array

Hide properties

bytes array

logprob number

token string

Refusal object

A refusal from the model.

Hide properties

refusal string

The refusal explanationfrom the model.

type string

The type of the refusal. Always `refusal`.

id string

The unique ID of the output message.

role string

The role of the output message. Always `assistant`.

status string

The status of the message input. One of `in_progress`, `completed`, or `incomplete`. Populated when input items are returned via API.

type string
The type of the output message. Always `message`.

File search tool call object
The results of a file search tool call. See the file search guide for more information.

Hide properties

id string
The unique ID of the file search tool call.

queries array
The queries used to search for files.

status string
The status of the file search tool call. One of `in_progress`, `searching`, `incomplete` or `failed`,

type string
The type of the file search tool call. Always `file_search_call`.

results array or null
The results of the file search tool call.

Hide properties

attributes map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard. Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters, booleans, or numbers.

file_id string
The unique ID of the file.

filename string
The name of the file.

score number
The relevance score of the file - a value between 0 and 1.

text string
The text that was retrieved from the file.

Computer tool call object
A tool call to a computer use tool. See the computer use guide for more information.

Hide properties

action object

Hide possible types

Click object
A click action.

Hide properties

button string
Indicates which mouse button was pressed during the click. One of `left`, `right`, `wheel`, `back`, or `forward`.

type string
Specifies the event type. For a click action, this property is always set to `click`.

x integer
The x-coordinate where the click occurred.

y integer
The y-coordinate where the click occurred.

DoubleClick object
A double click action.

Hide properties

type string
Specifies the event type. For a double click action, this property is always set to `double_click`.

x integer
The x-coordinate where the double click occurred.

y integer
The y-coordinate where the double click occurred.

Drag object

A drag action.

Hide properties

path array

An array of coordinates representing the path of the drag action. Coordinates will appear as an array of objects, eg

```
[
  { x: 100, y: 200 },
  { x: 200, y: 300 }
]
```

Hide properties

x integer

The x-coordinate.

y integer

The y-coordinate.

type string

Specifies the event type. For a drag action, this property is always set to `drag`.

KeyPress object

A collection of keypresses the model would like to perform.

Hide properties

keys array

The combination of keys the model is requesting to be pressed. This is an array of strings, each representing a key.

type string

Specifies the event type. For a keypress action, this property is always set to `keypress`.

Move object

A mouse move action.

Hide properties

type string
Specifies the event type. For a move action, this property is always set to `move`.

x integer
The x-coordinate to move to.

y integer
The y-coordinate to move to.

Screenshot object
A screenshot action.

Hide properties

type string
Specifies the event type. For a screenshot action, this property is always set to `screenshot`.

Scroll object
A scroll action.

Hide properties

scroll_x integer
The horizontal scroll distance.

scroll_y integer
The vertical scroll distance.

type string
Specifies the event type. For a scroll action, this property is always set to `scroll`.

x integer
The x-coordinate where the scroll occurred.

y integer
The y-coordinate where the scroll occurred.

Type object
An action to type in text.

Hide properties

text string
The text to type.

type string
Specifies the event type. For a type action, this property is always set to `type`.

Wait object
A wait action.

Hide properties

type string
Specifies the event type. For a wait action, this property is always set to `wait`.

call_id string
An identifier used when responding to the tool call with output.

id string
The unique ID of the computer call.

pending_safety_checks array
The pending safety checks for the computer call.

Hide properties

code string
The type of the pending safety check.

id string
The ID of the pending safety check.

message string
Details about the pending safety check.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

type string
The type of the computer call. Always `computer_call`.

Computer tool call output object

The output of a computer tool call.

Hide properties

call_id string

The ID of the computer tool call that produced the output.

output object

A computer screenshot image used with the computer use tool.

Hide properties

type string

Specifies the event type. For a computer screenshot, this property is always set to `computer_screenshot`.

file_id string

The identifier of an uploaded file that contains the screenshot.

image_url string

The URL of the screenshot image.

type string

The type of the computer tool call output. Always `computer_call_output`.

acknowledged_safety_checks array

The safety checks reported by the API that have been acknowledged by the developer.

Hide properties

id string

The ID of the pending safety check.

code string

The type of the pending safety check.

message string

Details about the pending safety check.

id string

The ID of the computer tool call output.

status string

The status of the message input. One of `in_progress`, `completed`, or `incomplete`. Populated when input items are returned via API.

Web search tool call object

The results of a web search tool call. See the web search guide for more information.

Hide properties

action object

An object describing the specific action taken in this web search call. Includes details on how the model used the web (search, open_page, find).

Hide possible types

Search action object

Action type "search" - Performs a web search query.

Hide properties

query string

The search query.

type string

The action type.

Open page action object

Action type "open_page" - Opens a specific URL from search results.

Hide properties

type string

The action type.

url string

The URL opened by the model.

Find action object

Action type "find": Searches for a pattern within a loaded page.

Hide properties

pattern string

The pattern or text to search for within the page.

type string

The action type.

url string
The URL of the page searched for the pattern.

id string
The unique ID of the web search tool call.

status string
The status of the web search tool call.

type string
The type of the web search tool call. Always `web_search_call`.

Function tool call object
A tool call to run a function. See the function calling guide for more information.

Hide properties

arguments string
A JSON string of the arguments to pass to the function.

call_id string
The unique ID of the function tool call generated by the model.

name string
The name of the function to run.

type string
The type of the function tool call. Always `function_call`.

id string
The unique ID of the function tool call.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Function tool call output object
The output of a function tool call.

Hide properties

call_id string

The unique ID of the function tool call generated by the model.

output string
A JSON string of the output of the function tool call.

type string
The type of the function tool call output. Always `function_call_output`.

id string
The unique ID of the function tool call output. Populated when this item is returned via API.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Reasoning object
A description of the chain of thought used by a reasoning model while generating a response. Be sure to include these items in your `input` to the Responses API for subsequent turns of a conversation if you are manually managing context.

Hide properties

id string
The unique identifier of the reasoning content.

summary array
Reasoning text contents.

Hide properties

text string
A short summary of the reasoning used by the model when generating the response.

type string
The type of the object. Always `summary_text`.

type string
The type of the object. Always `reasoning`.

encrypted_content string or null
The encrypted content of the reasoning item - populated when a response is generated with `reasoning.encrypted_content` in the `include` parameter.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Image generation call object
An image generation request made by the model.

Hide properties

id string
The unique ID of the image generation call.

result string or null
The generated image encoded in base64.

status string
The status of the image generation call.

type string
The type of the image generation call. Always `image_generation_call`.

Code interpreter tool call object
A tool call to run code.

Hide properties

code string or null
The code to run, or null if not available.

container_id string
The ID of the container used to run the code.

id string
The unique ID of the code interpreter tool call.

outputs array or null
The outputs generated by the code interpreter, such as logs or images. Can be null if no outputs are available.

Hide possible types

Code interpreter output logs object
The logs output from the code interpreter.

Hide properties

logs string
The logs output from the code interpreter.

type string
The type of the output. Always 'logs'.

Code interpreter output image object
The image output from the code interpreter.

Hide properties

type string
The type of the output. Always 'image'.

url string
The URL of the image output from the code interpreter.

status string
The status of the code interpreter tool call.

type string
The type of the code interpreter tool call. Always `code_interpreter_call`.

Local shell call object
A tool call to run a command on the local shell.

Hide properties

action object
Execute a shell command on the server.

Hide properties

command array
The command to run.

env object
Environment variables to set for the command.

type string
The type of the local shell action. Always `exec`.

timeout_ms integer or null

Optional timeout in milliseconds for the command.

user string or null
Optional user to run the command as.

working_directory string or null
Optional working directory to run the command in.

call_id string
The unique ID of the local shell tool call generated by the model.

id string
The unique ID of the local shell call.

status string
The status of the local shell call.

type string
The type of the local shell call. Always `local_shell_call`.

Local shell call output object
The output of a local shell tool call.

Hide properties

id string
The unique ID of the local shell tool call generated by the model.

output string
A JSON string of the output of the local shell tool call.

type string
The type of the local shell tool call output. Always `local_shell_call_output`.

status string or null
The status of the item. One of `in_progress`, `completed`, or `incomplete`.

MCP list tools object
A list of tools available on an MCP server.

Hide properties

id string

The unique ID of the list.

server_label string
The label of the MCP server.

tools array
The tools available on the server.

Hide properties

input_schema object
The JSON schema describing the tool's input.

name string
The name of the tool.

annotations object or null
Additional annotations about the tool.

description string or null
The description of the tool.

type string
The type of the item. Always `mcp_list_tools`.

error string or null
Error message if the server could not list tools.

MCP approval request object
A request for human approval of a tool invocation.

Hide properties

arguments string
A JSON string of arguments for the tool.

id string
The unique ID of the approval request.

name string
The name of the tool to run.

server_label string
The label of the MCP server making the request.

type string
The type of the item. Always `mcp_approval_request`.

MCP approval response object
A response to an MCP approval request.

Hide properties

approval_request_id string
The ID of the approval request being answered.

approve boolean
Whether the request was approved.

type string
The type of the item. Always `mcp_approval_response`.

id string or null
The unique ID of the approval response

reason string or null
Optional reason for the decision.

MCP tool call object
An invocation of a tool on an MCP server.

Hide properties

arguments string
A JSON string of the arguments passed to the tool.

id string
The unique ID of the tool call.

name string
The name of the tool that was run.

server_label string
The label of the MCP server running the tool.

type string
The type of the item. Always `mcp_call`.

error string or null
The error from the tool call, if any.

output string or null
The output from the tool call.

Item reference object
An internal identifier for an item to reference.

Hide properties

id string
The ID of the item to reference.

type string
The type of item to reference. Always `item_reference`.

metadata map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

model string
Model ID used to generate the response, like `gpt-4o` or `o3`. OpenAI offers a wide range of models with different capabilities, performance characteristics, and price points. Refer to the model guide to browse and compare available models.

object string
The object type of this resource - always set to `response`.

output array
An array of content items generated by the model.

The length and order of items in the `output` array is dependent on the model's response.

Rather than accessing the first item in the `output` array and assuming it's an `assistant` message with the content generated by the model, you might consider using the `output_text`

property where supported in SDKs.

Hide possible types

Output message object

An output message from the model.

Hide properties

content array

The content of the output message.

Hide possible types

Output text object

A text output from the model.

Hide properties

annotations array

The annotations of the text output.

Hide possible types

File citation object

A citation to a file.

Hide properties

file_id string

The ID of the file.

filename string

The filename of the file cited.

index integer

The index of the file in the list of files.

type string

The type of the file citation. Always `file_citation`.

URL citation object

A citation for a web resource used to generate a model response.

Hide properties

end_index integer

The index of the last character of the URL citation in the message.

start_index integer
The index of the first character of the URL citation in the message.

title string
The title of the web resource.

type string
The type of the URL citation. Always `url_citation`.

url string
The URL of the web resource.

Container file citation object
A citation for a container file used to generate a model response.

Hide properties

container_id string
The ID of the container file.

end_index integer
The index of the last character of the container file citation in the message.

file_id string
The ID of the file.

filename string
The filename of the container file cited.

start_index integer
The index of the first character of the container file citation in the message.

type string
The type of the container file citation. Always `container_file_citation`.

File path object
A path to a file.

Hide properties

file_id string
The ID of the file.

index integer
The index of the file in the list of files.

type string
The type of the file path. Always `file_path` .

text string
The text output from the model.

type string
The type of the output text. Always `output_text` .

logprobs array
Hide properties

bytes array

logprob number

token string

top_logprobs array
Hide properties

bytes array

logprob number

token string

Refusal object
A refusal from the model.
Hide properties

refusal string
The refusal explanationfrom the model.

type string
The type of the refusal. Always `refusal` .

id string
The unique ID of the output message.

role string
The role of the output message. Always `assistant`.

status string
The status of the message input. One of `in_progress`, `completed`, or `incomplete`. Populated when input items are returned via API.

type string
The type of the output message. Always `message`.

File search tool call object
The results of a file search tool call. See the file search guide for more information.

Hide properties

id string
The unique ID of the file search tool call.

queries array
The queries used to search for files.

status string
The status of the file search tool call. One of `in_progress`, `searching`, `incomplete` or `failed`,

type string
The type of the file search tool call. Always `file_search_call`.

results array or null
The results of the file search tool call.

Hide properties

attributes map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard. Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters, booleans, or numbers.

file_id string

The unique ID of the file.

filename string
The name of the file.

score number
The relevance score of the file - a value between 0 and 1.

text string
The text that was retrieved from the file.

Function tool call object
A tool call to run a function. See the function calling guide for more information.

Hide properties

arguments string
A JSON string of the arguments to pass to the function.

call_id string
The unique ID of the function tool call generated by the model.

name string
The name of the function to run.

type string
The type of the function tool call. Always `function_call`.

id string
The unique ID of the function tool call.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Web search tool call object
The results of a web search tool call. See the web search guide for more information.

Hide properties

action object
An object describing the specific action taken in this web search call. Includes details on how the model used the web (search, open_page, find).

Hide possible types

Search action object
Action type "search" - Performs a web search query.

Hide properties

query string
The search query.

type string
The action type.

Open page action object
Action type "open_page" - Opens a specific URL from search results.

Hide properties

type string
The action type.

url string
The URL opened by the model.

Find action object
Action type "find": Searches for a pattern within a loaded page.

Hide properties

pattern string
The pattern or text to search for within the page.

type string
The action type.

url string
The URL of the page searched for the pattern.

id string
The unique ID of the web search tool call.

status string
The status of the web search tool call.

type string
The type of the web search tool call. Always `web_search_call`.

Computer tool call object
A tool call to a computer use tool. See the computer use guide for more information.

Hide properties

action object

Hide possible types

Click object
A click action.

Hide properties

button string
Indicates which mouse button was pressed during the click. One of `left`, `right`, `wheel`, `back`, or `forward`.

type string
Specifies the event type. For a click action, this property is always set to `click`.

x integer
The x-coordinate where the click occurred.

y integer
The y-coordinate where the click occurred.

DoubleClick object
A double click action.

Hide properties

type string
Specifies the event type. For a double click action, this property is always set to `double_click`.

x integer
The x-coordinate where the double click occurred.

y integer
The y-coordinate where the double click occurred.

Drag object

A drag action.

Hide properties

path array

An array of coordinates representing the path of the drag action. Coordinates will appear as an array of objects, eg

```
[
  { x: 100, y: 200 },
  { x: 200, y: 300 }
]
```

Hide properties

x integer

The x-coordinate.

y integer

The y-coordinate.

type string

Specifies the event type. For a drag action, this property is always set to `drag`.

KeyPress object

A collection of keypresses the model would like to perform.

Hide properties

keys array

The combination of keys the model is requesting to be pressed. This is an array of strings, each representing a key.

type string

Specifies the event type. For a keypress action, this property is always set to `keypress`.

Move object

A mouse move action.

Hide properties

type string

Specifies the event type. For a move action, this property is always set to `move`.

x integer
The x-coordinate to move to.

y integer
The y-coordinate to move to.

Screenshot object
A screenshot action.

Hide properties

type string
Specifies the event type. For a screenshot action, this property is always set to `screenshot`.

Scroll object
A scroll action.

Hide properties

scroll_x integer
The horizontal scroll distance.

scroll_y integer
The vertical scroll distance.

type string
Specifies the event type. For a scroll action, this property is always set to `scroll`.

x integer
The x-coordinate where the scroll occurred.

y integer
The y-coordinate where the scroll occurred.

Type object
An action to type in text.

Hide properties

text string
The text to type.

type string
Specifies the event type. For a type action, this property is always set to `type`.

Wait object
A wait action.

Hide properties

type string
Specifies the event type. For a wait action, this property is always set to `wait`.

call_id string
An identifier used when responding to the tool call with output.

id string
The unique ID of the computer call.

pending_safety_checks array
The pending safety checks for the computer call.

Hide properties

code string
The type of the pending safety check.

id string
The ID of the pending safety check.

message string
Details about the pending safety check.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

type string
The type of the computer call. Always `computer_call`.

Reasoning object
A description of the chain of thought used by a reasoning model while generating a response. Be sure to include these items in your `input` to the Responses API for subsequent turns of a

conversation if you are manually managing context.

Hide properties

id string
The unique identifier of the reasoning content.

summary array
Reasoning text contents.

Hide properties

text string
A short summary of the reasoning used by the model when generating the response.

type string
The type of the object. Always `summary_text`.

type string
The type of the object. Always `reasoning`.

encrypted_content string or null
The encrypted content of the reasoning item - populated when a response is generated with `reasoning.encrypted_content` in the `include` parameter.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Image generation call object
An image generation request made by the model.

Hide properties

id string
The unique ID of the image generation call.

result string or null
The generated image encoded in base64.

status string
The status of the image generation call.

type string
The type of the image generation call. Always `image_generation_call`.

Code interpreter tool call object
A tool call to run code.

Hide properties

code string or null
The code to run, or null if not available.

container_id string
The ID of the container used to run the code.

id string
The unique ID of the code interpreter tool call.

outputs array or null
The outputs generated by the code interpreter, such as logs or images. Can be null if no outputs are available.

Hide possible types

Code interpreter output logs object
The logs output from the code interpreter.

Hide properties

logs string
The logs output from the code interpreter.

type string
The type of the output. Always 'logs'.

Code interpreter output image object
The image output from the code interpreter.

Hide properties

type string
The type of the output. Always 'image'.

url string
The URL of the image output from the code interpreter.

status string
The status of the code interpreter tool call.

type string
The type of the code interpreter tool call. Always `code_interpreter_call`.

Local shell call object
A tool call to run a command on the local shell.

Hide properties

action object
Execute a shell command on the server.

Hide properties

command array
The command to run.

env object
Environment variables to set for the command.

type string
The type of the local shell action. Always `exec`.

timeout_ms integer or null
Optional timeout in milliseconds for the command.

user string or null
Optional user to run the command as.

working_directory string or null
Optional working directory to run the command in.

call_id string
The unique ID of the local shell tool call generated by the model.

id string
The unique ID of the local shell call.

status string
The status of the local shell call.

type string
The type of the local shell call. Always `local_shell_call` .

MCP tool call object
An invocation of a tool on an MCP server.

Hide properties

arguments string
A JSON string of the arguments passed to the tool.

id string
The unique ID of the tool call.

name string
The name of the tool that was run.

server_label string
The label of the MCP server running the tool.

type string
The type of the item. Always `mcp_call` .

error string or null
The error from the tool call, if any.

output string or null
The output from the tool call.

MCP list tools object
A list of tools available on an MCP server.

Hide properties

id string
The unique ID of the list.

server_label string
The label of the MCP server.

tools array
The tools available on the server.

Hide properties

input_schema object
The JSON schema describing the tool's input.

name string
The name of the tool.

annotations object or null
Additional annotations about the tool.

description string or null
The description of the tool.

type string
The type of the item. Always `mcp_list_tools`.

error string or null
Error message if the server could not list tools.

MCP approval request object
A request for human approval of a tool invocation.

Hide properties

arguments string
A JSON string of arguments for the tool.

id string
The unique ID of the approval request.

name string
The name of the tool to run.

server_label string
The label of the MCP server making the request.

type string
The type of the item. Always `mcp_approval_request`.

parallel_tool_calls boolean
Whether to allow the model to run tool calls in parallel.

temperature number or null

What sampling temperature to use, between 0 and 2. Higher values like 0.8 will make the output more random, while lower values like 0.2 will make it more focused and deterministic. We generally recommend altering this or `top_p` but not both.

tool_choice string or object

How the model should select which tool (or tools) to use when generating a response. See the `tools` parameter to see how to specify which tools the model can call.

Hide possible types

Tool choice mode string

Controls which (if any) tool is called by the model.

`none` means the model will not call any tool and instead generates a message.

`auto` means the model can pick between generating a message or calling one or more tools.

`required` means the model must call one or more tools.

Hosted tool object

Indicates that the model should use a built-in tool to generate a response. Learn more about built-in tools.

Hide properties

type string

The type of hosted tool the model should to use. Learn more about built-in tools.

Allowed values are:

`file_search`

`web_search_preview`

`computer_use_preview`

`code_interpreter`

`image_generation`

Function tool object

Use this option to force the model to call a specific function.

Hide properties

name string

The name of the function to call.

type string

For function calling, the type is always `function`.

MCP tool object

Use this option to force the model to call a specific tool on a remote MCP server.

Hide properties

server_label string

The label of the MCP server to use.

type string

For MCP tools, the type is always `mcp`.

name string or null

The name of the tool to call on the server.

tools array

An array of tools the model may call while generating a response. You can specify which tool to use by setting the `tool_choice` parameter.

The two categories of tools you can provide the model are:

Built-in tools: Tools that are provided by OpenAI that extend the model's capabilities, like web search or file search. Learn more about built-in tools.

Function calls (custom tools): Functions that are defined by you, enabling the model to call your own code. Learn more about function calling.

Hide possible types

Function object

Defines a function in your own code the model can choose to call. Learn more about function calling.

Hide properties

name string

The name of the function to call.

parameters object

A JSON schema object describing the parameters of the function.

strict boolean

Whether to enforce strict parameter validation. Default `true`.

type string
The type of the function tool. Always `function`.

description string
A description of the function. Used by the model to determine whether or not to call the function.

File search object
A tool that searches for relevant content from uploaded files. Learn more about the file search tool.

Hide properties

type string
The type of the file search tool. Always `file_search`.

vector_store_ids array
The IDs of the vector stores to search.

filters object
A filter to apply.

Hide possible types

Comparison Filter object
A filter used to compare a specified attribute key to a given value using a defined comparison operation.

Hide properties

key string
The key to compare against the value.

type string
Specifies the comparison operator: `eq`, `ne`, `gt`, `gte`, `lt`, `lte`.

- `eq`: equals
- `ne`: not equal
- `gt`: greater than
- `gte`: greater than or equal
- `lt`: less than
- `lte`: less than or equal

value string / number / boolean
The value to compare against the attribute key; supports string, number, or boolean

types.

Compound Filter object

Combine multiple filters using `and` or `or`.

Hide properties

filters array

Array of filters to combine. Items can be `ComparisonFilter` or `CompoundFilter`.

Hide possible types

Comparison Filter object

A filter used to compare a specified attribute key to a given value using a defined comparison operation.

Hide properties

key string

The key to compare against the value.

type string

Specifies the comparison operator: `eq`, `ne`, `gt`, `gte`, `lt`, `lte`.

- `eq`: equals
- `ne`: not equal
- `gt`: greater than
- `gte`: greater than or equal
- `lt`: less than
- `lte`: less than or equal

value string / number / boolean

The value to compare against the attribute key; supports string, number, or boolean types.

type string

Type of operation: `and` or `or`.

max_num_results integer

The maximum number of results to return. This number should be between 1 and 50 inclusive.

ranking_options object
Ranking options for search.

Hide properties

ranker string
The ranker to use for the file search.

score_threshold number
The score threshold for the file search, a number between 0 and 1. Numbers closer to 1 will attempt to return only the most relevant results, but may return fewer results.

Web search preview object
This tool searches the web for relevant results to use in a response. Learn more about the web search tool.

Hide properties

type string
The type of the web search tool. One of `web_search_preview` or `web_search_preview_2025_03_11`.

search_context_size string
High level guidance for the amount of context window space to use for the search. One of `low`, `medium`, or `high`. `medium` is the default.

user_location object
The user's location.

Hide properties

type string
The type of location approximation. Always `approximate`.

city string
Free text input for the city of the user, e.g. `San Francisco`.

country string
The two-letter ISO country code of the user, e.g. `US`.

region string
Free text input for the region of the user, e.g. `California`.

timezone string

The IANA timezone of the user, e.g. `America/Los_Angeles` .

Computer use preview object

A tool that controls a virtual computer. Learn more about the computer tool.

Hide properties

`display_height` integer

The height of the computer display.

`display_width` integer

The width of the computer display.

`environment` string

The type of computer environment to control.

`type` string

The type of the computer use tool. Always `computer_use_preview` .

MCP tool object

Give the model access to additional tools via remote Model Context Protocol (MCP) servers. Learn more about MCP.

Hide properties

`server_label` string

A label for this MCP server, used to identify it in tool calls.

`server_url` string

The URL for the MCP server.

`type` string

The type of the MCP tool. Always `mcp` .

`allowed_tools` array or object

List of allowed tool names or a filter object.

Hide possible types

MCP allowed tools array

A string array of allowed tool names

MCP allowed tools filter object

A filter object to specify which tools are allowed.

Hide properties

tool_names array
List of allowed tool names.

headers object or null
Optional HTTP headers to send to the MCP server. Use for authentication or other purposes.

require_approval object or string
Specify which of the MCP server's tools require approval.

Hide possible types

MCP tool approval filter object

Hide properties

always object
A list of tools that always require approval.

Hide properties

tool_names array
List of tools that require approval.

never object
A list of tools that never require approval.

Hide properties

tool_names array
List of tools that do not require approval.

MCP tool approval setting string
Specify a single approval policy for all tools. One of `always` or `never`. When set to `always`, all tools will require approval. When set to `never`, all tools will not require approval.

server_description string
Optional description of the MCP server, used to provide more context.

Code interpreter object
A tool that runs Python code to help generate a response to a prompt.

Hide properties

container string or object
The code interpreter container. Can be a container ID or an object that specifies uploaded file IDs to make available to your code.

Hide possible types

string
The container ID.

CodeInterpreterContainerAuto object
Configuration for a code interpreter container. Optionally specify the IDs of the files to run the code on.

Hide properties

type string
Always `auto`.

file_ids array
An optional list of uploaded files to make available to your code.

type string
The type of the code interpreter tool. Always `code_interpreter`.

Image generation tool object
A tool that generates images using a model like `gpt-image-1`.

Hide properties

type string
The type of the image generation tool. Always `image_generation`.

background string
Background type for the generated image. One of `transparent`, `opaque`, or `auto`. Default: `auto`.

input_image_mask object
Optional mask for inpainting. Contains `image_url` (string, optional) and `file_id` (string, optional).

Hide properties

file_id string

File ID for the mask image.

image_url string
Base64-encoded mask image.

model string
The image generation model to use. Default: `gpt-image-1`.

moderation string
Moderation level for the generated image. Default: `auto`.

output_compression integer
Compression level for the output image. Default: 100.

output_format string
The output format of the generated image. One of `png`, `webp`, or `jpeg`. Default: `png`.

partial_images integer
Number of partial images to generate in streaming mode, from 0 (default value) to 3.

quality string
The quality of the generated image. One of `low`, `medium`, `high`, or `auto`. Default: `auto`.

size string
The size of the generated image. One of `1024x1024`, `1024x1536`, `1536x1024`, or `auto`. Default: `auto`.

Local shell tool object
A tool that allows the model to execute shell commands in a local environment.

^ Hide properties

type string
The type of the local shell tool. Always `local_shell`.

top_p number or null
An alternative to sampling with temperature, called nucleus sampling, where the model considers the results of the tokens with top_p probability mass. So 0.1 means only the tokens comprising the top 10% probability mass are considered.

We generally recommend altering this or `temperature` but not both.

background boolean or null
Whether to run the model response in the background. Learn more.

max_output_tokens integer or null
An upper bound for the number of tokens that can be generated for a response, including visible output tokens and reasoning tokens.

max_tool_calls integer or null
The maximum number of total calls to built-in tools that can be processed in a response. This maximum number applies across all built-in tool calls, not per individual tool. Any further attempts to call a tool by the model will be ignored.

output_text string or null SDK Only
SDK-only convenience property that contains the aggregated text output from all `output_text` items in the `output` array, if any are present. Supported in the Python and JavaScript SDKs.

previous_response_id string or null
The unique ID of the previous response to the model. Use this to create multi-turn conversations. Learn more about conversation state.

prompt object or null
Reference to a prompt template and its variables. Learn more.

Hide properties

id string
The unique identifier of the prompt template to use.

variables map
Optional map of values to substitute in for variables in your prompt. The substitution values can either be strings, or other Response input types like images or files.

version string or null
Optional version of the prompt template.

reasoning object or null
o-series models only

Configuration options for reasoning models.

Hide properties

effort string or null
o-series models only

Constrains effort on reasoning for reasoning models. Currently supported values are `low`, `medium`, and `high`. Reducing reasoning effort can result in faster responses and fewer tokens used on reasoning in a response.

`generate_summary` Deprecated string or null

Deprecated: use `summary` instead.

A summary of the reasoning performed by the model. This can be useful for debugging and understanding the model's reasoning process. One of `auto`, `concise`, or `detailed`.

`summary` string or null

A summary of the reasoning performed by the model. This can be useful for debugging and understanding the model's reasoning process. One of `auto`, `concise`, or `detailed`.

`service_tier` string or null

Specifies the processing type used for serving the request.

> If set to 'auto', then the request will be processed with the service tier configured in the Project settings. Unless otherwise configured, the Project will use 'default'.
>
> If set to 'default', then the requset will be processed with the standard pricing and performance for the selected model.
>
> If set to 'flex' or 'priority', then the request will be processed with the corresponding service tier. Contact sales to learn more about Priority processing.
>
> When not set, the default behavior is 'auto'.

When the `service_tier` parameter is set, the response body will include the `service_tier` value based on the processing mode actually used to serve the request. This response value may be different from the value set in the parameter.

`status` string

The status of the response generation. One of `completed`, `failed`, `in_progress`, `cancelled`, `queued`, or `incomplete`.

`text` object

Configuration options for a text response from the model. Can be plain text or structured JSON data. Learn more:

> Text inputs and outputs
>
> Structured Outputs

^ Hide properties

`format` object

An object specifying the format that the model must output.

Configuring `{ "type": "json_schema" }` enables Structured Outputs, which ensures the model will match your supplied JSON schema. Learn more in the Structured Outputs guide.

The default format is `{ "type": "text" }` with no additional options.

Not recommended for gpt-4o and newer models:

Setting to `{ "type": "json_object" }` enables the older JSON mode, which ensures the message the model generates is valid JSON. Using `json_schema` is preferred for models that support it.

Hide possible types

Text object

Default response format. Used to generate text responses.

Hide properties

type string

The type of response format being defined. Always `text`.

JSON schema object

JSON Schema response format. Used to generate structured JSON responses. Learn more about Structured Outputs.

Hide properties

name string

The name of the response format. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

schema object

The schema for the response format, described as a JSON Schema object. Learn how to build JSON schemas here.

type string

The type of response format being defined. Always `json_schema`.

description string

A description of what the response format is for, used by the model to determine how to respond in the format.

strict boolean or null

Whether to enable strict schema adherence when generating the output. If set to true, the model will always follow the exact schema defined in the `schema` field. Only a subset of JSON Schema is supported when `strict` is `true`. To learn more, read the Structured Outputs guide.

JSON object object

JSON object response format. An older method of generating JSON responses. Using `json_schema` is recommended for models that support it. Note that the model will not generate JSON without a system or user message instructing it to do so.

Hide properties

type string

The type of response format being defined. Always `json_object`.

top_logprobs integer or null

An integer between 0 and 20 specifying the number of most likely tokens to return at each token position, each with an associated log probability.

truncation string or null

The truncation strategy to use for the model response.

`auto`: If the context of this response and previous ones exceeds the model's context window size, the model will truncate the response to fit the context window by dropping input items in the middle of the conversation.

`disabled` (default): If a model response will exceed the context window size for a model, the request will fail with a 400 error.

usage object

Represents token usage details including input tokens, output tokens, a breakdown of output tokens, and the total tokens used.

Hide properties

input_tokens integer

The number of input tokens.

input_tokens_details object

A detailed breakdown of the input tokens.

Hide properties

cached_tokens integer

The number of tokens that were retrieved from the cache. More on prompt caching.

output_tokens integer

The number of output tokens.

output_tokens_details object

A detailed breakdown of the output tokens.

Hide properties

reasoning_tokens integer

The number of reasoning tokens.

total_tokens integer

The total number of tokens used.

user string

A stable identifier for your end-users. Used to boost cache hit rates by better bucketing similar requests and to help OpenAI detect and prevent abuse. Learn more.

sequence_number integer

The sequence number of this event.

type string

The type of the event. Always `response.in_progress`.

OBJECT response.in_progress

```
{
  "type": "response.in_progress",
  "response": {
    "id": "resp_67ccfcdd16748190a91872c75d38539e09e4d4aac714747c",
    "object": "response",
    "created_at": 1741487325,
    "status": "in_progress",
    "error": null,
    "incomplete_details": null,
    "instructions": null,
    "max_output_tokens": null,
    "model": "gpt-4o-2024-08-06",
    "output": [],
    "parallel_tool_calls": true,
    "previous_response_id": null,
    "reasoning": {
      "effort": null,
      "summary": null
    },
    "store": true,
```

```
    "temperature": 1,
    "text": {
      "format": {
        "type": "text"
      }
    },
    "tool_choice": "auto",
    "tools": [],
    "top_p": 1,
    "truncation": "disabled",
    "usage": null,
    "user": null,
    "metadata": {}
  },
  "sequence_number": 1
}
```

response.completed

Emitted when the model response is complete.

response object

Properties of the completed response.

Hide properties

created_at number

Unix timestamp (in seconds) of when this Response was created.

error object or null

An error object returned when the model fails to generate a Response.

Hide properties

code string

The error code for the response.

message string

A human-readable description of the error.

id string

Unique identifier for this Response.

`incomplete_details` object or null

Details about why the response is incomplete.

Hide properties

`reason` string

The reason why the response is incomplete.

`instructions` string or array

A system (or developer) message inserted into the model's context.

When using along with `previous_response_id`, the instructions from a previous response will not be carried over to the next response. This makes it simple to swap out system (or developer) messages in new responses.

Hide possible types

string

A text input to the model, equivalent to a text input with the `developer` role.

Input item list array

A list of one or many input items to the model, containing different content types.

Hide possible types

Input message object

A message input to the model with a role indicating instruction following hierarchy. Instructions given with the `developer` or `system` role take precedence over instructions given with the `user` role. Messages with the `assistant` role are presumed to have been generated by the model in previous interactions.

Hide properties

`content` string or array

Text, image, or audio input to the model, used to generate a response. Can also contain previous assistant responses.

Hide possible types

Text input string

A text input to the model.

Input item content list array

A list of one or many input items to the model, containing different content types.

Hide possible types

Input text object

A text input to the model.

Hide properties

text string
The text input to the model.

type string
The type of the input item. Always `input_text`.

Input image object
An image input to the model. Learn about image inputs.

Hide properties

detail string
The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

type string
The type of the input item. Always `input_image`.

file_id string
The ID of the file to be sent to the model.

image_url string
The URL of the image to be sent to the model. A fully qualified URL or base64 encoded image in a data URL.

Input file object
A file input to the model.

Hide properties

type string
The type of the input item. Always `input_file`.

file_data string
The content of the file to be sent to the model.

file_id string
The ID of the file to be sent to the model.

file_url string
The URL of the file to be sent to the model.

filename string
The name of the file to be sent to the model.

role string
The role of the message input. One of `user`, `assistant`, `system`, or `developer`.

type string
The type of the message input. Always `message`.

Item object
An item representing part of the context for the response to be generated by the model. Can contain text, images, and audio inputs, as well as previous assistant responses and tool call outputs.

Hide possible types

Input message object
A message input to the model with a role indicating instruction following hierarchy. Instructions given with the `developer` or `system` role take precedence over instructions given with the `user` role.

Hide properties

content array
A list of one or many input items to the model, containing different content types.

Hide possible types

Input text object
A text input to the model.

Hide properties

text string
The text input to the model.

type string
The type of the input item. Always `input_text`.

Input image object
An image input to the model. Learn about image inputs.

Hide properties

detail string

The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

`type` string
The type of the input item. Always `input_image`.

`file_id` string
The ID of the file to be sent to the model.

`image_url` string
The URL of the image to be sent to the model. A fully qualified URL or base64 encoded image in a data URL.

Input file object
A file input to the model.

Hide properties

`type` string
The type of the input item. Always `input_file`.

`file_data` string
The content of the file to be sent to the model.

`file_id` string
The ID of the file to be sent to the model.

`file_url` string
The URL of the file to be sent to the model.

`filename` string
The name of the file to be sent to the model.

`role` string
The role of the message input. One of `user`, `system`, or `developer`.

`status` string
The status of item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

type string
The type of the message input. Always set to `message`.

Output message object
An output message from the model.

Hide properties

content array
The content of the output message.

Hide possible types

Output text object
A text output from the model.

Hide properties

annotations array
The annotations of the text output.

Hide possible types

File citation object
A citation to a file.

Hide properties

file_id string
The ID of the file.

filename string
The filename of the file cited.

index integer
The index of the file in the list of files.

type string
The type of the file citation. Always `file_citation`.

URL citation object
A citation for a web resource used to generate a model response.

Hide properties

end_index integer
The index of the last character of the URL citation in the message.

start_index integer
The index of the first character of the URL citation in the message.

title string
The title of the web resource.

type string
The type of the URL citation. Always `url_citation`.

url string
The URL of the web resource.

Container file citation object
A citation for a container file used to generate a model response.

Hide properties

container_id string
The ID of the container file.

end_index integer
The index of the last character of the container file citation in the message.

file_id string
The ID of the file.

filename string
The filename of the container file cited.

start_index integer
The index of the first character of the container file citation in the message.

type string
The type of the container file citation. Always `container_file_citation`.

File path object
A path to a file.

Hide properties

file_id string
The ID of the file.

index integer
The index of the file in the list of files.

type string
The type of the file path. Always `file_path` .

text string
The text output from the model.

type string
The type of the output text. Always `output_text` .

logprobs array
Hide properties

bytes array

logprob number

token string

top_logprobs array
Hide properties

bytes array

logprob number

token string

Refusal object
A refusal from the model.
Hide properties

refusal string
The refusal explanationfrom the model.

type string
The type of the refusal. Always `refusal`.

id string
The unique ID of the output message.

role string
The role of the output message. Always `assistant`.

status string
The status of the message input. One of `in_progress`, `completed`, or `incomplete`. Populated when input items are returned via API.

type string
The type of the output message. Always `message`.

File search tool call object
The results of a file search tool call. See the file search guide for more information.

Hide properties

id string
The unique ID of the file search tool call.

queries array
The queries used to search for files.

status string
The status of the file search tool call. One of `in_progress`, `searching`, `incomplete` or `failed`,

type string
The type of the file search tool call. Always `file_search_call`.

results array or null
The results of the file search tool call.

Hide properties

attributes map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard. Keys are strings with a maximum length of 64

characters. Values are strings with a maximum length of 512 characters, booleans, or numbers.

file_id string
The unique ID of the file.

filename string
The name of the file.

score number
The relevance score of the file - a value between 0 and 1.

text string
The text that was retrieved from the file.

Computer tool call object
A tool call to a computer use tool. See the computer use guide for more information.

Hide properties

action object

Hide possible types

Click object
A click action.

Hide properties

button string
Indicates which mouse button was pressed during the click. One of `left`, `right`, `wheel`, `back`, or `forward`.

type string
Specifies the event type. For a click action, this property is always set to `click`.

x integer
The x-coordinate where the click occurred.

y integer
The y-coordinate where the click occurred.

DoubleClick object
A double click action.

Hide properties

type string
Specifies the event type. For a double click action, this property is always set to `double_click`.

x integer
The x-coordinate where the double click occurred.

y integer
The y-coordinate where the double click occurred.

Drag object
A drag action.

Hide properties

path array
An array of coordinates representing the path of the drag action. Coordinates will appear as an array of objects, eg

```
[
  { x: 100, y: 200 },
  { x: 200, y: 300 }
]
```

Hide properties

x integer
The x-coordinate.

y integer
The y-coordinate.

type string
Specifies the event type. For a drag action, this property is always set to `drag`.

KeyPress object
A collection of keypresses the model would like to perform.

Hide properties

keys array

The combination of keys the model is requesting to be pressed. This is an array of strings, each representing a key.

type string
Specifies the event type. For a keypress action, this property is always set to `keypress`.

Move object
A mouse move action.

Hide properties

type string
Specifies the event type. For a move action, this property is always set to `move`.

x integer
The x-coordinate to move to.

y integer
The y-coordinate to move to.

Screenshot object
A screenshot action.

Hide properties

type string
Specifies the event type. For a screenshot action, this property is always set to `screenshot`.

Scroll object
A scroll action.

Hide properties

scroll_x integer
The horizontal scroll distance.

scroll_y integer
The vertical scroll distance.

type string
Specifies the event type. For a scroll action, this property is always set to `scroll`.

x integer
The x-coordinate where the scroll occurred.

y integer
The y-coordinate where the scroll occurred.

Type object
An action to type in text.

Hide properties

text string
The text to type.

type string
Specifies the event type. For a type action, this property is always set to `type`.

Wait object
A wait action.

Hide properties

type string
Specifies the event type. For a wait action, this property is always set to `wait`.

call_id string
An identifier used when responding to the tool call with output.

id string
The unique ID of the computer call.

pending_safety_checks array
The pending safety checks for the computer call.

Hide properties

code string
The type of the pending safety check.

id string
The ID of the pending safety check.

message string

Details about the pending safety check.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

type string
The type of the computer call. Always `computer_call`.

Computer tool call output object
The output of a computer tool call.

Hide properties

call_id string
The ID of the computer tool call that produced the output.

output object
A computer screenshot image used with the computer use tool.

Hide properties

type string
Specifies the event type. For a computer screenshot, this property is always set to `computer_screenshot`.

file_id string
The identifier of an uploaded file that contains the screenshot.

image_url string
The URL of the screenshot image.

type string
The type of the computer tool call output. Always `computer_call_output`.

acknowledged_safety_checks array
The safety checks reported by the API that have been acknowledged by the developer.

Hide properties

id string
The ID of the pending safety check.

code string

The type of the pending safety check.

message string
Details about the pending safety check.

id string
The ID of the computer tool call output.

status string
The status of the message input. One of `in_progress`, `completed`, or `incomplete`. Populated when input items are returned via API.

Web search tool call object
The results of a web search tool call. See the web search guide for more information.

Hide properties

action object
An object describing the specific action taken in this web search call. Includes details on how the model used the web (search, open_page, find).

Hide possible types

Search action object
Action type "search" - Performs a web search query.

Hide properties

query string
The search query.

type string
The action type.

Open page action object
Action type "open_page" - Opens a specific URL from search results.

Hide properties

type string
The action type.

url string
The URL opened by the model.

Find action object

Action type "find": Searches for a pattern within a loaded page.

Hide properties

pattern string

The pattern or text to search for within the page.

type string

The action type.

url string

The URL of the page searched for the pattern.

id string

The unique ID of the web search tool call.

status string

The status of the web search tool call.

type string

The type of the web search tool call. Always `web_search_call`.

Function tool call object

A tool call to run a function. See the function calling guide for more information.

Hide properties

arguments string

A JSON string of the arguments to pass to the function.

call_id string

The unique ID of the function tool call generated by the model.

name string

The name of the function to run.

type string

The type of the function tool call. Always `function_call`.

id string

The unique ID of the function tool call.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Function tool call output object
The output of a function tool call.

Hide properties

call_id string
The unique ID of the function tool call generated by the model.

output string
A JSON string of the output of the function tool call.

type string
The type of the function tool call output. Always `function_call_output`.

id string
The unique ID of the function tool call output. Populated when this item is returned via API.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Reasoning object
A description of the chain of thought used by a reasoning model while generating a response. Be sure to include these items in your `input` to the Responses API for subsequent turns of a conversation if you are manually managing context.

Hide properties

id string
The unique identifier of the reasoning content.

summary array
Reasoning text contents.

Hide properties

text string
A short summary of the reasoning used by the model when generating the response.

type string
The type of the object. Always `summary_text`.

type string
The type of the object. Always `reasoning`.

encrypted_content string or null
The encrypted content of the reasoning item - populated when a response is generated with `reasoning.encrypted_content` in the `include` parameter.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Image generation call object
An image generation request made by the model.

Hide properties

id string
The unique ID of the image generation call.

result string or null
The generated image encoded in base64.

status string
The status of the image generation call.

type string
The type of the image generation call. Always `image_generation_call`.

Code interpreter tool call object
A tool call to run code.

Hide properties

code string or null
The code to run, or null if not available.

container_id string
The ID of the container used to run the code.

id string

The unique ID of the code interpreter tool call.

outputs array or null

The outputs generated by the code interpreter, such as logs or images. Can be null if no outputs are available.

Hide possible types

Code interpreter output logs object

The logs output from the code interpreter.

Hide properties

logs string

The logs output from the code interpreter.

type string

The type of the output. Always 'logs'.

Code interpreter output image object

The image output from the code interpreter.

Hide properties

type string

The type of the output. Always 'image'.

url string

The URL of the image output from the code interpreter.

status string

The status of the code interpreter tool call.

type string

The type of the code interpreter tool call. Always `code_interpreter_call`.

Local shell call object

A tool call to run a command on the local shell.

Hide properties

action object

Execute a shell command on the server.

Hide properties

command array
The command to run.

env object
Environment variables to set for the command.

type string
The type of the local shell action. Always `exec`.

timeout_ms integer or null
Optional timeout in milliseconds for the command.

user string or null
Optional user to run the command as.

working_directory string or null
Optional working directory to run the command in.

call_id string
The unique ID of the local shell tool call generated by the model.

id string
The unique ID of the local shell call.

status string
The status of the local shell call.

type string
The type of the local shell call. Always `local_shell_call`.

Local shell call output object
The output of a local shell tool call.

Hide properties

id string
The unique ID of the local shell tool call generated by the model.

output string
A JSON string of the output of the local shell tool call.

type string

The type of the local shell tool call output. Always `local_shell_call_output`.

status string or null
The status of the item. One of `in_progress`, `completed`, or `incomplete`.

MCP list tools object
A list of tools available on an MCP server.

Hide properties

id string
The unique ID of the list.

server_label string
The label of the MCP server.

tools array
The tools available on the server.

Hide properties

input_schema object
The JSON schema describing the tool's input.

name string
The name of the tool.

annotations object or null
Additional annotations about the tool.

description string or null
The description of the tool.

type string
The type of the item. Always `mcp_list_tools`.

error string or null
Error message if the server could not list tools.

MCP approval request object
A request for human approval of a tool invocation.

Hide properties

arguments string
A JSON string of arguments for the tool.

id string
The unique ID of the approval request.

name string
The name of the tool to run.

server_label string
The label of the MCP server making the request.

type string
The type of the item. Always `mcp_approval_request`.

MCP approval response object
A response to an MCP approval request.

Hide properties

approval_request_id string
The ID of the approval request being answered.

approve boolean
Whether the request was approved.

type string
The type of the item. Always `mcp_approval_response`.

id string or null
The unique ID of the approval response

reason string or null
Optional reason for the decision.

MCP tool call object
An invocation of a tool on an MCP server.

Hide properties

arguments string
A JSON string of the arguments passed to the tool.

id string
The unique ID of the tool call.

name string
The name of the tool that was run.

server_label string
The label of the MCP server running the tool.

type string
The type of the item. Always `mcp_call`.

error string or null
The error from the tool call, if any.

output string or null
The output from the tool call.

Item reference object
An internal identifier for an item to reference.

Hide properties

id string
The ID of the item to reference.

type string
The type of item to reference. Always `item_reference`.

metadata map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

model string
Model ID used to generate the response, like `gpt-4o` or `o3`. OpenAI offers a wide range of models with different capabilities, performance characteristics, and price points. Refer to the model guide to browse and compare available models.

object string
The object type of this resource - always set to `response`.

output array
An array of content items generated by the model.

The length and order of items in the `output` array is dependent on the model's response.

Rather than accessing the first item in the `output` array and assuming it's an `assistant` message with the content generated by the model, you might consider using the `output_text` property where supported in SDKs.

Hide possible types

Output message object
An output message from the model.

Hide properties

content array
The content of the output message.

Hide possible types

Output text object
A text output from the model.

Hide properties

annotations array
The annotations of the text output.

Hide possible types

File citation object
A citation to a file.

Hide properties

file_id string
The ID of the file.

filename string
The filename of the file cited.

index integer
The index of the file in the list of files.

type string

The type of the file citation. Always `file_citation`.

URL citation object

A citation for a web resource used to generate a model response.

Hide properties

end_index integer

The index of the last character of the URL citation in the message.

start_index integer

The index of the first character of the URL citation in the message.

title string

The title of the web resource.

type string

The type of the URL citation. Always `url_citation`.

url string

The URL of the web resource.

Container file citation object

A citation for a container file used to generate a model response.

Hide properties

container_id string

The ID of the container file.

end_index integer

The index of the last character of the container file citation in the message.

file_id string

The ID of the file.

filename string

The filename of the container file cited.

start_index integer

The index of the first character of the container file citation in the message.

type string
The type of the container file citation. Always `container_file_citation` .

File path object
A path to a file.

Hide properties

file_id string
The ID of the file.

index integer
The index of the file in the list of files.

type string
The type of the file path. Always `file_path` .

text string
The text output from the model.

type string
The type of the output text. Always `output_text` .

logprobs array

Hide properties

bytes array

logprob number

token string

top_logprobs array

Hide properties

bytes array

logprob number

token string

Refusal object

A refusal from the model.

Hide properties

refusal string

The refusal explanationfrom the model.

type string

The type of the refusal. Always `refusal`.

id string

The unique ID of the output message.

role string

The role of the output message. Always `assistant`.

status string

The status of the message input. One of `in_progress`, `completed`, or `incomplete`. Populated when input items are returned via API.

type string

The type of the output message. Always `message`.

File search tool call object

The results of a file search tool call. See the file search guide for more information.

Hide properties

id string

The unique ID of the file search tool call.

queries array

The queries used to search for files.

status string

The status of the file search tool call. One of `in_progress`, `searching`, `incomplete` or `failed`,

type string

The type of the file search tool call. Always `file_search_call`.

results array or null

The results of the file search tool call.

Hide properties

attributes map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard. Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters, booleans, or numbers.

file_id string
The unique ID of the file.

filename string
The name of the file.

score number
The relevance score of the file - a value between 0 and 1.

text string
The text that was retrieved from the file.

Function tool call object
A tool call to run a function. See the function calling guide for more information.

Hide properties

arguments string
A JSON string of the arguments to pass to the function.

call_id string
The unique ID of the function tool call generated by the model.

name string
The name of the function to run.

type string
The type of the function tool call. Always `function_call`.

id string
The unique ID of the function tool call.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when

items are returned via API.

Web search tool call object

The results of a web search tool call. See the web search guide for more information.

Hide properties

action object

An object describing the specific action taken in this web search call. Includes details on how the model used the web (search, open_page, find).

Hide possible types

Search action object

Action type "search" - Performs a web search query.

Hide properties

query string

The search query.

type string

The action type.

Open page action object

Action type "open_page" - Opens a specific URL from search results.

Hide properties

type string

The action type.

url string

The URL opened by the model.

Find action object

Action type "find": Searches for a pattern within a loaded page.

Hide properties

pattern string

The pattern or text to search for within the page.

type string

The action type.

url string
The URL of the page searched for the pattern.

id string
The unique ID of the web search tool call.

status string
The status of the web search tool call.

type string
The type of the web search tool call. Always `web_search_call`.

Computer tool call object
A tool call to a computer use tool. See the computer use guide for more information.

Hide properties

action object

Hide possible types

Click object
A click action.

Hide properties

button string
Indicates which mouse button was pressed during the click. One of `left`, `right`, `wheel`, `back`, or `forward`.

type string
Specifies the event type. For a click action, this property is always set to `click`.

x integer
The x-coordinate where the click occurred.

y integer
The y-coordinate where the click occurred.

DoubleClick object
A double click action.

Hide properties

type string
Specifies the event type. For a double click action, this property is always set to `double_click`.

x integer
The x-coordinate where the double click occurred.

y integer
The y-coordinate where the double click occurred.

Drag object
A drag action.

Hide properties

path array
An array of coordinates representing the path of the drag action. Coordinates will appear as an array of objects, eg

```
[
  { x: 100, y: 200 },
  { x: 200, y: 300 }
]
```

Hide properties

x integer
The x-coordinate.

y integer
The y-coordinate.

type string
Specifies the event type. For a drag action, this property is always set to `drag`.

KeyPress object
A collection of keypresses the model would like to perform.

Hide properties

keys array
The combination of keys the model is requesting to be pressed. This is an array of strings,

each representing a key.

type string
Specifies the event type. For a keypress action, this property is always set to `keypress`.

Move object
A mouse move action.

Hide properties

type string
Specifies the event type. For a move action, this property is always set to `move`.

x integer
The x-coordinate to move to.

y integer
The y-coordinate to move to.

Screenshot object
A screenshot action.

Hide properties

type string
Specifies the event type. For a screenshot action, this property is always set to `screenshot`.

Scroll object
A scroll action.

Hide properties

scroll_x integer
The horizontal scroll distance.

scroll_y integer
The vertical scroll distance.

type string
Specifies the event type. For a scroll action, this property is always set to `scroll`.

x integer
The x-coordinate where the scroll occurred.

y integer
The y-coordinate where the scroll occurred.

Type object
An action to type in text.

^ Hide properties

text string
The text to type.

type string
Specifies the event type. For a type action, this property is always set to `type`.

Wait object
A wait action.

^ Hide properties

type string
Specifies the event type. For a wait action, this property is always set to `wait`.

call_id string
An identifier used when responding to the tool call with output.

id string
The unique ID of the computer call.

pending_safety_checks array
The pending safety checks for the computer call.

^ Hide properties

code string
The type of the pending safety check.

id string
The ID of the pending safety check.

message string
Details about the pending safety check.

status string

The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

type string
The type of the computer call. Always `computer_call`.

Reasoning object
A description of the chain of thought used by a reasoning model while generating a response. Be sure to include these items in your `input` to the Responses API for subsequent turns of a conversation if you are manually managing context.

Hide properties

id string
The unique identifier of the reasoning content.

summary array
Reasoning text contents.

Hide properties

text string
A short summary of the reasoning used by the model when generating the response.

type string
The type of the object. Always `summary_text`.

type string
The type of the object. Always `reasoning`.

encrypted_content string or null
The encrypted content of the reasoning item - populated when a response is generated with `reasoning.encrypted_content` in the `include` parameter.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Image generation call object
An image generation request made by the model.

Hide properties

id string

The unique ID of the image generation call.

result string or null
The generated image encoded in base64.

status string
The status of the image generation call.

type string
The type of the image generation call. Always `image_generation_call`.

Code interpreter tool call object
A tool call to run code.

Hide properties

code string or null
The code to run, or null if not available.

container_id string
The ID of the container used to run the code.

id string
The unique ID of the code interpreter tool call.

outputs array or null
The outputs generated by the code interpreter, such as logs or images. Can be null if no outputs are available.

Hide possible types

Code interpreter output logs object
The logs output from the code interpreter.

Hide properties

logs string
The logs output from the code interpreter.

type string
The type of the output. Always 'logs'.

Code interpreter output image object
The image output from the code interpreter.

Hide properties

type string
The type of the output. Always 'image'.

url string
The URL of the image output from the code interpreter.

status string
The status of the code interpreter tool call.

type string
The type of the code interpreter tool call. Always `code_interpreter_call`.

Local shell call object
A tool call to run a command on the local shell.

Hide properties

action object
Execute a shell command on the server.

Hide properties

command array
The command to run.

env object
Environment variables to set for the command.

type string
The type of the local shell action. Always `exec`.

timeout_ms integer or null
Optional timeout in milliseconds for the command.

user string or null
Optional user to run the command as.

working_directory string or null
Optional working directory to run the command in.

call_id string

The unique ID of the local shell tool call generated by the model.

id string
The unique ID of the local shell call.

status string
The status of the local shell call.

type string
The type of the local shell call. Always `local_shell_call`.

MCP tool call object
An invocation of a tool on an MCP server.

Hide properties

arguments string
A JSON string of the arguments passed to the tool.

id string
The unique ID of the tool call.

name string
The name of the tool that was run.

server_label string
The label of the MCP server running the tool.

type string
The type of the item. Always `mcp_call`.

error string or null
The error from the tool call, if any.

output string or null
The output from the tool call.

MCP list tools object
A list of tools available on an MCP server.

Hide properties

id string
The unique ID of the list.

server_label string
The label of the MCP server.

tools array
The tools available on the server.

Hide properties

input_schema object
The JSON schema describing the tool's input.

name string
The name of the tool.

annotations object or null
Additional annotations about the tool.

description string or null
The description of the tool.

type string
The type of the item. Always `mcp_list_tools`.

error string or null
Error message if the server could not list tools.

MCP approval request object
A request for human approval of a tool invocation.

Hide properties

arguments string
A JSON string of arguments for the tool.

id string
The unique ID of the approval request.

name string
The name of the tool to run.

server_label string
The label of the MCP server making the request.

type string
The type of the item. Always `mcp_approval_request`.

parallel_tool_calls boolean
Whether to allow the model to run tool calls in parallel.

temperature number or null
What sampling temperature to use, between 0 and 2. Higher values like 0.8 will make the output more random, while lower values like 0.2 will make it more focused and deterministic. We generally recommend altering this or `top_p` but not both.

tool_choice string or object
How the model should select which tool (or tools) to use when generating a response. See the `tools` parameter to see how to specify which tools the model can call.

^ Hide possible types

Tool choice mode string
Controls which (if any) tool is called by the model.

`none` means the model will not call any tool and instead generates a message.

`auto` means the model can pick between generating a message or calling one or more tools.

`required` means the model must call one or more tools.

Hosted tool object
Indicates that the model should use a built-in tool to generate a response. Learn more about built-in tools.

^ Hide properties

type string
The type of hosted tool the model should to use. Learn more about built-in tools.

Allowed values are:

- `file_search`
- `web_search_preview`
- `computer_use_preview`
- `code_interpreter`
- `image_generation`

Function tool object
Use this option to force the model to call a specific function.

Hide properties

name string
The name of the function to call.

type string
For function calling, the type is always `function`.

MCP tool object
Use this option to force the model to call a specific tool on a remote MCP server.

Hide properties

server_label string
The label of the MCP server to use.

type string
For MCP tools, the type is always `mcp`.

name string or null
The name of the tool to call on the server.

tools array
An array of tools the model may call while generating a response. You can specify which tool to use by setting the `tool_choice` parameter.

The two categories of tools you can provide the model are:

Built-in tools: Tools that are provided by OpenAI that extend the model's capabilities, like web search or file search. Learn more about built-in tools.

Function calls (custom tools): Functions that are defined by you, enabling the model to call your own code. Learn more about function calling.

Hide possible types

Function object

Defines a function in your own code the model can choose to call. Learn more about function calling.

Hide properties

name string

The name of the function to call.

parameters object

A JSON schema object describing the parameters of the function.

strict boolean

Whether to enforce strict parameter validation. Default `true`.

type string

The type of the function tool. Always `function`.

description string

A description of the function. Used by the model to determine whether or not to call the function.

File search object

A tool that searches for relevant content from uploaded files. Learn more about the file search tool.

Hide properties

type string

The type of the file search tool. Always `file_search`.

vector_store_ids array

The IDs of the vector stores to search.

filters object

A filter to apply.

Hide possible types

Comparison Filter object

A filter used to compare a specified attribute key to a given value using a defined comparison operation.

Hide properties

key string
The key to compare against the value.

type string
Specifies the comparison operator: `eq`, `ne`, `gt`, `gte`, `lt`, `lte`.

- `eq`: equals
- `ne`: not equal
- `gt`: greater than
- `gte`: greater than or equal
- `lt`: less than
- `lte`: less than or equal

value string / number / boolean
The value to compare against the attribute key; supports string, number, or boolean types.

Compound Filter object
Combine multiple filters using `and` or `or`.

Hide properties

filters array
Array of filters to combine. Items can be `ComparisonFilter` or `CompoundFilter`.

Hide possible types

Comparison Filter object
A filter used to compare a specified attribute key to a given value using a defined comparison operation.

Hide properties

key string
The key to compare against the value.

type string
Specifies the comparison operator: `eq`, `ne`, `gt`, `gte`, `lt`, `lte`.

- `eq`: equals
- `ne`: not equal
- `gt`: greater than

`gte` : greater than or equal

`lt` : less than

`lte` : less than or equal

value string / number / boolean
The value to compare against the attribute key; supports string, number, or boolean types.

type string
Type of operation: `and` or `or` .

max_num_results integer
The maximum number of results to return. This number should be between 1 and 50 inclusive.

ranking_options object
Ranking options for search.

Hide properties

ranker string
The ranker to use for the file search.

score_threshold number
The score threshold for the file search, a number between 0 and 1. Numbers closer to 1 will attempt to return only the most relevant results, but may return fewer results.

Web search preview object
This tool searches the web for relevant results to use in a response. Learn more about the web search tool.

Hide properties

type string
The type of the web search tool. One of `web_search_preview` or `web_search_preview_2025_03_11` .

search_context_size string
High level guidance for the amount of context window space to use for the search. One of `low`, `medium`, or `high`. `medium` is the default.

user_location object
The user's location.

Hide properties

type string
The type of location approximation. Always `approximate`.

city string
Free text input for the city of the user, e.g. `San Francisco`.

country string
The two-letter ISO country code of the user, e.g. `US`.

region string
Free text input for the region of the user, e.g. `California`.

timezone string
The IANA timezone of the user, e.g. `America/Los_Angeles`.

Computer use preview object
A tool that controls a virtual computer. Learn more about the computer tool.

Hide properties

display_height integer
The height of the computer display.

display_width integer
The width of the computer display.

environment string
The type of computer environment to control.

type string
The type of the computer use tool. Always `computer_use_preview`.

MCP tool object
Give the model access to additional tools via remote Model Context Protocol (MCP) servers. Learn

more about MCP.

Hide properties

server_label string
A label for this MCP server, used to identify it in tool calls.

server_url string
The URL for the MCP server.

type string
The type of the MCP tool. Always `mcp` .

allowed_tools array or object
List of allowed tool names or a filter object.

Hide possible types

MCP allowed tools array
A string array of allowed tool names

MCP allowed tools filter object
A filter object to specify which tools are allowed.

Hide properties

tool_names array
List of allowed tool names.

headers object or null
Optional HTTP headers to send to the MCP server. Use for authentication or other purposes.

require_approval object or string
Specify which of the MCP server's tools require approval.

Hide possible types

MCP tool approval filter object

Hide properties

always object
A list of tools that always require approval.

Hide properties

tool_names array
List of tools that require approval.

never object
A list of tools that never require approval.

Hide properties

tool_names array
List of tools that do not require approval.

MCP tool approval setting string
Specify a single approval policy for all tools. One of `always` or `never`. When set to `always`, all tools will require approval. When set to `never`, all tools will not require approval.

server_description string
Optional description of the MCP server, used to provide more context.

Code interpreter object
A tool that runs Python code to help generate a response to a prompt.

Hide properties

container string or object
The code interpreter container. Can be a container ID or an object that specifies uploaded file IDs to make available to your code.

Hide possible types

string
The container ID.

CodeInterpreterContainerAuto object
Configuration for a code interpreter container. Optionally specify the IDs of the files to run the code on.

Hide properties

type string
Always `auto`.

file_ids array
An optional list of uploaded files to make available to your code.

type string
The type of the code interpreter tool. Always `code_interpreter` .

Image generation tool object
A tool that generates images using a model like `gpt-image-1` .

Hide properties

type string
The type of the image generation tool. Always `image_generation` .

background string
Background type for the generated image. One of `transparent` , `opaque` , or `auto` . Default: `auto` .

input_image_mask object
Optional mask for inpainting. Contains `image_url` (string, optional) and `file_id` (string, optional).

Hide properties

file_id string
File ID for the mask image.

image_url string
Base64-encoded mask image.

model string
The image generation model to use. Default: `gpt-image-1` .

moderation string
Moderation level for the generated image. Default: `auto` .

output_compression integer
Compression level for the output image. Default: 100.

output_format string
The output format of the generated image. One of `png` , `webp` , or `jpeg` . Default: `png` .

partial_images integer
Number of partial images to generate in streaming mode, from 0 (default value) to 3.

quality string
The quality of the generated image. One of `low`, `medium`, `high`, or `auto`. Default: `auto`.

size string
The size of the generated image. One of `1024x1024`, `1024x1536`, `1536x1024`, or `auto`. Default: `auto`.

Local shell tool object
A tool that allows the model to execute shell commands in a local environment.

Hide properties

type string
The type of the local shell tool. Always `local_shell`.

top_p number or null
An alternative to sampling with temperature, called nucleus sampling, where the model considers the results of the tokens with top_p probability mass. So 0.1 means only the tokens comprising the top 10% probability mass are considered.

We generally recommend altering this or `temperature` but not both.

background boolean or null
Whether to run the model response in the background. Learn more.

max_output_tokens integer or null
An upper bound for the number of tokens that can be generated for a response, including visible output tokens and reasoning tokens.

max_tool_calls integer or null
The maximum number of total calls to built-in tools that can be processed in a response. This maximum number applies across all built-in tool calls, not per individual tool. Any further attempts to call a tool by the model will be ignored.

output_text string or null SDK Only
SDK-only convenience property that contains the aggregated text output from all `output_text` items in the `output` array, if any are present. Supported in the Python and JavaScript SDKs.

previous_response_id string or null

The unique ID of the previous response to the model. Use this to create multi-turn conversations. Learn more about conversation state.

prompt object or null
Reference to a prompt template and its variables. Learn more.

Hide properties

id string
The unique identifier of the prompt template to use.

variables map
Optional map of values to substitute in for variables in your prompt. The substitution values can either be strings, or other Response input types like images or files.

version string or null
Optional version of the prompt template.

reasoning object or null
o-series models only

Configuration options for reasoning models.

Hide properties

effort string or null
o-series models only

Constrains effort on reasoning for reasoning models. Currently supported values are `low`, `medium`, and `high`. Reducing reasoning effort can result in faster responses and fewer tokens used on reasoning in a response.

generate_summary Deprecated string or null
Deprecated: use `summary` instead.

A summary of the reasoning performed by the model. This can be useful for debugging and understanding the model's reasoning process. One of `auto`, `concise`, or `detailed`.

summary string or null
A summary of the reasoning performed by the model. This can be useful for debugging and understanding the model's reasoning process. One of `auto`, `concise`, or `detailed`.

service_tier string or null
Specifies the processing type used for serving the request.

If set to 'auto', then the request will be processed with the service tier configured in the Project

settings. Unless otherwise configured, the Project will use 'default'.

If set to 'default', then the requset will be processed with the standard pricing and performance for the selected model.

If set to 'flex' or 'priority', then the request will be processed with the corresponding service tier. Contact sales to learn more about Priority processing.

When not set, the default behavior is 'auto'.

When the `service_tier` parameter is set, the response body will include the `service_tier` value based on the processing mode actually used to serve the request. This response value may be different from the value set in the parameter.

`status` string
The status of the response generation. One of `completed`, `failed`, `in_progress`, `cancelled`, `queued`, or `incomplete`.

`text` object
Configuration options for a text response from the model. Can be plain text or structured JSON data. Learn more:

Text inputs and outputs

Structured Outputs

Hide properties

`format` object
An object specifying the format that the model must output.

Configuring `{ "type": "json_schema" }` enables Structured Outputs, which ensures the model will match your supplied JSON schema. Learn more in the Structured Outputs guide.

The default format is `{ "type": "text" }` with no additional options.

Not recommended for gpt-4o and newer models:

Setting to `{ "type": "json_object" }` enables the older JSON mode, which ensures the message the model generates is valid JSON. Using `json_schema` is preferred for models that support it.

Hide possible types

Text object
Default response format. Used to generate text responses.

Hide properties

`type` string
The type of response format being defined. Always `text`.

JSON schema object

JSON Schema response format. Used to generate structured JSON responses. Learn more about Structured Outputs.

Hide properties

name string

The name of the response format. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

schema object

The schema for the response format, described as a JSON Schema object. Learn how to build JSON schemas here.

type string

The type of response format being defined. Always `json_schema`.

description string

A description of what the response format is for, used by the model to determine how to respond in the format.

strict boolean or null

Whether to enable strict schema adherence when generating the output. If set to true, the model will always follow the exact schema defined in the `schema` field. Only a subset of JSON Schema is supported when `strict` is `true`. To learn more, read the Structured Outputs guide.

JSON object object

JSON object response format. An older method of generating JSON responses. Using `json_schema` is recommended for models that support it. Note that the model will not generate JSON without a system or user message instructing it to do so.

Hide properties

type string

The type of response format being defined. Always `json_object`.

top_logprobs integer or null

An integer between 0 and 20 specifying the number of most likely tokens to return at each token position, each with an associated log probability.

truncation string or null

The truncation strategy to use for the model response.

`auto` : If the context of this response and previous ones exceeds the model's context window size, the model will truncate the response to fit the context window by dropping input items in the middle of the conversation.

`disabled` (default): If a model response will exceed the context window size for a model, the request will fail with a 400 error.

`usage` object

Represents token usage details including input tokens, output tokens, a breakdown of output tokens, and the total tokens used.

Hide properties

`input_tokens` integer

The number of input tokens.

`input_tokens_details` object

A detailed breakdown of the input tokens.

Hide properties

`cached_tokens` integer

The number of tokens that were retrieved from the cache. More on prompt caching.

`output_tokens` integer

The number of output tokens.

`output_tokens_details` object

A detailed breakdown of the output tokens.

Hide properties

`reasoning_tokens` integer

The number of reasoning tokens.

`total_tokens` integer

The total number of tokens used.

`user` string

A stable identifier for your end-users. Used to boost cache hit rates by better bucketing similar requests and to help OpenAI detect and prevent abuse. Learn more.

sequence_number integer

The sequence number for this event.

type string

The type of the event. Always `response.completed` .

OBJECT response.completed


```
{
  "type": "response.completed",
  "response": {
    "id": "resp_123",
    "object": "response",
    "created_at": 1740855869,
    "status": "completed",
    "error": null,
    "incomplete_details": null,
    "input": [],
    "instructions": null,
    "max_output_tokens": null,
    "model": "gpt-4o-mini-2024-07-18",
    "output": [
      {
        "id": "msg_123",
        "type": "message",
        "role": "assistant",
        "content": [
          {
            "type": "output_text",
            "text": "In a shimmering forest under a sky full of stars, a lonely unicorn na
            "annotations": []
          }
        ]
      }
    ],
    "previous_response_id": null,
    "reasoning_effort": null,
    "store": false,
    "temperature": 1,
    "text": {
      "format": {
        "type": "text"
      }
    },
    "tool_choice": "auto",
```

```
    "tools": [],
    "top_p": 1,
    "truncation": "disabled",
    "usage": {
      "input_tokens": 0,
      "output_tokens": 0,
      "output_tokens_details": {
        "reasoning_tokens": 0
      },
      "total_tokens": 0
    },
    "user": null,
    "metadata": {}
  },
  "sequence_number": 1
}
```

response.failed

An event that is emitted when a response fails.

response object

The response that failed.

Hide properties

created_at number

Unix timestamp (in seconds) of when this Response was created.

error object or null

An error object returned when the model fails to generate a Response.

Hide properties

code string

The error code for the response.

message string

A human-readable description of the error.

id string

Unique identifier for this Response.

`incomplete_details` object or null

Details about why the response is incomplete.

Hide properties

`reason` string

The reason why the response is incomplete.

`instructions` string or array

A system (or developer) message inserted into the model's context.

When using along with `previous_response_id`, the instructions from a previous response will not be carried over to the next response. This makes it simple to swap out system (or developer) messages in new responses.

Hide possible types

string

A text input to the model, equivalent to a text input with the `developer` role.

Input item list array

A list of one or many input items to the model, containing different content types.

Hide possible types

Input message object

A message input to the model with a role indicating instruction following hierarchy. Instructions given with the `developer` or `system` role take precedence over instructions given with the `user` role. Messages with the `assistant` role are presumed to have been generated by the model in previous interactions.

Hide properties

`content` string or array

Text, image, or audio input to the model, used to generate a response. Can also contain previous assistant responses.

Hide possible types

Text input string

A text input to the model.

Input item content list array

A list of one or many input items to the model, containing different content types.

Hide possible types

Input text object

A text input to the model.

Hide properties

text string
The text input to the model.

type string
The type of the input item. Always `input_text`.

Input image object
An image input to the model. Learn about image inputs.

Hide properties

detail string
The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

type string
The type of the input item. Always `input_image`.

file_id string
The ID of the file to be sent to the model.

image_url string
The URL of the image to be sent to the model. A fully qualified URL or base64 encoded image in a data URL.

Input file object
A file input to the model.

Hide properties

type string
The type of the input item. Always `input_file`.

file_data string
The content of the file to be sent to the model.

file_id string
The ID of the file to be sent to the model.

file_url string
The URL of the file to be sent to the model.

filename string
The name of the file to be sent to the model.

role string
The role of the message input. One of `user`, `assistant`, `system`, or `developer`.

type string
The type of the message input. Always `message`.

Item object
An item representing part of the context for the response to be generated by the model. Can contain text, images, and audio inputs, as well as previous assistant responses and tool call outputs.

Hide possible types

Input message object
A message input to the model with a role indicating instruction following hierarchy. Instructions given with the `developer` or `system` role take precedence over instructions given with the `user` role.

Hide properties

content array
A list of one or many input items to the model, containing different content types.

Hide possible types

Input text object
A text input to the model.

Hide properties

text string
The text input to the model.

type string
The type of the input item. Always `input_text`.

Input image object
An image input to the model. Learn about image inputs.

Hide properties

detail string

The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

`type` string
The type of the input item. Always `input_image`.

`file_id` string
The ID of the file to be sent to the model.

`image_url` string
The URL of the image to be sent to the model. A fully qualified URL or base64 encoded image in a data URL.

Input file object
A file input to the model.

Hide properties

`type` string
The type of the input item. Always `input_file`.

`file_data` string
The content of the file to be sent to the model.

`file_id` string
The ID of the file to be sent to the model.

`file_url` string
The URL of the file to be sent to the model.

`filename` string
The name of the file to be sent to the model.

`role` string
The role of the message input. One of `user`, `system`, or `developer`.

`status` string
The status of item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

type string
The type of the message input. Always set to `message`.

Output message object
An output message from the model.

Hide properties

content array
The content of the output message.

Hide possible types

Output text object
A text output from the model.

Hide properties

annotations array
The annotations of the text output.

Hide possible types

File citation object
A citation to a file.

Hide properties

file_id string
The ID of the file.

filename string
The filename of the file cited.

index integer
The index of the file in the list of files.

type string
The type of the file citation. Always `file_citation`.

URL citation object
A citation for a web resource used to generate a model response.

Hide properties

end_index integer
The index of the last character of the URL citation in the message.

start_index integer
The index of the first character of the URL citation in the message.

title string
The title of the web resource.

type string
The type of the URL citation. Always `url_citation`.

url string
The URL of the web resource.

Container file citation object
A citation for a container file used to generate a model response.

Hide properties

container_id string
The ID of the container file.

end_index integer
The index of the last character of the container file citation in the message.

file_id string
The ID of the file.

filename string
The filename of the container file cited.

start_index integer
The index of the first character of the container file citation in the message.

type string
The type of the container file citation. Always `container_file_citation`.

File path object
A path to a file.

Hide properties

file_id string
The ID of the file.

index integer
The index of the file in the list of files.

type string
The type of the file path. Always `file_path` .

text string
The text output from the model.

type string
The type of the output text. Always `output_text` .

logprobs array
Hide properties

bytes array

logprob number

token string

top_logprobs array
Hide properties

bytes array

logprob number

token string

Refusal object
A refusal from the model.
Hide properties

refusal string
The refusal explanationfrom the model.

type string
The type of the refusal. Always `refusal`.

id string
The unique ID of the output message.

role string
The role of the output message. Always `assistant`.

status string
The status of the message input. One of `in_progress`, `completed`, or `incomplete`. Populated when input items are returned via API.

type string
The type of the output message. Always `message`.

File search tool call object
The results of a file search tool call. See the file search guide for more information.

Hide properties

id string
The unique ID of the file search tool call.

queries array
The queries used to search for files.

status string
The status of the file search tool call. One of `in_progress`, `searching`, `incomplete` or `failed`,

type string
The type of the file search tool call. Always `file_search_call`.

results array or null
The results of the file search tool call.

Hide properties

attributes map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard. Keys are strings with a maximum length of 64

characters. Values are strings with a maximum length of 512 characters, booleans, or numbers.

file_id string
The unique ID of the file.

filename string
The name of the file.

score number
The relevance score of the file - a value between 0 and 1.

text string
The text that was retrieved from the file.

Computer tool call object
A tool call to a computer use tool. See the computer use guide for more information.

Hide properties

action object

Hide possible types

Click object
A click action.

Hide properties

button string
Indicates which mouse button was pressed during the click. One of `left`, `right`, `wheel`, `back`, or `forward`.

type string
Specifies the event type. For a click action, this property is always set to `click`.

x integer
The x-coordinate where the click occurred.

y integer
The y-coordinate where the click occurred.

DoubleClick object
A double click action.

Hide properties

type string
Specifies the event type. For a double click action, this property is always set to `double_click`.

x integer
The x-coordinate where the double click occurred.

y integer
The y-coordinate where the double click occurred.

Drag object
A drag action.

Hide properties

path array
An array of coordinates representing the path of the drag action. Coordinates will appear as an array of objects, eg

```
[
  { x: 100, y: 200 },
  { x: 200, y: 300 }
]
```

Hide properties

x integer
The x-coordinate.

y integer
The y-coordinate.

type string
Specifies the event type. For a drag action, this property is always set to `drag`.

KeyPress object
A collection of keypresses the model would like to perform.

Hide properties

keys array

The combination of keys the model is requesting to be pressed. This is an array of strings, each representing a key.

type string
Specifies the event type. For a keypress action, this property is always set to `keypress`.

Move object
A mouse move action.

Hide properties

type string
Specifies the event type. For a move action, this property is always set to `move`.

x integer
The x-coordinate to move to.

y integer
The y-coordinate to move to.

Screenshot object
A screenshot action.

Hide properties

type string
Specifies the event type. For a screenshot action, this property is always set to `screenshot`.

Scroll object
A scroll action.

Hide properties

scroll_x integer
The horizontal scroll distance.

scroll_y integer
The vertical scroll distance.

type string
Specifies the event type. For a scroll action, this property is always set to `scroll`.

x integer
The x-coordinate where the scroll occurred.

y integer
The y-coordinate where the scroll occurred.

Type object
An action to type in text.

Hide properties

text string
The text to type.

type string
Specifies the event type. For a type action, this property is always set to `type`.

Wait object
A wait action.

Hide properties

type string
Specifies the event type. For a wait action, this property is always set to `wait`.

call_id string
An identifier used when responding to the tool call with output.

id string
The unique ID of the computer call.

pending_safety_checks array
The pending safety checks for the computer call.

Hide properties

code string
The type of the pending safety check.

id string
The ID of the pending safety check.

message string

Details about the pending safety check.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

type string
The type of the computer call. Always `computer_call`.

Computer tool call output object
The output of a computer tool call.

Hide properties

call_id string
The ID of the computer tool call that produced the output.

output object
A computer screenshot image used with the computer use tool.

Hide properties

type string
Specifies the event type. For a computer screenshot, this property is always set to `computer_screenshot`.

file_id string
The identifier of an uploaded file that contains the screenshot.

image_url string
The URL of the screenshot image.

type string
The type of the computer tool call output. Always `computer_call_output`.

acknowledged_safety_checks array
The safety checks reported by the API that have been acknowledged by the developer.

Hide properties

id string
The ID of the pending safety check.

code string

The type of the pending safety check.

message string
Details about the pending safety check.

id string
The ID of the computer tool call output.

status string
The status of the message input. One of `in_progress`, `completed`, or `incomplete`. Populated when input items are returned via API.

Web search tool call object
The results of a web search tool call. See the web search guide for more information.

Hide properties

action object
An object describing the specific action taken in this web search call. Includes details on how the model used the web (search, open_page, find).

Hide possible types

Search action object
Action type "search" - Performs a web search query.

Hide properties

query string
The search query.

type string
The action type.

Open page action object
Action type "open_page" - Opens a specific URL from search results.

Hide properties

type string
The action type.

url string
The URL opened by the model.

Find action object

Action type "find": Searches for a pattern within a loaded page.

Hide properties

pattern string

The pattern or text to search for within the page.

type string

The action type.

url string

The URL of the page searched for the pattern.

id string

The unique ID of the web search tool call.

status string

The status of the web search tool call.

type string

The type of the web search tool call. Always `web_search_call`.

Function tool call object

A tool call to run a function. See the function calling guide for more information.

Hide properties

arguments string

A JSON string of the arguments to pass to the function.

call_id string

The unique ID of the function tool call generated by the model.

name string

The name of the function to run.

type string

The type of the function tool call. Always `function_call`.

id string

The unique ID of the function tool call.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Function tool call output object
The output of a function tool call.

Hide properties

call_id string
The unique ID of the function tool call generated by the model.

output string
A JSON string of the output of the function tool call.

type string
The type of the function tool call output. Always `function_call_output`.

id string
The unique ID of the function tool call output. Populated when this item is returned via API.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Reasoning object
A description of the chain of thought used by a reasoning model while generating a response. Be sure to include these items in your `input` to the Responses API for subsequent turns of a conversation if you are manually managing context.

Hide properties

id string
The unique identifier of the reasoning content.

summary array
Reasoning text contents.

Hide properties

text string
A short summary of the reasoning used by the model when generating the response.

type string
The type of the object. Always `summary_text`.

type string
The type of the object. Always `reasoning`.

encrypted_content string or null
The encrypted content of the reasoning item - populated when a response is generated with `reasoning.encrypted_content` in the `include` parameter.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Image generation call object
An image generation request made by the model.

Hide properties

id string
The unique ID of the image generation call.

result string or null
The generated image encoded in base64.

status string
The status of the image generation call.

type string
The type of the image generation call. Always `image_generation_call`.

Code interpreter tool call object
A tool call to run code.

Hide properties

code string or null
The code to run, or null if not available.

container_id string
The ID of the container used to run the code.

id string

The unique ID of the code interpreter tool call.

outputs array or null

The outputs generated by the code interpreter, such as logs or images. Can be null if no outputs are available.

Hide possible types

Code interpreter output logs object

The logs output from the code interpreter.

Hide properties

logs string

The logs output from the code interpreter.

type string

The type of the output. Always 'logs'.

Code interpreter output image object

The image output from the code interpreter.

Hide properties

type string

The type of the output. Always 'image'.

url string

The URL of the image output from the code interpreter.

status string

The status of the code interpreter tool call.

type string

The type of the code interpreter tool call. Always `code_interpreter_call`.

Local shell call object

A tool call to run a command on the local shell.

Hide properties

action object

Execute a shell command on the server.

Hide properties

command array
The command to run.

env object
Environment variables to set for the command.

type string
The type of the local shell action. Always `exec`.

timeout_ms integer or null
Optional timeout in milliseconds for the command.

user string or null
Optional user to run the command as.

working_directory string or null
Optional working directory to run the command in.

call_id string
The unique ID of the local shell tool call generated by the model.

id string
The unique ID of the local shell call.

status string
The status of the local shell call.

type string
The type of the local shell call. Always `local_shell_call`.

Local shell call output object
The output of a local shell tool call.

Hide properties

id string
The unique ID of the local shell tool call generated by the model.

output string
A JSON string of the output of the local shell tool call.

type string

The type of the local shell tool call output. Always `local_shell_call_output`.

status string or null
The status of the item. One of `in_progress`, `completed`, or `incomplete`.

MCP list tools object
A list of tools available on an MCP server.

Hide properties

id string
The unique ID of the list.

server_label string
The label of the MCP server.

tools array
The tools available on the server.

Hide properties

input_schema object
The JSON schema describing the tool's input.

name string
The name of the tool.

annotations object or null
Additional annotations about the tool.

description string or null
The description of the tool.

type string
The type of the item. Always `mcp_list_tools`.

error string or null
Error message if the server could not list tools.

MCP approval request object
A request for human approval of a tool invocation.

Hide properties

arguments string
A JSON string of arguments for the tool.

id string
The unique ID of the approval request.

name string
The name of the tool to run.

server_label string
The label of the MCP server making the request.

type string
The type of the item. Always `mcp_approval_request`.

MCP approval response object
A response to an MCP approval request.

Hide properties

approval_request_id string
The ID of the approval request being answered.

approve boolean
Whether the request was approved.

type string
The type of the item. Always `mcp_approval_response`.

id string or null
The unique ID of the approval response

reason string or null
Optional reason for the decision.

MCP tool call object
An invocation of a tool on an MCP server.

Hide properties

arguments string
A JSON string of the arguments passed to the tool.

id string
The unique ID of the tool call.

name string
The name of the tool that was run.

server_label string
The label of the MCP server running the tool.

type string
The type of the item. Always `mcp_call`.

error string or null
The error from the tool call, if any.

output string or null
The output from the tool call.

Item reference object
An internal identifier for an item to reference.

Hide properties

id string
The ID of the item to reference.

type string
The type of item to reference. Always `item_reference`.

metadata map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

model string
Model ID used to generate the response, like `gpt-4o` or `o3`. OpenAI offers a wide range of models with different capabilities, performance characteristics, and price points. Refer to the model guide to browse and compare available models.

object string
The object type of this resource - always set to `response`.

output array
An array of content items generated by the model.

The length and order of items in the `output` array is dependent on the model's response.

Rather than accessing the first item in the `output` array and assuming it's an `assistant` message with the content generated by the model, you might consider using the `output_text` property where supported in SDKs.

Hide possible types

Output message object
An output message from the model.

Hide properties

content array
The content of the output message.

Hide possible types

Output text object
A text output from the model.

Hide properties

annotations array
The annotations of the text output.

Hide possible types

File citation object
A citation to a file.

Hide properties

file_id string
The ID of the file.

filename string
The filename of the file cited.

index integer
The index of the file in the list of files.

type string

The type of the file citation. Always `file_citation`.

URL citation object

A citation for a web resource used to generate a model response.

Hide properties

end_index integer

The index of the last character of the URL citation in the message.

start_index integer

The index of the first character of the URL citation in the message.

title string

The title of the web resource.

type string

The type of the URL citation. Always `url_citation`.

url string

The URL of the web resource.

Container file citation object

A citation for a container file used to generate a model response.

Hide properties

container_id string

The ID of the container file.

end_index integer

The index of the last character of the container file citation in the message.

file_id string

The ID of the file.

filename string

The filename of the container file cited.

start_index integer

The index of the first character of the container file citation in the message.

type string
The type of the container file citation. Always `container_file_citation` .

File path object
A path to a file.

Hide properties

file_id string
The ID of the file.

index integer
The index of the file in the list of files.

type string
The type of the file path. Always `file_path` .

text string
The text output from the model.

type string
The type of the output text. Always `output_text` .

logprobs array

Hide properties

bytes array

logprob number

token string

top_logprobs array

Hide properties

bytes array

logprob number

token string

Refusal object

A refusal from the model.

Hide properties

refusal string

The refusal explanationfrom the model.

type string

The type of the refusal. Always `refusal`.

id string

The unique ID of the output message.

role string

The role of the output message. Always `assistant`.

status string

The status of the message input. One of `in_progress`, `completed`, or `incomplete`. Populated when input items are returned via API.

type string

The type of the output message. Always `message`.

File search tool call object

The results of a file search tool call. See the file search guide for more information.

Hide properties

id string

The unique ID of the file search tool call.

queries array

The queries used to search for files.

status string

The status of the file search tool call. One of `in_progress`, `searching`, `incomplete` or `failed`,

type string

The type of the file search tool call. Always `file_search_call`.

results array or null

The results of the file search tool call.

Hide properties

attributes map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard. Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters, booleans, or numbers.

file_id string
The unique ID of the file.

filename string
The name of the file.

score number
The relevance score of the file - a value between 0 and 1.

text string
The text that was retrieved from the file.

Function tool call object
A tool call to run a function. See the function calling guide for more information.

Hide properties

arguments string
A JSON string of the arguments to pass to the function.

call_id string
The unique ID of the function tool call generated by the model.

name string
The name of the function to run.

type string
The type of the function tool call. Always `function_call`.

id string
The unique ID of the function tool call.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when

items are returned via API.

Web search tool call object

The results of a web search tool call. See the web search guide for more information.

Hide properties

action object

An object describing the specific action taken in this web search call. Includes details on how the model used the web (search, open_page, find).

Hide possible types

Search action object

Action type "search" - Performs a web search query.

Hide properties

query string

The search query.

type string

The action type.

Open page action object

Action type "open_page" - Opens a specific URL from search results.

Hide properties

type string

The action type.

url string

The URL opened by the model.

Find action object

Action type "find": Searches for a pattern within a loaded page.

Hide properties

pattern string

The pattern or text to search for within the page.

type string

The action type.

url string
The URL of the page searched for the pattern.

id string
The unique ID of the web search tool call.

status string
The status of the web search tool call.

type string
The type of the web search tool call. Always `web_search_call`.

Computer tool call object
A tool call to a computer use tool. See the computer use guide for more information.

Hide properties

action object

Hide possible types

Click object
A click action.

Hide properties

button string
Indicates which mouse button was pressed during the click. One of `left`, `right`, `wheel`, `back`, or `forward`.

type string
Specifies the event type. For a click action, this property is always set to `click`.

x integer
The x-coordinate where the click occurred.

y integer
The y-coordinate where the click occurred.

DoubleClick object
A double click action.

Hide properties

type string
Specifies the event type. For a double click action, this property is always set to `double_click`.

x integer
The x-coordinate where the double click occurred.

y integer
The y-coordinate where the double click occurred.

Drag object
A drag action.

Hide properties

path array
An array of coordinates representing the path of the drag action. Coordinates will appear as an array of objects, eg

```
[
  { x: 100, y: 200 },
  { x: 200, y: 300 }
]
```

Hide properties

x integer
The x-coordinate.

y integer
The y-coordinate.

type string
Specifies the event type. For a drag action, this property is always set to `drag`.

KeyPress object
A collection of keypresses the model would like to perform.

Hide properties

keys array
The combination of keys the model is requesting to be pressed. This is an array of strings,

each representing a key.

type string
Specifies the event type. For a keypress action, this property is always set to `keypress`.

Move object
A mouse move action.

Hide properties

type string
Specifies the event type. For a move action, this property is always set to `move`.

x integer
The x-coordinate to move to.

y integer
The y-coordinate to move to.

Screenshot object
A screenshot action.

Hide properties

type string
Specifies the event type. For a screenshot action, this property is always set to `screenshot`.

Scroll object
A scroll action.

Hide properties

scroll_x integer
The horizontal scroll distance.

scroll_y integer
The vertical scroll distance.

type string
Specifies the event type. For a scroll action, this property is always set to `scroll`.

x integer
The x-coordinate where the scroll occurred.

y integer
The y-coordinate where the scroll occurred.

Type object
An action to type in text.

Hide properties

text string
The text to type.

type string
Specifies the event type. For a type action, this property is always set to `type`.

Wait object
A wait action.

Hide properties

type string
Specifies the event type. For a wait action, this property is always set to `wait`.

call_id string
An identifier used when responding to the tool call with output.

id string
The unique ID of the computer call.

pending_safety_checks array
The pending safety checks for the computer call.

Hide properties

code string
The type of the pending safety check.

id string
The ID of the pending safety check.

message string
Details about the pending safety check.

status string

The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

type string
The type of the computer call. Always `computer_call`.

Reasoning object
A description of the chain of thought used by a reasoning model while generating a response. Be sure to include these items in your `input` to the Responses API for subsequent turns of a conversation if you are manually managing context.

Hide properties

id string
The unique identifier of the reasoning content.

summary array
Reasoning text contents.

Hide properties

text string
A short summary of the reasoning used by the model when generating the response.

type string
The type of the object. Always `summary_text`.

type string
The type of the object. Always `reasoning`.

encrypted_content string or null
The encrypted content of the reasoning item - populated when a response is generated with `reasoning.encrypted_content` in the `include` parameter.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Image generation call object
An image generation request made by the model.

Hide properties

id string

The unique ID of the image generation call.

result string or null
The generated image encoded in base64.

status string
The status of the image generation call.

type string
The type of the image generation call. Always `image_generation_call`.

Code interpreter tool call object
A tool call to run code.

Hide properties

code string or null
The code to run, or null if not available.

container_id string
The ID of the container used to run the code.

id string
The unique ID of the code interpreter tool call.

outputs array or null
The outputs generated by the code interpreter, such as logs or images. Can be null if no outputs are available.

Hide possible types

Code interpreter output logs object
The logs output from the code interpreter.

Hide properties

logs string
The logs output from the code interpreter.

type string
The type of the output. Always 'logs'.

Code interpreter output image object
The image output from the code interpreter.

Hide properties

type string
The type of the output. Always 'image'.

url string
The URL of the image output from the code interpreter.

status string
The status of the code interpreter tool call.

type string
The type of the code interpreter tool call. Always `code_interpreter_call`.

Local shell call object
A tool call to run a command on the local shell.

Hide properties

action object
Execute a shell command on the server.

Hide properties

command array
The command to run.

env object
Environment variables to set for the command.

type string
The type of the local shell action. Always `exec`.

timeout_ms integer or null
Optional timeout in milliseconds for the command.

user string or null
Optional user to run the command as.

working_directory string or null
Optional working directory to run the command in.

call_id string

The unique ID of the local shell tool call generated by the model.

id string
The unique ID of the local shell call.

status string
The status of the local shell call.

type string
The type of the local shell call. Always `local_shell_call`.

MCP tool call object
An invocation of a tool on an MCP server.

Hide properties

arguments string
A JSON string of the arguments passed to the tool.

id string
The unique ID of the tool call.

name string
The name of the tool that was run.

server_label string
The label of the MCP server running the tool.

type string
The type of the item. Always `mcp_call`.

error string or null
The error from the tool call, if any.

output string or null
The output from the tool call.

MCP list tools object
A list of tools available on an MCP server.

Hide properties

id string
The unique ID of the list.

server_label string
The label of the MCP server.

tools array
The tools available on the server.

^ Hide properties

input_schema object
The JSON schema describing the tool's input.

name string
The name of the tool.

annotations object or null
Additional annotations about the tool.

description string or null
The description of the tool.

type string
The type of the item. Always `mcp_list_tools`.

error string or null
Error message if the server could not list tools.

MCP approval request object
A request for human approval of a tool invocation.

^ Hide properties

arguments string
A JSON string of arguments for the tool.

id string
The unique ID of the approval request.

name string
The name of the tool to run.

server_label string
The label of the MCP server making the request.

type string
The type of the item. Always `mcp_approval_request`.

parallel_tool_calls boolean
Whether to allow the model to run tool calls in parallel.

temperature number or null
What sampling temperature to use, between 0 and 2. Higher values like 0.8 will make the output more random, while lower values like 0.2 will make it more focused and deterministic. We generally recommend altering this or `top_p` but not both.

tool_choice string or object
How the model should select which tool (or tools) to use when generating a response. See the `tools` parameter to see how to specify which tools the model can call.

^ Hide possible types

Tool choice mode string
Controls which (if any) tool is called by the model.

`none` means the model will not call any tool and instead generates a message.

`auto` means the model can pick between generating a message or calling one or more tools.

`required` means the model must call one or more tools.

Hosted tool object
Indicates that the model should use a built-in tool to generate a response. Learn more about built-in tools.

^ Hide properties

type string
The type of hosted tool the model should to use. Learn more about built-in tools.

Allowed values are:

- `file_search`
- `web_search_preview`
- `computer_use_preview`
- `code_interpreter`
- `image_generation`

Function tool object
Use this option to force the model to call a specific function.

Hide properties

name string
The name of the function to call.

type string
For function calling, the type is always `function`.

MCP tool object
Use this option to force the model to call a specific tool on a remote MCP server.

Hide properties

server_label string
The label of the MCP server to use.

type string
For MCP tools, the type is always `mcp`.

name string or null
The name of the tool to call on the server.

tools array
An array of tools the model may call while generating a response. You can specify which tool to use by setting the `tool_choice` parameter.

The two categories of tools you can provide the model are:

Built-in tools: Tools that are provided by OpenAI that extend the model's capabilities, like web search or file search. Learn more about built-in tools.

Function calls (custom tools): Functions that are defined by you, enabling the model to call your own code. Learn more about function calling.

Hide possible types

Function object

Defines a function in your own code the model can choose to call. Learn more about function calling.

Hide properties

name string

The name of the function to call.

parameters object

A JSON schema object describing the parameters of the function.

strict boolean

Whether to enforce strict parameter validation. Default `true`.

type string

The type of the function tool. Always `function`.

description string

A description of the function. Used by the model to determine whether or not to call the function.

File search object

A tool that searches for relevant content from uploaded files. Learn more about the file search tool.

Hide properties

type string

The type of the file search tool. Always `file_search`.

vector_store_ids array

The IDs of the vector stores to search.

filters object

A filter to apply.

Hide possible types

Comparison Filter object

A filter used to compare a specified attribute key to a given value using a defined comparison operation.

Hide properties

key string
The key to compare against the value.

type string
Specifies the comparison operator: `eq`, `ne`, `gt`, `gte`, `lt`, `lte`.

- `eq`: equals
- `ne`: not equal
- `gt`: greater than
- `gte`: greater than or equal
- `lt`: less than
- `lte`: less than or equal

value string / number / boolean
The value to compare against the attribute key; supports string, number, or boolean types.

Compound Filter object
Combine multiple filters using `and` or `or`.

Hide properties

filters array
Array of filters to combine. Items can be `ComparisonFilter` or `CompoundFilter`.

Hide possible types

Comparison Filter object
A filter used to compare a specified attribute key to a given value using a defined comparison operation.

Hide properties

key string
The key to compare against the value.

type string
Specifies the comparison operator: `eq`, `ne`, `gt`, `gte`, `lt`, `lte`.

- `eq`: equals
- `ne`: not equal
- `gt`: greater than

`gte` : greater than or equal

`lt` : less than

`lte` : less than or equal

`value` string / number / boolean
The value to compare against the attribute key; supports string, number, or boolean types.

`type` string
Type of operation: `and` or `or`.

`max_num_results` integer
The maximum number of results to return. This number should be between 1 and 50 inclusive.

`ranking_options` object
Ranking options for search.

Hide properties

`ranker` string
The ranker to use for the file search.

`score_threshold` number
The score threshold for the file search, a number between 0 and 1. Numbers closer to 1 will attempt to return only the most relevant results, but may return fewer results.

Web search preview object
This tool searches the web for relevant results to use in a response. Learn more about the web search tool.

Hide properties

`type` string
The type of the web search tool. One of `web_search_preview` or `web_search_preview_2025_03_11`.

search_context_size string
High level guidance for the amount of context window space to use for the search. One of `low`, `medium`, or `high`. `medium` is the default.

user_location object
The user's location.

^ Hide properties

type string
The type of location approximation. Always `approximate`.

city string
Free text input for the city of the user, e.g. `San Francisco`.

country string
The two-letter ISO country code of the user, e.g. `US`.

region string
Free text input for the region of the user, e.g. `California`.

timezone string
The IANA timezone of the user, e.g. `America/Los_Angeles`.

Computer use preview object
A tool that controls a virtual computer. Learn more about the computer tool.

^ Hide properties

display_height integer
The height of the computer display.

display_width integer
The width of the computer display.

environment string
The type of computer environment to control.

type string
The type of the computer use tool. Always `computer_use_preview`.

MCP tool object
Give the model access to additional tools via remote Model Context Protocol (MCP) servers. Learn

more about MCP.

Hide properties

server_label string
A label for this MCP server, used to identify it in tool calls.

server_url string
The URL for the MCP server.

type string
The type of the MCP tool. Always `mcp` .

allowed_tools array or object
List of allowed tool names or a filter object.

Hide possible types

MCP allowed tools array
A string array of allowed tool names

MCP allowed tools filter object
A filter object to specify which tools are allowed.

Hide properties

tool_names array
List of allowed tool names.

headers object or null
Optional HTTP headers to send to the MCP server. Use for authentication or other purposes.

require_approval object or string
Specify which of the MCP server's tools require approval.

Hide possible types

MCP tool approval filter object

Hide properties

always object
A list of tools that always require approval.

Hide properties

tool_names array
List of tools that require approval.

never object
A list of tools that never require approval.

Hide properties

tool_names array
List of tools that do not require approval.

MCP tool approval setting string
Specify a single approval policy for all tools. One of `always` or `never`. When set to `always`, all tools will require approval. When set to `never`, all tools will not require approval.

server_description string
Optional description of the MCP server, used to provide more context.

Code interpreter object
A tool that runs Python code to help generate a response to a prompt.

Hide properties

container string or object
The code interpreter container. Can be a container ID or an object that specifies uploaded file IDs to make available to your code.

Hide possible types

string
The container ID.

CodeInterpreterContainerAuto object
Configuration for a code interpreter container. Optionally specify the IDs of the files to run the code on.

Hide properties

type string
Always `auto`.

file_ids array
An optional list of uploaded files to make available to your code.

type string
The type of the code interpreter tool. Always `code_interpreter` .

Image generation tool object
A tool that generates images using a model like `gpt-image-1` .

Hide properties

type string
The type of the image generation tool. Always `image_generation` .

background string
Background type for the generated image. One of `transparent` , `opaque` , or `auto` . Default: `auto` .

input_image_mask object
Optional mask for inpainting. Contains `image_url` (string, optional) and `file_id` (string, optional).

Hide properties

file_id string
File ID for the mask image.

image_url string
Base64-encoded mask image.

model string
The image generation model to use. Default: `gpt-image-1` .

moderation string
Moderation level for the generated image. Default: `auto` .

output_compression integer
Compression level for the output image. Default: 100.

output_format string
The output format of the generated image. One of `png` , `webp` , or `jpeg` . Default: `png` .

partial_images integer
Number of partial images to generate in streaming mode, from 0 (default value) to 3.

quality string
The quality of the generated image. One of `low`, `medium`, `high`, or `auto`. Default: `auto`.

size string
The size of the generated image. One of `1024x1024`, `1024x1536`, `1536x1024`, or `auto`. Default: `auto`.

Local shell tool object
A tool that allows the model to execute shell commands in a local environment.

Hide properties

type string
The type of the local shell tool. Always `local_shell`.

top_p number or null
An alternative to sampling with temperature, called nucleus sampling, where the model considers the results of the tokens with top_p probability mass. So 0.1 means only the tokens comprising the top 10% probability mass are considered.

We generally recommend altering this or `temperature` but not both.

background boolean or null
Whether to run the model response in the background. Learn more.

max_output_tokens integer or null
An upper bound for the number of tokens that can be generated for a response, including visible output tokens and reasoning tokens.

max_tool_calls integer or null
The maximum number of total calls to built-in tools that can be processed in a response. This maximum number applies across all built-in tool calls, not per individual tool. Any further attempts to call a tool by the model will be ignored.

output_text string or null SDK Only
SDK-only convenience property that contains the aggregated text output from all `output_text` items in the `output` array, if any are present. Supported in the Python and JavaScript SDKs.

previous_response_id string or null

The unique ID of the previous response to the model. Use this to create multi-turn conversations. Learn more about conversation state.

prompt object or null
Reference to a prompt template and its variables. Learn more.

Hide properties

id string
The unique identifier of the prompt template to use.

variables map
Optional map of values to substitute in for variables in your prompt. The substitution values can either be strings, or other Response input types like images or files.

version string or null
Optional version of the prompt template.

reasoning object or null
o-series models only

Configuration options for reasoning models.

Hide properties

effort string or null
o-series models only

Constrains effort on reasoning for reasoning models. Currently supported values are `low`, `medium`, and `high`. Reducing reasoning effort can result in faster responses and fewer tokens used on reasoning in a response.

generate_summary Deprecated string or null
Deprecated: use `summary` instead.

A summary of the reasoning performed by the model. This can be useful for debugging and understanding the model's reasoning process. One of `auto`, `concise`, or `detailed`.

summary string or null
A summary of the reasoning performed by the model. This can be useful for debugging and understanding the model's reasoning process. One of `auto`, `concise`, or `detailed`.

service_tier string or null
Specifies the processing type used for serving the request.

If set to 'auto', then the request will be processed with the service tier configured in the Project

settings. Unless otherwise configured, the Project will use 'default'.

If set to 'default', then the requset will be processed with the standard pricing and performance for the selected model.

If set to 'flex' or 'priority', then the request will be processed with the corresponding service tier. Contact sales to learn more about Priority processing.

When not set, the default behavior is 'auto'.

When the `service_tier` parameter is set, the response body will include the `service_tier` value based on the processing mode actually used to serve the request. This response value may be different from the value set in the parameter.

`status` string
The status of the response generation. One of `completed`, `failed`, `in_progress`, `cancelled`, `queued`, or `incomplete`.

`text` object
Configuration options for a text response from the model. Can be plain text or structured JSON data. Learn more:

Text inputs and outputs

Structured Outputs

Hide properties

`format` object
An object specifying the format that the model must output.

Configuring `{ "type": "json_schema" }` enables Structured Outputs, which ensures the model will match your supplied JSON schema. Learn more in the Structured Outputs guide.

The default format is `{ "type": "text" }` with no additional options.

Not recommended for gpt-4o and newer models:

Setting to `{ "type": "json_object" }` enables the older JSON mode, which ensures the message the model generates is valid JSON. Using `json_schema` is preferred for models that support it.

Hide possible types

Text object
Default response format. Used to generate text responses.

Hide properties

`type` string
The type of response format being defined. Always `text`.

JSON schema object

JSON Schema response format. Used to generate structured JSON responses. Learn more about Structured Outputs.

Hide properties

name string

The name of the response format. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

schema object

The schema for the response format, described as a JSON Schema object. Learn how to build JSON schemas here.

type string

The type of response format being defined. Always `json_schema`.

description string

A description of what the response format is for, used by the model to determine how to respond in the format.

strict boolean or null

Whether to enable strict schema adherence when generating the output. If set to true, the model will always follow the exact schema defined in the `schema` field. Only a subset of JSON Schema is supported when `strict` is `true`. To learn more, read the Structured Outputs guide.

JSON object object

JSON object response format. An older method of generating JSON responses. Using `json_schema` is recommended for models that support it. Note that the model will not generate JSON without a system or user message instructing it to do so.

Hide properties

type string

The type of response format being defined. Always `json_object`.

top_logprobs integer or null

An integer between 0 and 20 specifying the number of most likely tokens to return at each token position, each with an associated log probability.

truncation string or null

The truncation strategy to use for the model response.

`auto` : If the context of this response and previous ones exceeds the model's context window size, the model will truncate the response to fit the context window by dropping input items in the middle of the conversation.

`disabled` (default): If a model response will exceed the context window size for a model, the request will fail with a 400 error.

`usage` object

Represents token usage details including input tokens, output tokens, a breakdown of output tokens, and the total tokens used.

^ Hide properties

`input_tokens` integer

The number of input tokens.

`input_tokens_details` object

A detailed breakdown of the input tokens.

^ Hide properties

`cached_tokens` integer

The number of tokens that were retrieved from the cache. More on prompt caching.

`output_tokens` integer

The number of output tokens.

`output_tokens_details` object

A detailed breakdown of the output tokens.

^ Hide properties

`reasoning_tokens` integer

The number of reasoning tokens.

`total_tokens` integer

The total number of tokens used.

`user` string

A stable identifier for your end-users. Used to boost cache hit rates by better bucketing similar requests and to help OpenAI detect and prevent abuse. Learn more.

sequence_number integer

The sequence number of this event.

type string

The type of the event. Always `response.failed`.

OBJECT response.failed

```
{
  "type": "response.failed",
  "response": {
    "id": "resp_123",
    "object": "response",
    "created_at": 1740855869,
    "status": "failed",
    "error": {
      "code": "server_error",
      "message": "The model failed to generate a response."
    },
    "incomplete_details": null,
    "instructions": null,
    "max_output_tokens": null,
    "model": "gpt-4o-mini-2024-07-18",
    "output": [],
    "previous_response_id": null,
    "reasoning_effort": null,
    "store": false,
    "temperature": 1,
    "text": {
      "format": {
        "type": "text"
      }
    },
    "tool_choice": "auto",
    "tools": [],
    "top_p": 1,
    "truncation": "disabled",
    "usage": null,
    "user": null,
    "metadata": {}
  }
}
```

response.incomplete

An event that is emitted when a response finishes as incomplete.

response object
The response that was incomplete.

Hide properties

created_at number
Unix timestamp (in seconds) of when this Response was created.

error object or null
An error object returned when the model fails to generate a Response.

Hide properties

code string
The error code for the response.

message string
A human-readable description of the error.

id string
Unique identifier for this Response.

incomplete_details object or null
Details about why the response is incomplete.

Hide properties

reason string
The reason why the response is incomplete.

instructions string or array
A system (or developer) message inserted into the model's context.

When using along with `previous_response_id`, the instructions from a previous response will not be carried over to the next response. This makes it simple to swap out system (or developer) messages in new responses.

Hide possible types

string
A text input to the model, equivalent to a text input with the `developer` role.

Input item list array

A list of one or many input items to the model, containing different content types.

Hide possible types

Input message object

A message input to the model with a role indicating instruction following hierarchy. Instructions given with the `developer` or `system` role take precedence over instructions given with the `user` role. Messages with the `assistant` role are presumed to have been generated by the model in previous interactions.

Hide properties

content string or array

Text, image, or audio input to the model, used to generate a response. Can also contain previous assistant responses.

Hide possible types

Text input string

A text input to the model.

Input item content list array

A list of one or many input items to the model, containing different content types.

Hide possible types

Input text object

A text input to the model.

Hide properties

text string

The text input to the model.

type string

The type of the input item. Always `input_text`.

Input image object

An image input to the model. Learn about image inputs.

Hide properties

detail string

The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

type string

The type of the input item. Always `input_image`.

`file_id` string
The ID of the file to be sent to the model.

`image_url` string
The URL of the image to be sent to the model. A fully qualified URL or base64 encoded image in a data URL.

Input file object
A file input to the model.

Hide properties

`type` string
The type of the input item. Always `input_file`.

`file_data` string
The content of the file to be sent to the model.

`file_id` string
The ID of the file to be sent to the model.

`file_url` string
The URL of the file to be sent to the model.

`filename` string
The name of the file to be sent to the model.

`role` string
The role of the message input. One of `user`, `assistant`, `system`, or `developer`.

`type` string
The type of the message input. Always `message`.

Item object
An item representing part of the context for the response to be generated by the model. Can contain text, images, and audio inputs, as well as previous assistant responses and tool call outputs.

Hide possible types

Input message object

A message input to the model with a role indicating instruction following hierarchy. Instructions given with the `developer` or `system` role take precedence over instructions given with the `user` role.

Hide properties

content array

A list of one or many input items to the model, containing different content types.

Hide possible types

Input text object

A text input to the model.

Hide properties

text string

The text input to the model.

type string

The type of the input item. Always `input_text`.

Input image object

An image input to the model. Learn about image inputs.

Hide properties

detail string

The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

type string

The type of the input item. Always `input_image`.

file_id string

The ID of the file to be sent to the model.

image_url string

The URL of the image to be sent to the model. A fully qualified URL or base64 encoded image in a data URL.

Input file object

A file input to the model.

Hide properties

type string
The type of the input item. Always `input_file`.

file_data string
The content of the file to be sent to the model.

file_id string
The ID of the file to be sent to the model.

file_url string
The URL of the file to be sent to the model.

filename string
The name of the file to be sent to the model.

role string
The role of the message input. One of `user`, `system`, or `developer`.

status string
The status of item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

type string
The type of the message input. Always set to `message`.

Output message object
An output message from the model.

Hide properties

content array
The content of the output message.

Hide possible types

Output text object
A text output from the model.

Hide properties

annotations array
The annotations of the text output.

Hide possible types

File citation object

A citation to a file.

Hide properties

file_id string

The ID of the file.

filename string

The filename of the file cited.

index integer

The index of the file in the list of files.

type string

The type of the file citation. Always `file_citation`.

URL citation object

A citation for a web resource used to generate a model response.

Hide properties

end_index integer

The index of the last character of the URL citation in the message.

start_index integer

The index of the first character of the URL citation in the message.

title string

The title of the web resource.

type string

The type of the URL citation. Always `url_citation`.

url string

The URL of the web resource.

Container file citation object

A citation for a container file used to generate a model response.

Hide properties

container_id string

The ID of the container file.

end_index integer
The index of the last character of the container file citation in the message.

file_id string
The ID of the file.

filename string
The filename of the container file cited.

start_index integer
The index of the first character of the container file citation in the message.

type string
The type of the container file citation. Always `container_file_citation`.

File path object
A path to a file.

Hide properties

file_id string
The ID of the file.

index integer
The index of the file in the list of files.

type string
The type of the file path. Always `file_path`.

text string
The text output from the model.

type string
The type of the output text. Always `output_text`.

logprobs array

Hide properties

bytes array

logprob number

token string

top_logprobs array

Hide properties

bytes array

logprob number

token string

Refusal object

A refusal from the model.

Hide properties

refusal string

The refusal explanationfrom the model.

type string

The type of the refusal. Always `refusal`.

id string

The unique ID of the output message.

role string

The role of the output message. Always `assistant`.

status string

The status of the message input. One of `in_progress`, `completed`, or `incomplete`. Populated when input items are returned via API.

type string

The type of the output message. Always `message`.

File search tool call object

The results of a file search tool call. See the file search guide for more information.

Hide properties

id string
The unique ID of the file search tool call.

queries array
The queries used to search for files.

status string
The status of the file search tool call. One of `in_progress`, `searching`, `incomplete` or `failed`,

type string
The type of the file search tool call. Always `file_search_call`.

results array or null
The results of the file search tool call.

Hide properties

attributes map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard. Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters, booleans, or numbers.

file_id string
The unique ID of the file.

filename string
The name of the file.

score number
The relevance score of the file - a value between 0 and 1.

text string
The text that was retrieved from the file.

Computer tool call object
A tool call to a computer use tool. See the computer use guide for more information.

Hide properties

action object

Hide possible types

Click object

A click action.

Hide properties

button string

Indicates which mouse button was pressed during the click. One of `left`, `right`, `wheel`, `back`, or `forward`.

type string

Specifies the event type. For a click action, this property is always set to `click`.

x integer

The x-coordinate where the click occurred.

y integer

The y-coordinate where the click occurred.

DoubleClick object

A double click action.

Hide properties

type string

Specifies the event type. For a double click action, this property is always set to `double_click`.

x integer

The x-coordinate where the double click occurred.

y integer

The y-coordinate where the double click occurred.

Drag object

A drag action.

Hide properties

path array

An array of coordinates representing the path of the drag action. Coordinates will

appear as an array of objects, eg

```
[
  { x: 100, y: 200 },
  { x: 200, y: 300 }
]
```

Hide properties

x integer
The x-coordinate.

y integer
The y-coordinate.

type string
Specifies the event type. For a drag action, this property is always set to `drag`.

KeyPress object
A collection of keypresses the model would like to perform.

Hide properties

keys array
The combination of keys the model is requesting to be pressed. This is an array of strings, each representing a key.

type string
Specifies the event type. For a keypress action, this property is always set to `keypress`.

Move object
A mouse move action.

Hide properties

type string
Specifies the event type. For a move action, this property is always set to `move`.

x integer
The x-coordinate to move to.

y integer

The y-coordinate to move to.

Screenshot object

A screenshot action.

Hide properties

type string

Specifies the event type. For a screenshot action, this property is always set to `screenshot`.

Scroll object

A scroll action.

Hide properties

scroll_x integer

The horizontal scroll distance.

scroll_y integer

The vertical scroll distance.

type string

Specifies the event type. For a scroll action, this property is always set to `scroll`.

x integer

The x-coordinate where the scroll occurred.

y integer

The y-coordinate where the scroll occurred.

Type object

An action to type in text.

Hide properties

text string

The text to type.

type string

Specifies the event type. For a type action, this property is always set to `type`.

Wait

object

A wait action.

Hide properties

type string

Specifies the event type. For a wait action, this property is always set to `wait`.

call_id string

An identifier used when responding to the tool call with output.

id string

The unique ID of the computer call.

pending_safety_checks array

The pending safety checks for the computer call.

Hide properties

code string

The type of the pending safety check.

id string

The ID of the pending safety check.

message string

Details about the pending safety check.

status string

The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

type string

The type of the computer call. Always `computer_call`.

Computer tool call output object

The output of a computer tool call.

Hide properties

call_id string

The ID of the computer tool call that produced the output.

output object

A computer screenshot image used with the computer use tool.

Hide properties

type string
Specifies the event type. For a computer screenshot, this property is always set to `computer_screenshot`.

file_id string
The identifier of an uploaded file that contains the screenshot.

image_url string
The URL of the screenshot image.

type string
The type of the computer tool call output. Always `computer_call_output`.

acknowledged_safety_checks array
The safety checks reported by the API that have been acknowledged by the developer.

Hide properties

id string
The ID of the pending safety check.

code string
The type of the pending safety check.

message string
Details about the pending safety check.

id string
The ID of the computer tool call output.

status string
The status of the message input. One of `in_progress`, `completed`, or `incomplete`. Populated when input items are returned via API.

Web search tool call object
The results of a web search tool call. See the web search guide for more information.

Hide properties

action object

An object describing the specific action taken in this web search call. Includes details on how the model used the web (search, open_page, find).

Hide possible types

Search action object
Action type "search" - Performs a web search query.

Hide properties

query string
The search query.

type string
The action type.

Open page action object
Action type "open_page" - Opens a specific URL from search results.

Hide properties

type string
The action type.

url string
The URL opened by the model.

Find action object
Action type "find": Searches for a pattern within a loaded page.

Hide properties

pattern string
The pattern or text to search for within the page.

type string
The action type.

url string
The URL of the page searched for the pattern.

id string
The unique ID of the web search tool call.

status string

The status of the web search tool call.

type string
The type of the web search tool call. Always `web_search_call`.

Function tool call object
A tool call to run a function. See the function calling guide for more information.

Hide properties

arguments string
A JSON string of the arguments to pass to the function.

call_id string
The unique ID of the function tool call generated by the model.

name string
The name of the function to run.

type string
The type of the function tool call. Always `function_call`.

id string
The unique ID of the function tool call.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Function tool call output object
The output of a function tool call.

Hide properties

call_id string
The unique ID of the function tool call generated by the model.

output string
A JSON string of the output of the function tool call.

type string
The type of the function tool call output. Always `function_call_output`.

id string

The unique ID of the function tool call output. Populated when this item is returned via API.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Reasoning object
A description of the chain of thought used by a reasoning model while generating a response. Be sure to include these items in your `input` to the Responses API for subsequent turns of a conversation if you are manually managing context.

Hide properties

id string
The unique identifier of the reasoning content.

summary array
Reasoning text contents.

Hide properties

text string
A short summary of the reasoning used by the model when generating the response.

type string
The type of the object. Always `summary_text`.

type string
The type of the object. Always `reasoning`.

encrypted_content string or null
The encrypted content of the reasoning item - populated when a response is generated with `reasoning.encrypted_content` in the `include` parameter.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Image generation call object
An image generation request made by the model.

Hide properties

id string
The unique ID of the image generation call.

result string or null
The generated image encoded in base64.

status string
The status of the image generation call.

type string
The type of the image generation call. Always `image_generation_call`.

Code interpreter tool call object
A tool call to run code.

Hide properties

code string or null
The code to run, or null if not available.

container_id string
The ID of the container used to run the code.

id string
The unique ID of the code interpreter tool call.

outputs array or null
The outputs generated by the code interpreter, such as logs or images. Can be null if no outputs are available.

Hide possible types

Code interpreter output logs object
The logs output from the code interpreter.

Hide properties

logs string
The logs output from the code interpreter.

type string
The type of the output. Always 'logs'.

Code interpreter output image object

The image output from the code interpreter.

Hide properties

type string
The type of the output. Always 'image'.

url string
The URL of the image output from the code interpreter.

status string
The status of the code interpreter tool call.

type string
The type of the code interpreter tool call. Always `code_interpreter_call`.

Local shell call object
A tool call to run a command on the local shell.

Hide properties

action object
Execute a shell command on the server.

Hide properties

command array
The command to run.

env object
Environment variables to set for the command.

type string
The type of the local shell action. Always `exec`.

timeout_ms integer or null
Optional timeout in milliseconds for the command.

user string or null
Optional user to run the command as.

working_directory string or null
Optional working directory to run the command in.

call_id string
The unique ID of the local shell tool call generated by the model.

id string
The unique ID of the local shell call.

status string
The status of the local shell call.

type string
The type of the local shell call. Always `local_shell_call`.

Local shell call output object
The output of a local shell tool call.

Hide properties

id string
The unique ID of the local shell tool call generated by the model.

output string
A JSON string of the output of the local shell tool call.

type string
The type of the local shell tool call output. Always `local_shell_call_output`.

status string or null
The status of the item. One of `in_progress`, `completed`, or `incomplete`.

MCP list tools object
A list of tools available on an MCP server.

Hide properties

id string
The unique ID of the list.

server_label string
The label of the MCP server.

tools array
The tools available on the server.

Hide properties

input_schema object
The JSON schema describing the tool's input.

name string
The name of the tool.

annotations object or null
Additional annotations about the tool.

description string or null
The description of the tool.

type string
The type of the item. Always `mcp_list_tools`.

error string or null
Error message if the server could not list tools.

MCP approval request object
A request for human approval of a tool invocation.

Hide properties

arguments string
A JSON string of arguments for the tool.

id string
The unique ID of the approval request.

name string
The name of the tool to run.

server_label string
The label of the MCP server making the request.

type string
The type of the item. Always `mcp_approval_request`.

MCP approval response object
A response to an MCP approval request.

Hide properties

approval_request_id string
The ID of the approval request being answered.

approve boolean
Whether the request was approved.

type string
The type of the item. Always `mcp_approval_response`.

id string or null
The unique ID of the approval response

reason string or null
Optional reason for the decision.

MCP tool call object
An invocation of a tool on an MCP server.

^ Hide properties

arguments string
A JSON string of the arguments passed to the tool.

id string
The unique ID of the tool call.

name string
The name of the tool that was run.

server_label string
The label of the MCP server running the tool.

type string
The type of the item. Always `mcp_call`.

error string or null
The error from the tool call, if any.

output string or null
The output from the tool call.

Item reference object

An internal identifier for an item to reference.

Hide properties

id string

The ID of the item to reference.

type string

The type of item to reference. Always `item_reference`.

metadata map

Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

model string

Model ID used to generate the response, like `gpt-4o` or `o3`. OpenAI offers a wide range of models with different capabilities, performance characteristics, and price points. Refer to the model guide to browse and compare available models.

object string

The object type of this resource - always set to `response`.

output array

An array of content items generated by the model.

The length and order of items in the `output` array is dependent on the model's response.

Rather than accessing the first item in the `output` array and assuming it's an `assistant` message with the content generated by the model, you might consider using the `output_text` property where supported in SDKs.

Hide possible types

Output message object

An output message from the model.

Hide properties

content array

The content of the output message.

Hide possible types

Output text object

A text output from the model.

Hide properties

annotations array

The annotations of the text output.

Hide possible types

File citation object

A citation to a file.

Hide properties

file_id string

The ID of the file.

filename string

The filename of the file cited.

index integer

The index of the file in the list of files.

type string

The type of the file citation. Always `file_citation`.

URL citation object

A citation for a web resource used to generate a model response.

Hide properties

end_index integer

The index of the last character of the URL citation in the message.

start_index integer

The index of the first character of the URL citation in the message.

title string

The title of the web resource.

type string

The type of the URL citation. Always `url_citation`.

url string
The URL of the web resource.

Container file citation object
A citation for a container file used to generate a model response.

Hide properties

container_id string
The ID of the container file.

end_index integer
The index of the last character of the container file citation in the message.

file_id string
The ID of the file.

filename string
The filename of the container file cited.

start_index integer
The index of the first character of the container file citation in the message.

type string
The type of the container file citation. Always `container_file_citation`.

File path object
A path to a file.

Hide properties

file_id string
The ID of the file.

index integer
The index of the file in the list of files.

type string
The type of the file path. Always `file_path`.

text string
The text output from the model.

type string
The type of the output text. Always `output_text` .

logprobs array
Hide properties

bytes array

logprob number

token string

top_logprobs array
Hide properties

bytes array

logprob number

token string

Refusal object
A refusal from the model.
Hide properties

refusal string
The refusal explanationfrom the model.

type string
The type of the refusal. Always `refusal` .

id string
The unique ID of the output message.

role string
The role of the output message. Always `assistant` .

status string
The status of the message input. One of `in_progress`, `completed`, or `incomplete`. Populated when input items are returned via API.

type string
The type of the output message. Always `message`.

File search tool call object
The results of a file search tool call. See the file search guide for more information.

Hide properties

id string
The unique ID of the file search tool call.

queries array
The queries used to search for files.

status string
The status of the file search tool call. One of `in_progress`, `searching`, `incomplete` or `failed`,

type string
The type of the file search tool call. Always `file_search_call`.

results array or null
The results of the file search tool call.

Hide properties

attributes map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard. Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters, booleans, or numbers.

file_id string
The unique ID of the file.

filename string
The name of the file.

score number
The relevance score of the file - a value between 0 and 1.

text string
The text that was retrieved from the file.

Function tool call object
A tool call to run a function. See the function calling guide for more information.

Hide properties

arguments string
A JSON string of the arguments to pass to the function.

call_id string
The unique ID of the function tool call generated by the model.

name string
The name of the function to run.

type string
The type of the function tool call. Always `function_call`.

id string
The unique ID of the function tool call.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Web search tool call object
The results of a web search tool call. See the web search guide for more information.

Hide properties

action object
An object describing the specific action taken in this web search call. Includes details on how the model used the web (search, open_page, find).

Hide possible types

Search action object
Action type "search" - Performs a web search query.

Hide properties

query string
The search query.

type string
The action type.

Open page action object
Action type "open_page" - Opens a specific URL from search results.

Hide properties

type string
The action type.

url string
The URL opened by the model.

Find action object
Action type "find": Searches for a pattern within a loaded page.

Hide properties

pattern string
The pattern or text to search for within the page.

type string
The action type.

url string
The URL of the page searched for the pattern.

id string
The unique ID of the web search tool call.

status string
The status of the web search tool call.

type string
The type of the web search tool call. Always `web_search_call` .

Computer tool call object

A tool call to a computer use tool. See the computer use guide for more information.

Hide properties

action object

Hide possible types

Click object

A click action.

Hide properties

button string

Indicates which mouse button was pressed during the click. One of `left`, `right`, `wheel`, `back`, or `forward`.

type string

Specifies the event type. For a click action, this property is always set to `click`.

x integer

The x-coordinate where the click occurred.

y integer

The y-coordinate where the click occurred.

DoubleClick object

A double click action.

Hide properties

type string

Specifies the event type. For a double click action, this property is always set to `double_click`.

x integer

The x-coordinate where the double click occurred.

y integer

The y-coordinate where the double click occurred.

Drag object

A drag action.

Hide properties

path array

An array of coordinates representing the path of the drag action. Coordinates will appear as an array of objects, eg

```
[
  { x: 100, y: 200 },
  { x: 200, y: 300 }
]
```

Hide properties

x integer

The x-coordinate.

y integer

The y-coordinate.

type string

Specifies the event type. For a drag action, this property is always set to `drag`.

KeyPress object

A collection of keypresses the model would like to perform.

Hide properties

keys array

The combination of keys the model is requesting to be pressed. This is an array of strings, each representing a key.

type string

Specifies the event type. For a keypress action, this property is always set to `keypress`.

Move object

A mouse move action.

Hide properties

type string

Specifies the event type. For a move action, this property is always set to `move`.

x integer

The x-coordinate to move to.

y integer
The y-coordinate to move to.

Screenshot object
A screenshot action.

Hide properties

type string
Specifies the event type. For a screenshot action, this property is always set to `screenshot`.

Scroll object
A scroll action.

Hide properties

scroll_x integer
The horizontal scroll distance.

scroll_y integer
The vertical scroll distance.

type string
Specifies the event type. For a scroll action, this property is always set to `scroll`.

x integer
The x-coordinate where the scroll occurred.

y integer
The y-coordinate where the scroll occurred.

Type object
An action to type in text.

Hide properties

text string
The text to type.

type string
Specifies the event type. For a type action, this property is always set to `type`.

Wait object

A wait action.

Hide properties

type string

Specifies the event type. For a wait action, this property is always set to `wait`.

call_id string

An identifier used when responding to the tool call with output.

id string

The unique ID of the computer call.

pending_safety_checks array

The pending safety checks for the computer call.

Hide properties

code string

The type of the pending safety check.

id string

The ID of the pending safety check.

message string

Details about the pending safety check.

status string

The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

type string

The type of the computer call. Always `computer_call`.

Reasoning object

A description of the chain of thought used by a reasoning model while generating a response. Be sure to include these items in your `input` to the Responses API for subsequent turns of a conversation if you are manually managing context.

Hide properties

id string

The unique identifier of the reasoning content.

summary array
Reasoning text contents.

Hide properties

text string
A short summary of the reasoning used by the model when generating the response.

type string
The type of the object. Always `summary_text`.

type string
The type of the object. Always `reasoning`.

encrypted_content string or null
The encrypted content of the reasoning item - populated when a response is generated with `reasoning.encrypted_content` in the `include` parameter.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Image generation call object
An image generation request made by the model.

Hide properties

id string
The unique ID of the image generation call.

result string or null
The generated image encoded in base64.

status string
The status of the image generation call.

type string
The type of the image generation call. Always `image_generation_call`.

Code interpreter tool call object
A tool call to run code.

Hide properties

code string or null
The code to run, or null if not available.

container_id string
The ID of the container used to run the code.

id string
The unique ID of the code interpreter tool call.

outputs array or null
The outputs generated by the code interpreter, such as logs or images. Can be null if no outputs are available.

Hide possible types

Code interpreter output logs object
The logs output from the code interpreter.

Hide properties

logs string
The logs output from the code interpreter.

type string
The type of the output. Always 'logs'.

Code interpreter output image object
The image output from the code interpreter.

Hide properties

type string
The type of the output. Always 'image'.

url string
The URL of the image output from the code interpreter.

status string
The status of the code interpreter tool call.

type string
The type of the code interpreter tool call. Always `code_interpreter_call`.

Local shell call object

A tool call to run a command on the local shell.

Hide properties

action object

Execute a shell command on the server.

Hide properties

command array

The command to run.

env object

Environment variables to set for the command.

type string

The type of the local shell action. Always `exec`.

timeout_ms integer or null

Optional timeout in milliseconds for the command.

user string or null

Optional user to run the command as.

working_directory string or null

Optional working directory to run the command in.

call_id string

The unique ID of the local shell tool call generated by the model.

id string

The unique ID of the local shell call.

status string

The status of the local shell call.

type string

The type of the local shell call. Always `local_shell_call`.

MCP tool call object

An invocation of a tool on an MCP server.

Hide properties

arguments string
A JSON string of the arguments passed to the tool.

id string
The unique ID of the tool call.

name string
The name of the tool that was run.

server_label string
The label of the MCP server running the tool.

type string
The type of the item. Always `mcp_call`.

error string or null
The error from the tool call, if any.

output string or null
The output from the tool call.

MCP list tools object
A list of tools available on an MCP server.

Hide properties

id string
The unique ID of the list.

server_label string
The label of the MCP server.

tools array
The tools available on the server.

Hide properties

input_schema object
The JSON schema describing the tool's input.

name string
The name of the tool.

annotations object or null

Additional annotations about the tool.

description string or null
The description of the tool.

type string
The type of the item. Always `mcp_list_tools`.

error string or null
Error message if the server could not list tools.

MCP approval request object
A request for human approval of a tool invocation.

Hide properties

arguments string
A JSON string of arguments for the tool.

id string
The unique ID of the approval request.

name string
The name of the tool to run.

server_label string
The label of the MCP server making the request.

type string
The type of the item. Always `mcp_approval_request`.

parallel_tool_calls boolean
Whether to allow the model to run tool calls in parallel.

temperature number or null
What sampling temperature to use, between 0 and 2. Higher values like 0.8 will make the output more random, while lower values like 0.2 will make it more focused and deterministic. We generally recommend altering this or `top_p` but not both.

tool_choice string or object
How the model should select which tool (or tools) to use when generating a response. See the `tools`

parameter to see how to specify which tools the model can call.

Hide possible types

Tool choice mode string

Controls which (if any) tool is called by the model.

`none` means the model will not call any tool and instead generates a message.

`auto` means the model can pick between generating a message or calling one or more tools.

`required` means the model must call one or more tools.

Hosted tool object

Indicates that the model should use a built-in tool to generate a response. Learn more about built-in tools.

Hide properties

type string

The type of hosted tool the model should to use. Learn more about built-in tools.

Allowed values are:

`file_search`

`web_search_preview`

`computer_use_preview`

`code_interpreter`

`image_generation`

Function tool object

Use this option to force the model to call a specific function.

Hide properties

name string

The name of the function to call.

type string

For function calling, the type is always `function`.

MCP tool object

Use this option to force the model to call a specific tool on a remote MCP server.

Hide properties

server_label string

The label of the MCP server to use.

type string
For MCP tools, the type is always `mcp`.

name string or null
The name of the tool to call on the server.

tools array
An array of tools the model may call while generating a response. You can specify which tool to use by setting the `tool_choice` parameter.

The two categories of tools you can provide the model are:

Built-in tools: Tools that are provided by OpenAI that extend the model's capabilities, like web search or file search. Learn more about built-in tools.

Function calls (custom tools): Functions that are defined by you, enabling the model to call your own code. Learn more about function calling.

Hide possible types

Function object
Defines a function in your own code the model can choose to call. Learn more about function calling.

Hide properties

name string
The name of the function to call.

parameters object
A JSON schema object describing the parameters of the function.

strict boolean
Whether to enforce strict parameter validation. Default `true`.

type string
The type of the function tool. Always `function`.

description string
A description of the function. Used by the model to determine whether or not to call the function.

File search object
A tool that searches for relevant content from uploaded files. Learn more about the file search tool.

Hide properties

type string
The type of the file search tool. Always `file_search`.

vector_store_ids array
The IDs of the vector stores to search.

filters object
A filter to apply.

Hide possible types

Comparison Filter object
A filter used to compare a specified attribute key to a given value using a defined comparison operation.

Hide properties

key string
The key to compare against the value.

type string
Specifies the comparison operator: `eq`, `ne`, `gt`, `gte`, `lt`, `lte`.

- `eq`: equals
- `ne`: not equal
- `gt`: greater than
- `gte`: greater than or equal
- `lt`: less than
- `lte`: less than or equal

value string / number / boolean
The value to compare against the attribute key; supports string, number, or boolean types.

Compound Filter object
Combine multiple filters using `and` or `or`.

Hide properties

filters array
Array of filters to combine. Items can be `ComparisonFilter` or `CompoundFilter`.

Hide possible types

Comparison Filter object
A filter used to compare a specified attribute key to a given value using a defined comparison operation.

Hide properties

key string
The key to compare against the value.

type string
Specifies the comparison operator: `eq`, `ne`, `gt`, `gte`, `lt`, `lte`.

- `eq`: equals
- `ne`: not equal
- `gt`: greater than
- `gte`: greater than or equal
- `lt`: less than
- `lte`: less than or equal

value string / number / boolean
The value to compare against the attribute key; supports string, number, or boolean types.

type string
Type of operation: `and` or `or`.

max_num_results integer
The maximum number of results to return. This number should be between 1 and 50 inclusive.

ranking_options object
Ranking options for search.

Hide properties

ranker string
The ranker to use for the file search.

score_threshold number

The score threshold for the file search, a number between 0 and 1. Numbers closer to 1 will attempt to return only the most relevant results, but may return fewer results.

Web search preview object

This tool searches the web for relevant results to use in a response. Learn more about the web search tool.

Hide properties

type string

The type of the web search tool. One of `web_search_preview` or `web_search_preview_2025_03_11`.

search_context_size string

High level guidance for the amount of context window space to use for the search. One of `low`, `medium`, or `high`. `medium` is the default.

user_location object

The user's location.

Hide properties

type string

The type of location approximation. Always `approximate`.

city string

Free text input for the city of the user, e.g. `San Francisco`.

country string

The two-letter ISO country code of the user, e.g. `US`.

region string

Free text input for the region of the user, e.g. `California`.

timezone string

The IANA timezone of the user, e.g. `America/Los_Angeles`.

Computer use preview object

A tool that controls a virtual computer. Learn more about the computer tool.

Hide properties

display_height integer
The height of the computer display.

display_width integer
The width of the computer display.

environment string
The type of computer environment to control.

type string
The type of the computer use tool. Always `computer_use_preview`.

MCP tool object
Give the model access to additional tools via remote Model Context Protocol (MCP) servers. Learn more about MCP.

Hide properties

server_label string
A label for this MCP server, used to identify it in tool calls.

server_url string
The URL for the MCP server.

type string
The type of the MCP tool. Always `mcp`.

allowed_tools array or object
List of allowed tool names or a filter object.

Hide possible types

MCP allowed tools array
A string array of allowed tool names

MCP allowed tools filter object
A filter object to specify which tools are allowed.

Hide properties

tool_names array
List of allowed tool names.

headers object or null

Optional HTTP headers to send to the MCP server. Use for authentication or other purposes.

`require_approval` object or string

Specify which of the MCP server's tools require approval.

Hide possible types

MCP tool approval filter object

Hide properties

`always` object

A list of tools that always require approval.

Hide properties

`tool_names` array

List of tools that require approval.

`never` object

A list of tools that never require approval.

Hide properties

`tool_names` array

List of tools that do not require approval.

MCP tool approval setting string

Specify a single approval policy for all tools. One of `always` or `never`. When set to `always`, all tools will require approval. When set to `never`, all tools will not require approval.

`server_description` string

Optional description of the MCP server, used to provide more context.

Code interpreter object

A tool that runs Python code to help generate a response to a prompt.

Hide properties

`container` string or object

The code interpreter container. Can be a container ID or an object that specifies uploaded file IDs to make available to your code.

Hide possible types

string
The container ID.

CodeInterpreterContainerAuto object
Configuration for a code interpreter container. Optionally specify the IDs of the files to run the code on.

Hide properties

type string
Always `auto`.

file_ids array
An optional list of uploaded files to make available to your code.

type string
The type of the code interpreter tool. Always `code_interpreter`.

Image generation tool object
A tool that generates images using a model like `gpt-image-1`.

Hide properties

type string
The type of the image generation tool. Always `image_generation`.

background string
Background type for the generated image. One of `transparent`, `opaque`, or `auto`. Default: `auto`.

input_image_mask object
Optional mask for inpainting. Contains `image_url` (string, optional) and `file_id` (string, optional).

Hide properties

file_id string
File ID for the mask image.

image_url string
Base64-encoded mask image.

model string

The image generation model to use. Default: `gpt-image-1`.

`moderation` string
Moderation level for the generated image. Default: `auto`.

`output_compression` integer
Compression level for the output image. Default: 100.

`output_format` string
The output format of the generated image. One of `png`, `webp`, or `jpeg`. Default: `png`.

`partial_images` integer
Number of partial images to generate in streaming mode, from 0 (default value) to 3.

`quality` string
The quality of the generated image. One of `low`, `medium`, `high`, or `auto`. Default: `auto`.

`size` string
The size of the generated image. One of `1024x1024`, `1024x1536`, `1536x1024`, or `auto`. Default: `auto`.

Local shell tool object
A tool that allows the model to execute shell commands in a local environment.

Hide properties

`type` string
The type of the local shell tool. Always `local_shell`.

`top_p` number or null
An alternative to sampling with temperature, called nucleus sampling, where the model considers the results of the tokens with top_p probability mass. So 0.1 means only the tokens comprising the top 10% probability mass are considered.

We generally recommend altering this or `temperature` but not both.

`background` boolean or null
Whether to run the model response in the background. Learn more.

`max_output_tokens` integer or null
An upper bound for the number of tokens that can be generated for a response, including visible output tokens and reasoning tokens.

max_tool_calls integer or null
The maximum number of total calls to built-in tools that can be processed in a response. This maximum number applies across all built-in tool calls, not per individual tool. Any further attempts to call a tool by the model will be ignored.

output_text string or null SDK Only
SDK-only convenience property that contains the aggregated text output from all `output_text` items in the `output` array, if any are present. Supported in the Python and JavaScript SDKs.

previous_response_id string or null
The unique ID of the previous response to the model. Use this to create multi-turn conversations. Learn more about conversation state.

prompt object or null
Reference to a prompt template and its variables. Learn more.

Hide properties

id string
The unique identifier of the prompt template to use.

variables map
Optional map of values to substitute in for variables in your prompt. The substitution values can either be strings, or other Response input types like images or files.

version string or null
Optional version of the prompt template.

reasoning object or null
o-series models only

Configuration options for reasoning models.

Hide properties

effort string or null
o-series models only

Constrains effort on reasoning for reasoning models. Currently supported values are `low`, `medium`, and `high`. Reducing reasoning effort can result in faster responses and fewer tokens used on reasoning in a response.

generate_summary Deprecated string or null
Deprecated: use `summary` instead.

A summary of the reasoning performed by the model. This can be useful for debugging and

understanding the model's reasoning process. One of `auto`, `concise`, or `detailed`.

`summary` string or null
A summary of the reasoning performed by the model. This can be useful for debugging and understanding the model's reasoning process. One of `auto`, `concise`, or `detailed`.

`service_tier` string or null
Specifies the processing type used for serving the request.

> If set to 'auto', then the request will be processed with the service tier configured in the Project settings. Unless otherwise configured, the Project will use 'default'.
>
> If set to 'default', then the requset will be processed with the standard pricing and performance for the selected model.
>
> If set to 'flex' or 'priority', then the request will be processed with the corresponding service tier. Contact sales to learn more about Priority processing.
>
> When not set, the default behavior is 'auto'.

When the `service_tier` parameter is set, the response body will include the `service_tier` value based on the processing mode actually used to serve the request. This response value may be different from the value set in the parameter.

`status` string
The status of the response generation. One of `completed`, `failed`, `in_progress`, `cancelled`, `queued`, or `incomplete`.

`text` object
Configuration options for a text response from the model. Can be plain text or structured JSON data. Learn more:

> Text inputs and outputs
>
> Structured Outputs

Hide properties

`format` object
An object specifying the format that the model must output.

Configuring `{ "type": "json_schema" }` enables Structured Outputs, which ensures the model will match your supplied JSON schema. Learn more in the Structured Outputs guide.

The default format is `{ "type": "text" }` with no additional options.

Not recommended for gpt-4o and newer models:

Setting to `{ "type": "json_object" }` enables the older JSON mode, which ensures the message the model generates is valid JSON. Using `json_schema` is preferred for models that support it.

Hide possible types

Text object
Default response format. Used to generate text responses.

Hide properties

type string
The type of response format being defined. Always `text`.

JSON schema object
JSON Schema response format. Used to generate structured JSON responses. Learn more about Structured Outputs.

Hide properties

name string
The name of the response format. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

schema object
The schema for the response format, described as a JSON Schema object. Learn how to build JSON schemas here.

type string
The type of response format being defined. Always `json_schema`.

description string
A description of what the response format is for, used by the model to determine how to respond in the format.

strict boolean or null
Whether to enable strict schema adherence when generating the output. If set to true, the model will always follow the exact schema defined in the `schema` field. Only a subset of JSON Schema is supported when `strict` is `true`. To learn more, read the Structured Outputs guide.

JSON object object
JSON object response format. An older method of generating JSON responses. Using `json_schema` is recommended for models that support it. Note that the model will not generate JSON without a system or user message instructing it to do so.

Hide properties

type string

The type of response format being defined. Always `json_object`.

`top_logprobs` integer or null
An integer between 0 and 20 specifying the number of most likely tokens to return at each token position, each with an associated log probability.

`truncation` string or null
The truncation strategy to use for the model response.

`auto`: If the context of this response and previous ones exceeds the model's context window size, the model will truncate the response to fit the context window by dropping input items in the middle of the conversation.

`disabled` (default): If a model response will exceed the context window size for a model, the request will fail with a 400 error.

`usage` object
Represents token usage details including input tokens, output tokens, a breakdown of output tokens, and the total tokens used.

Hide properties

`input_tokens` integer
The number of input tokens.

`input_tokens_details` object
A detailed breakdown of the input tokens.

Hide properties

`cached_tokens` integer
The number of tokens that were retrieved from the cache. More on prompt caching.

`output_tokens` integer
The number of output tokens.

`output_tokens_details` object
A detailed breakdown of the output tokens.

Hide properties

`reasoning_tokens` integer
The number of reasoning tokens.

total_tokens integer
The total number of tokens used.

user string
A stable identifier for your end-users. Used to boost cache hit rates by better bucketing similar requests and to help OpenAI detect and prevent abuse. Learn more.

sequence_number integer
The sequence number of this event.

type string
The type of the event. Always `response.incomplete`.

OBJECT response.incomplete

```
{
  "type": "response.incomplete",
  "response": {
    "id": "resp_123",
    "object": "response",
    "created_at": 1740855869,
    "status": "incomplete",
    "error": null,
    "incomplete_details": {
      "reason": "max_tokens"
    },
    "instructions": null,
    "max_output_tokens": null,
    "model": "gpt-4o-mini-2024-07-18",
    "output": [],
    "previous_response_id": null,
    "reasoning_effort": null,
    "store": false,
    "temperature": 1,
    "text": {
      "format": {
        "type": "text"
      }
    },
    "tool_choice": "auto",
    "tools": [],
    "top_p": 1,
    "truncation": "disabled",
```

```
    "usage": null,
    "user": null,
    "metadata": {}
  },
  "sequence_number": 1
}
```

response.output_item.added

Emitted when a new output item is added.

item object

The output item that was added.

Hide possible types

Output message object

An output message from the model.

Hide properties

content array

The content of the output message.

Hide possible types

Output text object

A text output from the model.

Hide properties

annotations array

The annotations of the text output.

Hide possible types

File citation object

A citation to a file.

Hide properties

file_id string

The ID of the file.

filename string

The filename of the file cited.

index integer
The index of the file in the list of files.

type string
The type of the file citation. Always `file_citation` .

URL citation object
A citation for a web resource used to generate a model response.

Hide properties

end_index integer
The index of the last character of the URL citation in the message.

start_index integer
The index of the first character of the URL citation in the message.

title string
The title of the web resource.

type string
The type of the URL citation. Always `url_citation` .

url string
The URL of the web resource.

Container file citation object
A citation for a container file used to generate a model response.

Hide properties

container_id string
The ID of the container file.

end_index integer
The index of the last character of the container file citation in the message.

file_id string
The ID of the file.

filename string
The filename of the container file cited.

start_index integer
The index of the first character of the container file citation in the message.

type string
The type of the container file citation. Always `container_file_citation` .

File path object
A path to a file.

Hide properties

file_id string
The ID of the file.

index integer
The index of the file in the list of files.

type string
The type of the file path. Always `file_path` .

text string
The text output from the model.

type string
The type of the output text. Always `output_text` .

logprobs array

Hide properties

bytes array

logprob number

token string

top_logprobs array

Hide properties

bytes array

logprob number

token string

Refusal object

A refusal from the model.

Hide properties

refusal string

The refusal explanationfrom the model.

type string

The type of the refusal. Always `refusal`.

id string

The unique ID of the output message.

role string

The role of the output message. Always `assistant`.

status string

The status of the message input. One of `in_progress`, `completed`, or `incomplete`. Populated when input items are returned via API.

type string

The type of the output message. Always `message`.

File search tool call object

The results of a file search tool call. See the file search guide for more information.

Hide properties

id string

The unique ID of the file search tool call.

queries array

The queries used to search for files.

status string
The status of the file search tool call. One of `in_progress`, `searching`, `incomplete` or `failed`,

type string
The type of the file search tool call. Always `file_search_call`.

results array or null
The results of the file search tool call.

Hide properties

attributes map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard. Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters, booleans, or numbers.

file_id string
The unique ID of the file.

filename string
The name of the file.

score number
The relevance score of the file - a value between 0 and 1.

text string
The text that was retrieved from the file.

Function tool call object
A tool call to run a function. See the function calling guide for more information.

Hide properties

arguments string
A JSON string of the arguments to pass to the function.

call_id string
The unique ID of the function tool call generated by the model.

name string
The name of the function to run.

type string
The type of the function tool call. Always `function_call`.

id string
The unique ID of the function tool call.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Web search tool call object
The results of a web search tool call. See the web search guide for more information.

Hide properties

action object
An object describing the specific action taken in this web search call. Includes details on how the model used the web (search, open_page, find).

Hide possible types

Search action object
Action type "search" - Performs a web search query.

Hide properties

query string
The search query.

type string
The action type.

Open page action object
Action type "open_page" - Opens a specific URL from search results.

Hide properties

type string
The action type.

url string
The URL opened by the model.

Find action object
Action type "find": Searches for a pattern within a loaded page.

Hide properties

pattern string
The pattern or text to search for within the page.

type string
The action type.

url string
The URL of the page searched for the pattern.

id string
The unique ID of the web search tool call.

status string
The status of the web search tool call.

type string
The type of the web search tool call. Always `web_search_call`.

Computer tool call object
A tool call to a computer use tool. See the computer use guide for more information.

Hide properties

action object

Hide possible types

Click object
A click action.

Hide properties

button string
Indicates which mouse button was pressed during the click. One of `left`, `right`, `wheel`, `back`, or `forward`.

type string
Specifies the event type. For a click action, this property is always set to `click`.

x integer
The x-coordinate where the click occurred.

y integer

The y-coordinate where the click occurred.

DoubleClick object

A double click action.

Hide properties

type string

Specifies the event type. For a double click action, this property is always set to `double_click`.

x integer

The x-coordinate where the double click occurred.

y integer

The y-coordinate where the double click occurred.

Drag object

A drag action.

Hide properties

path array

An array of coordinates representing the path of the drag action. Coordinates will appear as an array of objects, eg

```
[
  { x: 100, y: 200 },
  { x: 200, y: 300 }
]
```

Hide properties

x integer

The x-coordinate.

y integer

The y-coordinate.

type string

Specifies the event type. For a drag action, this property is always set to `drag`.

KeyPress object

A collection of keypresses the model would like to perform.

Hide properties

keys array

The combination of keys the model is requesting to be pressed. This is an array of strings, each representing a key.

type string

Specifies the event type. For a keypress action, this property is always set to `keypress`.

Move object

A mouse move action.

Hide properties

type string

Specifies the event type. For a move action, this property is always set to `move`.

x integer
The x-coordinate to move to.

y integer
The y-coordinate to move to.

Screenshot object
A screenshot action.

Hide properties

type string
Specifies the event type. For a screenshot action, this property is always set to `screenshot`.

Scroll object
A scroll action.

Hide properties

scroll_x integer
The horizontal scroll distance.

scroll_y integer
The vertical scroll distance.

type string
Specifies the event type. For a scroll action, this property is always set to `scroll`.

x integer
The x-coordinate where the scroll occurred.

y integer
The y-coordinate where the scroll occurred.

Type object
An action to type in text.

Hide properties

text string
The text to type.

type string
Specifies the event type. For a type action, this property is always set to `type`.

Wait object
A wait action.

Hide properties

type string
Specifies the event type. For a wait action, this property is always set to `wait`.

call_id string
An identifier used when responding to the tool call with output.

id string
The unique ID of the computer call.

pending_safety_checks array
The pending safety checks for the computer call.

Hide properties

code string
The type of the pending safety check.

id string
The ID of the pending safety check.

message string
Details about the pending safety check.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

type string
The type of the computer call. Always `computer_call`.

Reasoning object
A description of the chain of thought used by a reasoning model while generating a response. Be sure to include these items in your `input` to the Responses API for subsequent turns of a conversation if you

are manually managing context.

Hide properties

id string
The unique identifier of the reasoning content.

summary array
Reasoning text contents.

Hide properties

text string
A short summary of the reasoning used by the model when generating the response.

type string
The type of the object. Always `summary_text`.

type string
The type of the object. Always `reasoning`.

encrypted_content string or null
The encrypted content of the reasoning item - populated when a response is generated with `reasoning.encrypted_content` in the `include` parameter.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Image generation call object
An image generation request made by the model.

Hide properties

id string
The unique ID of the image generation call.

result string or null
The generated image encoded in base64.

status string
The status of the image generation call.

type string
The type of the image generation call. Always `image_generation_call`.

Code interpreter tool call object
A tool call to run code.

Hide properties

code string or null
The code to run, or null if not available.

container_id string
The ID of the container used to run the code.

id string
The unique ID of the code interpreter tool call.

outputs array or null
The outputs generated by the code interpreter, such as logs or images. Can be null if no outputs are available.

Hide possible types

Code interpreter output logs object
The logs output from the code interpreter.

Hide properties

logs string
The logs output from the code interpreter.

type string
The type of the output. Always 'logs'.

Code interpreter output image object
The image output from the code interpreter.

Hide properties

type string
The type of the output. Always 'image'.

url string
The URL of the image output from the code interpreter.

status string
The status of the code interpreter tool call.

type string
The type of the code interpreter tool call. Always `code_interpreter_call`.

Local shell call object
A tool call to run a command on the local shell.

Hide properties

action object
Execute a shell command on the server.

Hide properties

command array
The command to run.

env object
Environment variables to set for the command.

type string
The type of the local shell action. Always `exec`.

timeout_ms integer or null
Optional timeout in milliseconds for the command.

user string or null
Optional user to run the command as.

working_directory string or null
Optional working directory to run the command in.

call_id string
The unique ID of the local shell tool call generated by the model.

id string
The unique ID of the local shell call.

status string
The status of the local shell call.

type string
The type of the local shell call. Always `local_shell_call` .

MCP tool call object
An invocation of a tool on an MCP server.

Hide properties

arguments string
A JSON string of the arguments passed to the tool.

id string
The unique ID of the tool call.

name string
The name of the tool that was run.

server_label string
The label of the MCP server running the tool.

type string
The type of the item. Always `mcp_call` .

error string or null
The error from the tool call, if any.

output string or null
The output from the tool call.

MCP list tools object
A list of tools available on an MCP server.

Hide properties

id string
The unique ID of the list.

server_label string
The label of the MCP server.

tools array
The tools available on the server.

Hide properties

input_schema object
The JSON schema describing the tool's input.

name string
The name of the tool.

annotations object or null
Additional annotations about the tool.

description string or null
The description of the tool.

type string
The type of the item. Always `mcp_list_tools`.

error string or null
Error message if the server could not list tools.

MCP approval request object
A request for human approval of a tool invocation.

Hide properties

arguments string
A JSON string of arguments for the tool.

id string
The unique ID of the approval request.

name string
The name of the tool to run.

server_label string
The label of the MCP server making the request.

type string
The type of the item. Always `mcp_approval_request`.

output_index integer
The index of the output item that was added.

sequence_number integer
The sequence number of this event.

type string
The type of the event. Always `response.output_item.added`.

OBJECT response.output_item.added

```
{
  "type": "response.output_item.added",
  "output_index": 0,
  "item": {
    "id": "msg_123",
    "status": "in_progress",
    "type": "message",
    "role": "assistant",
    "content": []
  },
  "sequence_number": 1
}
```

response.output_item.done

Emitted when an output item is marked done.

item object
The output item that was marked done.

Hide possible types

Output message object
An output message from the model.

Hide properties

content array
The content of the output message.

Hide possible types

Output text object
A text output from the model.

Hide properties

annotations array

The annotations of the text output.

Hide possible types

File citation object

A citation to a file.

Hide properties

file_id string

The ID of the file.

filename string

The filename of the file cited.

index integer

The index of the file in the list of files.

type string

The type of the file citation. Always `file_citation` .

URL citation object

A citation for a web resource used to generate a model response.

Hide properties

end_index integer

The index of the last character of the URL citation in the message.

start_index integer

The index of the first character of the URL citation in the message.

title string

The title of the web resource.

type string

The type of the URL citation. Always `url_citation` .

url string

The URL of the web resource.

Container file citation object

A citation for a container file used to generate a model response.

Hide properties

container_id string
The ID of the container file.

end_index integer
The index of the last character of the container file citation in the message.

file_id string
The ID of the file.

filename string
The filename of the container file cited.

start_index integer
The index of the first character of the container file citation in the message.

type string
The type of the container file citation. Always `container_file_citation`.

File path object
A path to a file.

Hide properties

file_id string
The ID of the file.

index integer
The index of the file in the list of files.

type string
The type of the file path. Always `file_path`.

text string
The text output from the model.

type string
The type of the output text. Always `output_text`.

logprobs array

Hide properties

bytes array

logprob number

token string

top_logprobs array

Hide properties

bytes array

logprob number

token string

Refusal object

A refusal from the model.

Hide properties

refusal string

The refusal explanationfrom the model.

type string

The type of the refusal. Always `refusal`.

id string

The unique ID of the output message.

role string

The role of the output message. Always `assistant`.

status string

The status of the message input. One of `in_progress`, `completed`, or `incomplete`. Populated when input items are returned via API.

type string

The type of the output message. Always `message`.

File search tool call object

The results of a file search tool call. See the file search guide for more information.

Hide properties

id string
The unique ID of the file search tool call.

queries array
The queries used to search for files.

status string
The status of the file search tool call. One of `in_progress`, `searching`, `incomplete` or `failed`,

type string
The type of the file search tool call. Always `file_search_call`.

results array or null
The results of the file search tool call.

Hide properties

attributes map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard. Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters, booleans, or numbers.

file_id string
The unique ID of the file.

filename string
The name of the file.

score number
The relevance score of the file - a value between 0 and 1.

text string
The text that was retrieved from the file.

Function tool call object
A tool call to run a function. See the function calling guide for more information.

Hide properties

arguments string

A JSON string of the arguments to pass to the function.

call_id string
The unique ID of the function tool call generated by the model.

name string
The name of the function to run.

type string
The type of the function tool call. Always `function_call`.

id string
The unique ID of the function tool call.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Web search tool call object
The results of a web search tool call. See the web search guide for more information.

Hide properties

action object
An object describing the specific action taken in this web search call. Includes details on how the model used the web (search, open_page, find).

Hide possible types

Search action object
Action type "search" - Performs a web search query.

Hide properties

query string
The search query.

type string
The action type.

Open page action object
Action type "open_page" - Opens a specific URL from search results.

Hide properties

type string

The action type.

url string
The URL opened by the model.

Find action object
Action type "find": Searches for a pattern within a loaded page.

Hide properties

pattern string
The pattern or text to search for within the page.

type string
The action type.

url string
The URL of the page searched for the pattern.

id string
The unique ID of the web search tool call.

status string
The status of the web search tool call.

type string
The type of the web search tool call. Always `web_search_call`.

Computer tool call object
A tool call to a computer use tool. See the computer use guide for more information.

Hide properties

action object

Hide possible types

Click object
A click action.

Hide properties

button string
Indicates which mouse button was pressed during the click. One of `left`, `right`, `wheel`, `back`, or `forward`.

type string
Specifies the event type. For a click action, this property is always set to `click`.

x integer
The x-coordinate where the click occurred.

y integer
The y-coordinate where the click occurred.

DoubleClick object
A double click action.

Hide properties

type string
Specifies the event type. For a double click action, this property is always set to `double_click`.

x integer
The x-coordinate where the double click occurred.

y integer
The y-coordinate where the double click occurred.

Drag object
A drag action.

Hide properties

path array
An array of coordinates representing the path of the drag action. Coordinates will appear as an array of objects, eg

```
[
  { x: 100, y: 200 },
  { x: 200, y: 300 }
]
```

Hide properties

x integer
The x-coordinate.

y integer
The y-coordinate.

type string
Specifies the event type. For a drag action, this property is always set to `drag`.

KeyPress object
A collection of keypresses the model would like to perform.

Hide properties

keys array
The combination of keys the model is requesting to be pressed. This is an array of strings, each representing a key.

type string
Specifies the event type. For a keypress action, this property is always set to `keypress`.

Move object
A mouse move action.

Hide properties

type string
Specifies the event type. For a move action, this property is always set to `move`.

x integer
The x-coordinate to move to.

y integer
The y-coordinate to move to.

Screenshot object
A screenshot action.

Hide properties

type string
Specifies the event type. For a screenshot action, this property is always set to `screenshot`.

Scroll object
A scroll action.

Hide properties

scroll_x integer
The horizontal scroll distance.

scroll_y integer
The vertical scroll distance.

type string
Specifies the event type. For a scroll action, this property is always set to `scroll`.

x integer
The x-coordinate where the scroll occurred.

y integer
The y-coordinate where the scroll occurred.

Type object
An action to type in text.

Hide properties

text string
The text to type.

type string
Specifies the event type. For a type action, this property is always set to `type`.

Wait object
A wait action.

Hide properties

type string
Specifies the event type. For a wait action, this property is always set to `wait`.

call_id string

An identifier used when responding to the tool call with output.

id string
The unique ID of the computer call.

pending_safety_checks array
The pending safety checks for the computer call.

Hide properties

code string
The type of the pending safety check.

id string
The ID of the pending safety check.

message string
Details about the pending safety check.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

type string
The type of the computer call. Always `computer_call`.

Reasoning object
A description of the chain of thought used by a reasoning model while generating a response. Be sure to include these items in your `input` to the Responses API for subsequent turns of a conversation if you are manually managing context.

Hide properties

id string
The unique identifier of the reasoning content.

summary array
Reasoning text contents.

Hide properties

text string
A short summary of the reasoning used by the model when generating the response.

type string

The type of the object. Always `summary_text`.

type string
The type of the object. Always `reasoning`.

encrypted_content string or null
The encrypted content of the reasoning item - populated when a response is generated with `reasoning.encrypted_content` in the `include` parameter.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Image generation call object
An image generation request made by the model.

Hide properties

id string
The unique ID of the image generation call.

result string or null
The generated image encoded in base64.

status string
The status of the image generation call.

type string
The type of the image generation call. Always `image_generation_call`.

Code interpreter tool call object
A tool call to run code.

Hide properties

code string or null
The code to run, or null if not available.

container_id string
The ID of the container used to run the code.

id string
The unique ID of the code interpreter tool call.

outputs array or null
The outputs generated by the code interpreter, such as logs or images. Can be null if no outputs are available.

Hide possible types

Code interpreter output logs object
The logs output from the code interpreter.

Hide properties

logs string
The logs output from the code interpreter.

type string
The type of the output. Always 'logs'.

Code interpreter output image object
The image output from the code interpreter.

Hide properties

type string
The type of the output. Always 'image'.

url string
The URL of the image output from the code interpreter.

status string
The status of the code interpreter tool call.

type string
The type of the code interpreter tool call. Always `code_interpreter_call`.

Local shell call object
A tool call to run a command on the local shell.

Hide properties

action object
Execute a shell command on the server.

Hide properties

command array
The command to run.

env object
Environment variables to set for the command.

type string
The type of the local shell action. Always `exec`.

timeout_ms integer or null
Optional timeout in milliseconds for the command.

user string or null
Optional user to run the command as.

working_directory string or null
Optional working directory to run the command in.

call_id string
The unique ID of the local shell tool call generated by the model.

id string
The unique ID of the local shell call.

status string
The status of the local shell call.

type string
The type of the local shell call. Always `local_shell_call`.

MCP tool call object
An invocation of a tool on an MCP server.

Hide properties

arguments string
A JSON string of the arguments passed to the tool.

id string
The unique ID of the tool call.

name string
The name of the tool that was run.

server_label string
The label of the MCP server running the tool.

type string
The type of the item. Always `mcp_call`.

error string or null
The error from the tool call, if any.

output string or null
The output from the tool call.

MCP list tools object
A list of tools available on an MCP server.

Hide properties

id string
The unique ID of the list.

server_label string
The label of the MCP server.

tools array
The tools available on the server.

Hide properties

input_schema object
The JSON schema describing the tool's input.

name string
The name of the tool.

annotations object or null
Additional annotations about the tool.

description string or null
The description of the tool.

type string
The type of the item. Always `mcp_list_tools`.

error string or null
Error message if the server could not list tools.

MCP approval request object
A request for human approval of a tool invocation.

Hide properties

arguments string
A JSON string of arguments for the tool.

id string
The unique ID of the approval request.

name string
The name of the tool to run.

server_label string
The label of the MCP server making the request.

type string
The type of the item. Always `mcp_approval_request`.

output_index integer
The index of the output item that was marked done.

sequence_number integer
The sequence number of this event.

type string
The type of the event. Always `response.output_item.done`.

OBJECT response.output_item.done

```
{
  "type": "response.output_item.done",
  "output_index": 0,
  "item": {
```

```
    "id": "msg_123",
    "status": "completed",
    "type": "message",
    "role": "assistant",
    "content": [
      {
        "type": "output_text",
        "text": "In a shimmering forest under a sky full of stars, a lonely unicorn named
        "annotations": []
      }
    ]
  },
  "sequence_number": 1
}
```

response.content_part.added

Emitted when a new content part is added.

content_index integer
The index of the content part that was added.

item_id string
The ID of the output item that the content part was added to.

output_index integer
The index of the output item that the content part was added to.

part object
The content part that was added.

Hide possible types

Output text object
A text output from the model.

Hide properties

annotations array
The annotations of the text output.

Hide possible types

File citation object

A citation to a file.

Hide properties

file_id string

The ID of the file.

filename string

The filename of the file cited.

index integer

The index of the file in the list of files.

type string

The type of the file citation. Always `file_citation` .

URL citation object

A citation for a web resource used to generate a model response.

Hide properties

end_index integer

The index of the last character of the URL citation in the message.

start_index integer

The index of the first character of the URL citation in the message.

title string

The title of the web resource.

type string

The type of the URL citation. Always `url_citation` .

url string

The URL of the web resource.

Container file citation object

A citation for a container file used to generate a model response.

Hide properties

container_id string

The ID of the container file.

end_index integer
The index of the last character of the container file citation in the message.

file_id string
The ID of the file.

filename string
The filename of the container file cited.

start_index integer
The index of the first character of the container file citation in the message.

type string
The type of the container file citation. Always `container_file_citation` .

File path object
A path to a file.

Hide properties

file_id string
The ID of the file.

index integer
The index of the file in the list of files.

type string
The type of the file path. Always `file_path` .

text string
The text output from the model.

type string
The type of the output text. Always `output_text` .

logprobs array

Hide properties

bytes array

logprob number

token string

top_logprobs array

Hide properties

bytes array

logprob number

token string

Refusal object

A refusal from the model.

Hide properties

refusal string

The refusal explanationfrom the model.

type string

The type of the refusal. Always `refusal`.

sequence_number integer

The sequence number of this event.

type string

The type of the event. Always `response.content_part.added`.

OBJECT response.content_part.added

```
{
  "type": "response.content_part.added",
  "item_id": "msg_123",
  "output_index": 0,
  "content_index": 0,
  "part": {
    "type": "output_text",
    "text": "",
    "annotations": []
```

```
  },
  "sequence_number": 1
}
```

response.content_part.done

Emitted when a content part is done.

content_index integer
The index of the content part that is done.

item_id string
The ID of the output item that the content part was added to.

output_index integer
The index of the output item that the content part was added to.

part object
The content part that is done.

Hide possible types

Output text object
A text output from the model.

Hide properties

annotations array
The annotations of the text output.

Hide possible types

File citation object
A citation to a file.

Hide properties

file_id string
The ID of the file.

filename string
The filename of the file cited.

index integer

The index of the file in the list of files.

type string
The type of the file citation. Always `file_citation` .

URL citation object
A citation for a web resource used to generate a model response.

Hide properties

end_index integer
The index of the last character of the URL citation in the message.

start_index integer
The index of the first character of the URL citation in the message.

title string
The title of the web resource.

type string
The type of the URL citation. Always `url_citation` .

url string
The URL of the web resource.

Container file citation object
A citation for a container file used to generate a model response.

Hide properties

container_id string
The ID of the container file.

end_index integer
The index of the last character of the container file citation in the message.

file_id string
The ID of the file.

filename string
The filename of the container file cited.

start_index integer
The index of the first character of the container file citation in the message.

type string
The type of the container file citation. Always `container_file_citation`.

File path object
A path to a file.

Hide properties

file_id string
The ID of the file.

index integer
The index of the file in the list of files.

type string
The type of the file path. Always `file_path`.

text string
The text output from the model.

type string
The type of the output text. Always `output_text`.

logprobs array

Hide properties

bytes array

logprob number

token string

top_logprobs array

Hide properties

bytes array

logprob number

token string

Refusal object

A refusal from the model.

Hide properties

refusal string

The refusal explanationfrom the model.

type string

The type of the refusal. Always `refusal` .

sequence_number integer

The sequence number of this event.

type string

The type of the event. Always `response.content_part.done` .

OBJECT response.content_part.done

```
{
  "type": "response.content_part.done",
  "item_id": "msg_123",
  "output_index": 0,
  "content_index": 0,
  "sequence_number": 1,
  "part": {
    "type": "output_text",
    "text": "In a shimmering forest under a sky full of stars, a lonely unicorn named Lila
    "annotations": []
  }
}
```

response.output_text.delta

Emitted when there is an additional text delta.

content_index integer
The index of the content part that the text delta was added to.

delta string
The text delta that was added.

item_id string
The ID of the output item that the text delta was added to.

output_index integer
The index of the output item that the text delta was added to.

sequence_number integer
The sequence number for this event.

type string
The type of the event. Always `response.output_text.delta`.

OBJECT response.output_text.delta

```
{
  "type": "response.output_text.delta",
  "item_id": "msg_123",
  "output_index": 0,
  "content_index": 0,
  "delta": "In",
  "sequence_number": 1
}
```

response.output_text.done

Emitted when text content is finalized.

content_index integer
The index of the content part that the text content is finalized.

item_id string

The ID of the output item that the text content is finalized.

output_index integer
The index of the output item that the text content is finalized.

sequence_number integer
The sequence number for this event.

text string
The text content that is finalized.

type string
The type of the event. Always `response.output_text.done`.

OBJECT response.output_text.done

```
{
  "type": "response.output_text.done",
  "item_id": "msg_123",
  "output_index": 0,
  "content_index": 0,
  "text": "In a shimmering forest under a sky full of stars, a lonely unicorn named Lila di
  "sequence_number": 1
}
```

response.refusal.delta

Emitted when there is a partial refusal text.

content_index integer
The index of the content part that the refusal text is added to.

delta string
The refusal text that is added.

item_id string
The ID of the output item that the refusal text is added to.

output_index integer
The index of the output item that the refusal text is added to.

sequence_number integer
The sequence number of this event.

type string
The type of the event. Always `response.refusal.delta`.

OBJECT response.refusal.delta

```
{
  "type": "response.refusal.delta",
  "item_id": "msg_123",
  "output_index": 0,
  "content_index": 0,
  "delta": "refusal text so far",
  "sequence_number": 1
}
```

response.refusal.done

Emitted when refusal text is finalized.

content_index integer
The index of the content part that the refusal text is finalized.

item_id string
The ID of the output item that the refusal text is finalized.

output_index integer
The index of the output item that the refusal text is finalized.

refusal string
The refusal text that is finalized.

sequence_number integer
The sequence number of this event.

type string
The type of the event. Always `response.refusal.done`.

OBJECT response.refusal.done

```
{
  "type": "response.refusal.done",
  "item_id": "item-abc",
  "output_index": 1,
  "content_index": 2,
  "refusal": "final refusal text",
  "sequence_number": 1
}
```

response.function_call_arguments.delta

Emitted when there is a partial function-call arguments delta.

delta string
The function-call arguments delta that is added.

item_id string
The ID of the output item that the function-call arguments delta is added to.

output_index integer
The index of the output item that the function-call arguments delta is added to.

sequence_number integer
The sequence number of this event.

type string
The type of the event. Always `response.function_call_arguments.delta`.

OBJECT response.function_call_arguments.delta

```
{
  "type": "response.function_call_arguments.delta",
```

```
  "item_id": "item-abc",
  "output_index": 0,
  "delta": "{ \"arg\":"
  "sequence_number": 1
}
```

response.function_call_arguments.done

Emitted when function-call arguments are finalized.

arguments string
The function-call arguments.

item_id string
The ID of the item.

output_index integer
The index of the output item.

sequence_number integer
The sequence number of this event.

type string

OBJECT response.function_call_arguments.done

```
{
  "type": "response.function_call_arguments.done",
  "item_id": "item-abc",
  "output_index": 1,
  "arguments": "{ \"arg\": 123 }",
  "sequence_number": 1
}
```

response.file_search_call.in_progress

Emitted when a file search call is initiated.

item_id string
The ID of the output item that the file search call is initiated.

output_index integer
The index of the output item that the file search call is initiated.

sequence_number integer
The sequence number of this event.

type string
The type of the event. Always `response.file_search_call.in_progress`.

OBJECT response.file_search_call.in_progress

```
{
  "type": "response.file_search_call.in_progress",
  "output_index": 0,
  "item_id": "fs_123",
  "sequence_number": 1
}
```

response.file_search_call.searching

Emitted when a file search is currently searching.

item_id string
The ID of the output item that the file search call is initiated.

output_index integer
The index of the output item that the file search call is searching.

sequence_number integer
The sequence number of this event.

type string
The type of the event. Always `response.file_search_call.searching`.

OBJECT response.file_search_call.searching

```
{
  "type": "response.file_search_call.searching",
  "output_index": 0,
  "item_id": "fs_123",
  "sequence_number": 1
}
```

response.file_search_call.completed

Emitted when a file search call is completed (results found).

item_id string
The ID of the output item that the file search call is initiated.

output_index integer
The index of the output item that the file search call is initiated.

sequence_number integer
The sequence number of this event.

type string
The type of the event. Always `response.file_search_call.completed`.

OBJECT response.file_search_call.completed

```
{
  "type": "response.file_search_call.completed",
  "output_index": 0,
  "item_id": "fs_123",
  "sequence_number": 1
}
```

response.web_search_call.in_progress

Emitted when a web search call is initiated.

item_id string
Unique ID for the output item associated with the web search call.

output_index integer
The index of the output item that the web search call is associated with.

sequence_number integer
The sequence number of the web search call being processed.

type string
The type of the event. Always `response.web_search_call.in_progress`.

OBJECT response.web_search_call.in_progress

```
{
  "type": "response.web_search_call.in_progress",
  "output_index": 0,
  "item_id": "ws_123",
  "sequence_number": 0
}
```

response.web_search_call.searching

Emitted when a web search call is executing.

item_id string
Unique ID for the output item associated with the web search call.

output_index integer
The index of the output item that the web search call is associated with.

sequence_number integer
The sequence number of the web search call being processed.

type string
The type of the event. Always `response.web_search_call.searching`.

OBJECT response.web_search_call.searching

```
{
  "type": "response.web_search_call.searching",
  "output_index": 0,
  "item_id": "ws_123",
  "sequence_number": 0
}
```

response.web_search_call.completed

Emitted when a web search call is completed.

item_id string
Unique ID for the output item associated with the web search call.

output_index integer
The index of the output item that the web search call is associated with.

sequence_number integer
The sequence number of the web search call being processed.

type string
The type of the event. Always `response.web_search_call.completed`.

OBJECT response.web_search_call.completed

```
{
  "type": "response.web_search_call.completed",
  "output_index": 0,
  "item_id": "ws_123",
  "sequence_number": 0
}
```

response.reasoning_summary_part.added

Emitted when a new reasoning summary part is added.

item_id string
The ID of the item this summary part is associated with.

output_index integer
The index of the output item this summary part is associated with.

part object
The summary part that was added.

Hide properties

text string
The text of the summary part.

type string
The type of the summary part. Always `summary_text`.

sequence_number integer
The sequence number of this event.

summary_index integer
The index of the summary part within the reasoning summary.

type string
The type of the event. Always `response.reasoning_summary_part.added`.

OBJECT response.reasoning_summary_part.added

```
{
  "type": "response.reasoning_summary_part.added",
  "item_id": "rs_6806bfca0b2481918a5748308061a2600d3ce51bdffd5476",
  "output_index": 0,
  "summary_index": 0,
  "part": {
    "type": "summary_text",
    "text": ""
  },
  "sequence_number": 1
}
```

response.reasoning_summary_part.done

Emitted when a reasoning summary part is completed.

item_id string
The ID of the item this summary part is associated with.

output_index integer
The index of the output item this summary part is associated with.

part object
The completed summary part.

Hide properties

text string
The text of the summary part.

type string
The type of the summary part. Always `summary_text`.

sequence_number integer
The sequence number of this event.

summary_index integer
The index of the summary part within the reasoning summary.

type string
The type of the event. Always `response.reasoning_summary_part.done`.

OBJECT response.reasoning_summary_part.done

```
{
  "type": "response.reasoning_summary_part.done",
  "item_id": "rs_6806bfca0b2481918a5748308061a2600d3ce51bdffd5476",
  "output_index": 0,
  "summary_index": 0,
  "part": {
    "type": "summary_text",
    "text": "**Responding to a greeting**\n\nThe user just said, \"Hello!\" So, it seems I
  },
  "sequence_number": 1
```

```
}
```

response.reasoning_summary_text.delta

Emitted when a delta is added to a reasoning summary text.

delta string
The text delta that was added to the summary.

item_id string
The ID of the item this summary text delta is associated with.

output_index integer
The index of the output item this summary text delta is associated with.

sequence_number integer
The sequence number of this event.

summary_index integer
The index of the summary part within the reasoning summary.

type string
The type of the event. Always `response.reasoning_summary_text.delta`.

OBJECT response.reasoning_summary_text.delta

```
{
  "type": "response.reasoning_summary_text.delta",
  "item_id": "rs_6806bfca0b2481918a5748308061a2600d3ce51bdffd5476",
  "output_index": 0,
  "summary_index": 0,
  "delta": "**Responding to a greeting**\n\nThe user just said, \"Hello!\" So, it seems I r
  "sequence_number": 1
}
```

response.reasoning_summary_text.done

Emitted when a reasoning summary text is completed.

item_id string
The ID of the item this summary text is associated with.

output_index integer
The index of the output item this summary text is associated with.

sequence_number integer
The sequence number of this event.

summary_index integer
The index of the summary part within the reasoning summary.

text string
The full text of the completed reasoning summary.

type string
The type of the event. Always `response.reasoning_summary_text.done`.

OBJECT response.reasoning_summary_text.done

```
{
  "type": "response.reasoning_summary_text.done",
  "item_id": "rs_6806bfca0b2481918a5748308061a2600d3ce51bdffd5476",
  "output_index": 0,
  "summary_index": 0,
  "text": "**Responding to a greeting**\n\nThe user just said, \"Hello!\" So, it seems I ne
  "sequence_number": 1
}
```

response.image_generation_call.completed

Emitted when an image generation tool call has completed and the final image is available.

item_id string
The unique identifier of the image generation item being processed.

output_index integer
The index of the output item in the response's output array.

sequence_number integer
The sequence number of this event.

type string
The type of the event. Always 'response.image_generation_call.completed'.

OBJECT response.image_generation_call.completed

```
{
  "type": "response.image_generation_call.completed",
  "output_index": 0,
  "item_id": "item-123",
  "sequence_number": 1
}
```

response.image_generation_call.generating

Emitted when an image generation tool call is actively generating an image (intermediate state).

item_id string
The unique identifier of the image generation item being processed.

output_index integer
The index of the output item in the response's output array.

sequence_number integer
The sequence number of the image generation item being processed.

type string
The type of the event. Always 'response.image_generation_call.generating'.

OBJECT response.image_generation_call.generating

```
{
  "type": "response.image_generation_call.generating",
  "output_index": 0,
  "item_id": "item-123",
  "sequence_number": 0
}
```

response.image_generation_call.in_progress

Emitted when an image generation tool call is in progress.

item_id string
The unique identifier of the image generation item being processed.

output_index integer
The index of the output item in the response's output array.

sequence_number integer
The sequence number of the image generation item being processed.

type string
The type of the event. Always 'response.image_generation_call.in_progress'.

OBJECT response.image_generation_call.in_progress

```
{
  "type": "response.image_generation_call.in_progress",
  "output_index": 0,
  "item_id": "item-123",
  "sequence_number": 0
}
```

response.image_generation_call.partial_image

Emitted when a partial image is available during image generation streaming.

item_id string
The unique identifier of the image generation item being processed.

output_index integer
The index of the output item in the response's output array.

partial_image_b64 string
Base64-encoded partial image data, suitable for rendering as an image.

partial_image_index integer
0-based index for the partial image (backend is 1-based, but this is 0-based for the user).

sequence_number integer
The sequence number of the image generation item being processed.

type string
The type of the event. Always 'response.image_generation_call.partial_image'.

OBJECT response.image_generation_call.partial_image

```
{
  "type": "response.image_generation_call.partial_image",
  "output_index": 0,
  "item_id": "item-123",
  "sequence_number": 0,
  "partial_image_index": 0,
  "partial_image_b64": "..."
}
```

response.mcp_call_arguments.delta

Emitted when there is a delta (partial update) to the arguments of an MCP tool call.

delta object
The partial update to the arguments for the MCP tool call.

item_id string

The unique identifier of the MCP tool call item being processed.

output_index integer
The index of the output item in the response's output array.

sequence_number integer
The sequence number of this event.

type string
The type of the event. Always 'response.mcp_call_arguments.delta'.

OBJECT response.mcp_call_arguments.delta

```
{
  "type": "response.mcp_call_arguments.delta",
  "output_index": 0,
  "item_id": "item-abc",
  "delta": {
    "arg1": "new_value1",
    "arg2": "new_value2"
  },
  "sequence_number": 1
}
```

response.mcp_call_arguments.done

Emitted when the arguments for an MCP tool call are finalized.

arguments object
The finalized arguments for the MCP tool call.

item_id string
The unique identifier of the MCP tool call item being processed.

output_index integer
The index of the output item in the response's output array.

sequence_number integer
The sequence number of this event.

type string
The type of the event. Always 'response.mcp_call_arguments.done'.

OBJECT response.mcp_call_arguments.done

```
{
  "type": "response.mcp_call_arguments.done",
  "output_index": 0,
  "item_id": "item-abc",
  "arguments": {
    "arg1": "value1",
    "arg2": "value2"
  },
  "sequence_number": 1
}
```

response.mcp_call.completed

Emitted when an MCP tool call has completed successfully.

sequence_number integer
The sequence number of this event.

type string
The type of the event. Always 'response.mcp_call.completed'.

OBJECT response.mcp_call.completed

```
{
  "type": "response.mcp_call.completed",
  "sequence_number": 1
}
```

response.mcp_call.failed

Emitted when an MCP tool call has failed.

sequence_number integer
The sequence number of this event.

type string
The type of the event. Always 'response.mcp_call.failed'.

OBJECT response.mcp_call.failed

```
{
  "type": "response.mcp_call.failed",
  "sequence_number": 1
}
```

response.mcp_call.in_progress

Emitted when an MCP tool call is in progress.

item_id string
The unique identifier of the MCP tool call item being processed.

output_index integer
The index of the output item in the response's output array.

sequence_number integer
The sequence number of this event.

type string
The type of the event. Always 'response.mcp_call.in_progress'.

OBJECT response.mcp_call.in_progress

```
{
  "type": "response.mcp_call.in_progress",
  "output_index": 0,
  "item_id": "item-abc",
  "sequence_number": 1
}
```

response.mcp_list_tools.completed

Emitted when the list of available MCP tools has been successfully retrieved.

sequence_number integer
The sequence number of this event.

type string
The type of the event. Always 'response.mcp_list_tools.completed'.

OBJECT response.mcp_list_tools.completed

```
{
  "type": "response.mcp_list_tools.completed",
  "sequence_number": 1
}
```

response.mcp_list_tools.failed

Emitted when the attempt to list available MCP tools has failed.

sequence_number integer
The sequence number of this event.

type string
The type of the event. Always 'response.mcp_list_tools.failed'.

OBJECT response.mcp_list_tools.failed

```
{
  "type": "response.mcp_list_tools.failed",
  "sequence_number": 1
}
```

response.mcp_list_tools.in_progress

Emitted when the system is in the process of retrieving the list of available MCP tools.

sequence_number integer
The sequence number of this event.

type string
The type of the event. Always 'response.mcp_list_tools.in_progress'.

OBJECT response.mcp_list_tools.in_progress

```
{
  "type": "response.mcp_list_tools.in_progress",
  "sequence_number": 1
}
```

response.code_interpreter_call.in_progress

Emitted when a code interpreter call is in progress.

item_id string
The unique identifier of the code interpreter tool call item.

output_index integer
The index of the output item in the response for which the code interpreter call is in progress.

sequence_number integer
The sequence number of this event, used to order streaming events.

type string
The type of the event. Always `response.code_interpreter_call.in_progress`.

OBJECT response.code_interpreter_call.in_progress

```
{
```

```
  "type": "response.code_interpreter_call.in_progress",
  "output_index": 0,
  "item_id": "ci_12345",
  "sequence_number": 1
}
```

response.code_interpreter_call.interpreting

Emitted when the code interpreter is actively interpreting the code snippet.

item_id string
The unique identifier of the code interpreter tool call item.

output_index integer
The index of the output item in the response for which the code interpreter is interpreting code.

sequence_number integer
The sequence number of this event, used to order streaming events.

type string
The type of the event. Always `response.code_interpreter_call.interpreting` .

OBJECT response.code_interpreter_call.interpreting

```
{
  "type": "response.code_interpreter_call.interpreting",
  "output_index": 4,
  "item_id": "ci_12345",
  "sequence_number": 1
}
```

response.code_interpreter_call.completed

Emitted when the code interpreter call is completed.

item_id string
The unique identifier of the code interpreter tool call item.

output_index integer
The index of the output item in the response for which the code interpreter call is completed.

sequence_number integer
The sequence number of this event, used to order streaming events.

type string
The type of the event. Always `response.code_interpreter_call.completed`.

OBJECT response.code_interpreter_call.completed

```
{
  "type": "response.code_interpreter_call.completed",
  "output_index": 5,
  "item_id": "ci_12345",
  "sequence_number": 1
}
```

response.code_interpreter_call_code.delta

Emitted when a partial code snippet is streamed by the code interpreter.

delta string
The partial code snippet being streamed by the code interpreter.

item_id string
The unique identifier of the code interpreter tool call item.

output_index integer
The index of the output item in the response for which the code is being streamed.

sequence_number integer
The sequence number of this event, used to order streaming events.

type string
The type of the event. Always `response.code_interpreter_call_code.delta`.

OBJECT response.code_interpreter_call_code.delta

```
{
  "type": "response.code_interpreter_call_code.delta",
  "output_index": 0,
  "item_id": "ci_12345",
  "delta": "print('Hello, world')",
  "sequence_number": 1
}
```

response.code_interpreter_call_code.done

Emitted when the code snippet is finalized by the code interpreter.

code string
The final code snippet output by the code interpreter.

item_id string
The unique identifier of the code interpreter tool call item.

output_index integer
The index of the output item in the response for which the code is finalized.

sequence_number integer
The sequence number of this event, used to order streaming events.

type string
The type of the event. Always `response.code_interpreter_call_code.done`.

OBJECT response.code_interpreter_call_code.done

```
{
  "type": "response.code_interpreter_call_code.done",
  "output_index": 3,
  "item_id": "ci_12345",
  "code": "print('done')",
  "sequence_number": 1
}
```

response.output_text.annotation.added

Emitted when an annotation is added to output text content.

annotation object
The annotation object being added. (See annotation schema for details.)

annotation_index integer
The index of the annotation within the content part.

content_index integer
The index of the content part within the output item.

item_id string
The unique identifier of the item to which the annotation is being added.

output_index integer
The index of the output item in the response's output array.

sequence_number integer
The sequence number of this event.

type string
The type of the event. Always 'response.output_text.annotation.added'.

OBJECT response.output_text.annotation.added

```
{
  "type": "response.output_text.annotation.added",
  "item_id": "item-abc",
  "output_index": 0,
  "content_index": 0,
  "annotation_index": 0,
  "annotation": {
    "type": "text_annotation",
    "text": "This is a test annotation",
    "start": 0,

```

```
    "end": 10
  },
  "sequence_number": 1
}
```

response.queued

Emitted when a response is queued and waiting to be processed.

response object
The full response object that is queued.

Hide properties

created_at number
Unix timestamp (in seconds) of when this Response was created.

error object or null
An error object returned when the model fails to generate a Response.

Hide properties

code string
The error code for the response.

message string
A human-readable description of the error.

id string
Unique identifier for this Response.

incomplete_details object or null
Details about why the response is incomplete.

Hide properties

reason string
The reason why the response is incomplete.

instructions string or array
A system (or developer) message inserted into the model's context.

When using along with `previous_response_id`, the instructions from a previous response will not be

carried over to the next response. This makes it simple to swap out system (or developer) messages in new responses.

Hide possible types

string
A text input to the model, equivalent to a text input with the `developer` role.

Input item list array
A list of one or many input items to the model, containing different content types.

Hide possible types

Input message object
A message input to the model with a role indicating instruction following hierarchy. Instructions given with the `developer` or `system` role take precedence over instructions given with the `user` role. Messages with the `assistant` role are presumed to have been generated by the model in previous interactions.

Hide properties

content string or array
Text, image, or audio input to the model, used to generate a response. Can also contain previous assistant responses.

Hide possible types

Text input string
A text input to the model.

Input item content list array
A list of one or many input items to the model, containing different content types.

Hide possible types

Input text object
A text input to the model.

Hide properties

text string
The text input to the model.

type string
The type of the input item. Always `input_text`.

Input image object
An image input to the model. Learn about image inputs.

Hide properties

detail string
The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

type string
The type of the input item. Always `input_image`.

file_id string
The ID of the file to be sent to the model.

image_url string
The URL of the image to be sent to the model. A fully qualified URL or base64 encoded image in a data URL.

Input file object
A file input to the model.

Hide properties

type string
The type of the input item. Always `input_file`.

file_data string
The content of the file to be sent to the model.

file_id string
The ID of the file to be sent to the model.

file_url string
The URL of the file to be sent to the model.

filename string
The name of the file to be sent to the model.

role string
The role of the message input. One of `user`, `assistant`, `system`, or `developer`.

type string
The type of the message input. Always `message`.

Item object

An item representing part of the context for the response to be generated by the model. Can contain text, images, and audio inputs, as well as previous assistant responses and tool call outputs.

Hide possible types

Input message object

A message input to the model with a role indicating instruction following hierarchy. Instructions given with the `developer` or `system` role take precedence over instructions given with the `user` role.

Hide properties

content array

A list of one or many input items to the model, containing different content types.

Hide possible types

Input text object

A text input to the model.

Hide properties

text string

The text input to the model.

type string

The type of the input item. Always `input_text`.

Input image object

An image input to the model. Learn about image inputs.

Hide properties

detail string

The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

type string

The type of the input item. Always `input_image`.

file_id string

The ID of the file to be sent to the model.

image_url string

The URL of the image to be sent to the model. A fully qualified URL or base64

encoded image in a data URL.

Input file object

A file input to the model.

Hide properties

type string

The type of the input item. Always `input_file`.

file_data string

The content of the file to be sent to the model.

file_id string

The ID of the file to be sent to the model.

file_url string

The URL of the file to be sent to the model.

filename string

The name of the file to be sent to the model.

role string

The role of the message input. One of `user`, `system`, or `developer`.

status string

The status of item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

type string

The type of the message input. Always set to `message`.

Output message object

An output message from the model.

Hide properties

content array

The content of the output message.

Hide possible types

Output text object

A text output from the model.

Hide properties

annotations array

The annotations of the text output.

Hide possible types

File citation object

A citation to a file.

Hide properties

file_id string

The ID of the file.

filename string

The filename of the file cited.

index integer

The index of the file in the list of files.

type string

The type of the file citation. Always `file_citation`.

URL citation object

A citation for a web resource used to generate a model response.

Hide properties

end_index integer

The index of the last character of the URL citation in the message.

start_index integer

The index of the first character of the URL citation in the message.

title string

The title of the web resource.

type string

The type of the URL citation. Always `url_citation`.

url string

The URL of the web resource.

Container file citation object

A citation for a container file used to generate a model response.

Hide properties

container_id string
The ID of the container file.

end_index integer
The index of the last character of the container file citation in the message.

file_id string
The ID of the file.

filename string
The filename of the container file cited.

start_index integer
The index of the first character of the container file citation in the message.

type string
The type of the container file citation. Always `container_file_citation`.

File path object
A path to a file.

Hide properties

file_id string
The ID of the file.

index integer
The index of the file in the list of files.

type string
The type of the file path. Always `file_path`.

text string
The text output from the model.

type string

The type of the output text. Always `output_text`.

logprobs array

Hide properties

bytes array

logprob number

token string

top_logprobs array

Hide properties

bytes array

logprob number

token string

Refusal object

A refusal from the model.

Hide properties

refusal string

The refusal explanationfrom the model.

type string

The type of the refusal. Always `refusal`.

id string

The unique ID of the output message.

role string

The role of the output message. Always `assistant`.

status string

The status of the message input. One of `in_progress`, `completed`, or `incomplete`. Populated when input items are returned via API.

type string
The type of the output message. Always `message`.

File search tool call object
The results of a file search tool call. See the file search guide for more information.

Hide properties

id string
The unique ID of the file search tool call.

queries array
The queries used to search for files.

status string
The status of the file search tool call. One of `in_progress`, `searching`, `incomplete` or `failed`,

type string
The type of the file search tool call. Always `file_search_call`.

results array or null
The results of the file search tool call.

Hide properties

attributes map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard. Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters, booleans, or numbers.

file_id string
The unique ID of the file.

filename string
The name of the file.

score number
The relevance score of the file - a value between 0 and 1.

text string
The text that was retrieved from the file.

Computer tool call object
A tool call to a computer use tool. See the computer use guide for more information.

Hide properties

action object

Hide possible types

Click object
A click action.

Hide properties

button string
Indicates which mouse button was pressed during the click. One of `left`, `right`, `wheel`, `back`, or `forward`.

type string
Specifies the event type. For a click action, this property is always set to `click`.

x integer
The x-coordinate where the click occurred.

y integer
The y-coordinate where the click occurred.

DoubleClick object
A double click action.

Hide properties

type string
Specifies the event type. For a double click action, this property is always set to `double_click`.

x integer
The x-coordinate where the double click occurred.

y integer
The y-coordinate where the double click occurred.

Drag object

A drag action.

Hide properties

path array

An array of coordinates representing the path of the drag action. Coordinates will appear as an array of objects, eg

```
[
  { x: 100, y: 200 },
  { x: 200, y: 300 }
]
```

Hide properties

x integer

The x-coordinate.

y integer

The y-coordinate.

type string

Specifies the event type. For a drag action, this property is always set to `drag`.

KeyPress object

A collection of keypresses the model would like to perform.

Hide properties

keys array

The combination of keys the model is requesting to be pressed. This is an array of strings, each representing a key.

type string

Specifies the event type. For a keypress action, this property is always set to `keypress`.

Move object

A mouse move action.

Hide properties

type string
Specifies the event type. For a move action, this property is always set to `move`.

x integer
The x-coordinate to move to.

y integer
The y-coordinate to move to.

Screenshot object
A screenshot action.

Hide properties

type string
Specifies the event type. For a screenshot action, this property is always set to `screenshot`.

Scroll object
A scroll action.

Hide properties

scroll_x integer
The horizontal scroll distance.

scroll_y integer
The vertical scroll distance.

type string
Specifies the event type. For a scroll action, this property is always set to `scroll`.

x integer
The x-coordinate where the scroll occurred.

y integer
The y-coordinate where the scroll occurred.

Type object
An action to type in text.

Hide properties

text string
The text to type.

type string
Specifies the event type. For a type action, this property is always set to `type`.

Wait object
A wait action.

Hide properties

type string
Specifies the event type. For a wait action, this property is always set to `wait`.

call_id string
An identifier used when responding to the tool call with output.

id string
The unique ID of the computer call.

pending_safety_checks array
The pending safety checks for the computer call.

Hide properties

code string
The type of the pending safety check.

id string
The ID of the pending safety check.

message string
Details about the pending safety check.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

type string
The type of the computer call. Always `computer_call`.

Computer tool call output object

The output of a computer tool call.

Hide properties

call_id string

The ID of the computer tool call that produced the output.

output object

A computer screenshot image used with the computer use tool.

Hide properties

type string

Specifies the event type. For a computer screenshot, this property is always set to `computer_screenshot`.

file_id string

The identifier of an uploaded file that contains the screenshot.

image_url string

The URL of the screenshot image.

type string

The type of the computer tool call output. Always `computer_call_output`.

acknowledged_safety_checks array

The safety checks reported by the API that have been acknowledged by the developer.

Hide properties

id string

The ID of the pending safety check.

code string

The type of the pending safety check.

message string

Details about the pending safety check.

id string

The ID of the computer tool call output.

status string

The status of the message input. One of `in_progress`, `completed`, or `incomplete`. Populated when input items are returned via API.

Web search tool call object

The results of a web search tool call. See the web search guide for more information.

Hide properties

action object

An object describing the specific action taken in this web search call. Includes details on how the model used the web (search, open_page, find).

Hide possible types

Search action object

Action type "search" - Performs a web search query.

Hide properties

query string

The search query.

type string

The action type.

Open page action object

Action type "open_page" - Opens a specific URL from search results.

Hide properties

type string

The action type.

url string

The URL opened by the model.

Find action object

Action type "find": Searches for a pattern within a loaded page.

Hide properties

pattern string

The pattern or text to search for within the page.

type string

The action type.

url string
The URL of the page searched for the pattern.

id string
The unique ID of the web search tool call.

status string
The status of the web search tool call.

type string
The type of the web search tool call. Always `web_search_call`.

Function tool call object
A tool call to run a function. See the function calling guide for more information.

Hide properties

arguments string
A JSON string of the arguments to pass to the function.

call_id string
The unique ID of the function tool call generated by the model.

name string
The name of the function to run.

type string
The type of the function tool call. Always `function_call`.

id string
The unique ID of the function tool call.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Function tool call output object
The output of a function tool call.

Hide properties

call_id string

The unique ID of the function tool call generated by the model.

output string
A JSON string of the output of the function tool call.

type string
The type of the function tool call output. Always `function_call_output`.

id string
The unique ID of the function tool call output. Populated when this item is returned via API.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Reasoning object
A description of the chain of thought used by a reasoning model while generating a response. Be sure to include these items in your `input` to the Responses API for subsequent turns of a conversation if you are manually managing context.

Hide properties

id string
The unique identifier of the reasoning content.

summary array
Reasoning text contents.

Hide properties

text string
A short summary of the reasoning used by the model when generating the response.

type string
The type of the object. Always `summary_text`.

type string
The type of the object. Always `reasoning`.

encrypted_content string or null
The encrypted content of the reasoning item - populated when a response is generated with `reasoning.encrypted_content` in the `include` parameter.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Image generation call object
An image generation request made by the model.

Hide properties

id string
The unique ID of the image generation call.

result string or null
The generated image encoded in base64.

status string
The status of the image generation call.

type string
The type of the image generation call. Always `image_generation_call`.

Code interpreter tool call object
A tool call to run code.

Hide properties

code string or null
The code to run, or null if not available.

container_id string
The ID of the container used to run the code.

id string
The unique ID of the code interpreter tool call.

outputs array or null
The outputs generated by the code interpreter, such as logs or images. Can be null if no outputs are available.

Hide possible types

Code interpreter output logs object
The logs output from the code interpreter.

Hide properties

logs string
The logs output from the code interpreter.

type string
The type of the output. Always 'logs'.

Code interpreter output image object
The image output from the code interpreter.

Hide properties

type string
The type of the output. Always 'image'.

url string
The URL of the image output from the code interpreter.

status string
The status of the code interpreter tool call.

type string
The type of the code interpreter tool call. Always `code_interpreter_call`.

Local shell call object
A tool call to run a command on the local shell.

Hide properties

action object
Execute a shell command on the server.

Hide properties

command array
The command to run.

env object
Environment variables to set for the command.

type string
The type of the local shell action. Always `exec`.

timeout_ms integer or null

Optional timeout in milliseconds for the command.

user string or null
Optional user to run the command as.

working_directory string or null
Optional working directory to run the command in.

call_id string
The unique ID of the local shell tool call generated by the model.

id string
The unique ID of the local shell call.

status string
The status of the local shell call.

type string
The type of the local shell call. Always `local_shell_call`.

Local shell call output object
The output of a local shell tool call.

Hide properties

id string
The unique ID of the local shell tool call generated by the model.

output string
A JSON string of the output of the local shell tool call.

type string
The type of the local shell tool call output. Always `local_shell_call_output`.

status string or null
The status of the item. One of `in_progress`, `completed`, or `incomplete`.

MCP list tools object
A list of tools available on an MCP server.

Hide properties

id string

The unique ID of the list.

server_label string
The label of the MCP server.

tools array
The tools available on the server.

Hide properties

input_schema object
The JSON schema describing the tool's input.

name string
The name of the tool.

annotations object or null
Additional annotations about the tool.

description string or null
The description of the tool.

type string
The type of the item. Always `mcp_list_tools`.

error string or null
Error message if the server could not list tools.

MCP approval request object
A request for human approval of a tool invocation.

Hide properties

arguments string
A JSON string of arguments for the tool.

id string
The unique ID of the approval request.

name string
The name of the tool to run.

server_label string
The label of the MCP server making the request.

type string
The type of the item. Always `mcp_approval_request`.

MCP approval response object
A response to an MCP approval request.

Hide properties

approval_request_id string
The ID of the approval request being answered.

approve boolean
Whether the request was approved.

type string
The type of the item. Always `mcp_approval_response`.

id string or null
The unique ID of the approval response

reason string or null
Optional reason for the decision.

MCP tool call object
An invocation of a tool on an MCP server.

Hide properties

arguments string
A JSON string of the arguments passed to the tool.

id string
The unique ID of the tool call.

name string
The name of the tool that was run.

server_label string
The label of the MCP server running the tool.

type string
The type of the item. Always `mcp_call`.

error string or null
The error from the tool call, if any.

output string or null
The output from the tool call.

Item reference object
An internal identifier for an item to reference.

Hide properties

id string
The ID of the item to reference.

type string
The type of item to reference. Always `item_reference`.

metadata map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

model string
Model ID used to generate the response, like `gpt-4o` or `o3`. OpenAI offers a wide range of models with different capabilities, performance characteristics, and price points. Refer to the model guide to browse and compare available models.

object string
The object type of this resource - always set to `response`.

output array
An array of content items generated by the model.

The length and order of items in the `output` array is dependent on the model's response.

Rather than accessing the first item in the `output` array and assuming it's an `assistant` message with the content generated by the model, you might consider using the `output_text`

property where supported in SDKs.

Hide possible types

Output message object

An output message from the model.

Hide properties

content array

The content of the output message.

Hide possible types

Output text object

A text output from the model.

Hide properties

annotations array

The annotations of the text output.

Hide possible types

File citation object

A citation to a file.

Hide properties

file_id string

The ID of the file.

filename string

The filename of the file cited.

index integer

The index of the file in the list of files.

type string

The type of the file citation. Always `file_citation`.

URL citation object

A citation for a web resource used to generate a model response.

Hide properties

end_index integer

The index of the last character of the URL citation in the message.

start_index integer
The index of the first character of the URL citation in the message.

title string
The title of the web resource.

type string
The type of the URL citation. Always `url_citation`.

url string
The URL of the web resource.

Container file citation object
A citation for a container file used to generate a model response.

Hide properties

container_id string
The ID of the container file.

end_index integer
The index of the last character of the container file citation in the message.

file_id string
The ID of the file.

filename string
The filename of the container file cited.

start_index integer
The index of the first character of the container file citation in the message.

type string
The type of the container file citation. Always `container_file_citation`.

File path object
A path to a file.

Hide properties

file_id string
The ID of the file.

index integer
The index of the file in the list of files.

type string
The type of the file path. Always `file_path`.

text string
The text output from the model.

type string
The type of the output text. Always `output_text`.

logprobs array
Hide properties

bytes array

logprob number

token string

top_logprobs array
Hide properties

bytes array

logprob number

token string

Refusal object
A refusal from the model.
Hide properties

refusal string
The refusal explanationfrom the model.

type string
The type of the refusal. Always `refusal`.

id string
The unique ID of the output message.

role string
The role of the output message. Always `assistant`.

status string
The status of the message input. One of `in_progress`, `completed`, or `incomplete`. Populated when input items are returned via API.

type string
The type of the output message. Always `message`.

File search tool call object
The results of a file search tool call. See the file search guide for more information.

Hide properties

id string
The unique ID of the file search tool call.

queries array
The queries used to search for files.

status string
The status of the file search tool call. One of `in_progress`, `searching`, `incomplete` or `failed`,

type string
The type of the file search tool call. Always `file_search_call`.

results array or null
The results of the file search tool call.

Hide properties

attributes map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard. Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters, booleans, or numbers.

file_id string

The unique ID of the file.

filename string
The name of the file.

score number
The relevance score of the file - a value between 0 and 1.

text string
The text that was retrieved from the file.

Function tool call object
A tool call to run a function. See the function calling guide for more information.

Hide properties

arguments string
A JSON string of the arguments to pass to the function.

call_id string
The unique ID of the function tool call generated by the model.

name string
The name of the function to run.

type string
The type of the function tool call. Always `function_call`.

id string
The unique ID of the function tool call.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Web search tool call object
The results of a web search tool call. See the web search guide for more information.

Hide properties

action object
An object describing the specific action taken in this web search call. Includes details on how the model used the web (search, open_page, find).

Hide possible types

Search action object
Action type "search" - Performs a web search query.

Hide properties

query string
The search query.

type string
The action type.

Open page action object
Action type "open_page" - Opens a specific URL from search results.

Hide properties

type string
The action type.

url string
The URL opened by the model.

Find action object
Action type "find": Searches for a pattern within a loaded page.

Hide properties

pattern string
The pattern or text to search for within the page.

type string
The action type.

url string
The URL of the page searched for the pattern.

id string
The unique ID of the web search tool call.

status string
The status of the web search tool call.

type string
The type of the web search tool call. Always `web_search_call`.

Computer tool call object
A tool call to a computer use tool. See the computer use guide for more information.

Hide properties

action object

Hide possible types

Click object
A click action.

Hide properties

button string
Indicates which mouse button was pressed during the click. One of `left`, `right`, `wheel`, `back`, or `forward`.

type string
Specifies the event type. For a click action, this property is always set to `click`.

x integer
The x-coordinate where the click occurred.

y integer
The y-coordinate where the click occurred.

DoubleClick object
A double click action.

Hide properties

type string
Specifies the event type. For a double click action, this property is always set to `double_click`.

x integer
The x-coordinate where the double click occurred.

y integer
The y-coordinate where the double click occurred.

Drag object

A drag action.

Hide properties

path array

An array of coordinates representing the path of the drag action. Coordinates will appear as an array of objects, eg

```
[
  { x: 100, y: 200 },
  { x: 200, y: 300 }
]
```

Hide properties

x integer

The x-coordinate.

y integer

The y-coordinate.

type string

Specifies the event type. For a drag action, this property is always set to `drag`.

KeyPress object

A collection of keypresses the model would like to perform.

Hide properties

keys array

The combination of keys the model is requesting to be pressed. This is an array of strings, each representing a key.

type string

Specifies the event type. For a keypress action, this property is always set to `keypress`.

Move object

A mouse move action.

Hide properties

type string

Specifies the event type. For a move action, this property is always set to `move`.

x integer
The x-coordinate to move to.

y integer
The y-coordinate to move to.

Screenshot object
A screenshot action.

Hide properties

type string
Specifies the event type. For a screenshot action, this property is always set to `screenshot`.

Scroll object
A scroll action.

Hide properties

scroll_x integer
The horizontal scroll distance.

scroll_y integer
The vertical scroll distance.

type string
Specifies the event type. For a scroll action, this property is always set to `scroll`.

x integer
The x-coordinate where the scroll occurred.

y integer
The y-coordinate where the scroll occurred.

Type object
An action to type in text.

Hide properties

text string
The text to type.

type string
Specifies the event type. For a type action, this property is always set to `type`.

Wait object
A wait action.

Hide properties

type string
Specifies the event type. For a wait action, this property is always set to `wait`.

call_id string
An identifier used when responding to the tool call with output.

id string
The unique ID of the computer call.

pending_safety_checks array
The pending safety checks for the computer call.

Hide properties

code string
The type of the pending safety check.

id string
The ID of the pending safety check.

message string
Details about the pending safety check.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

type string
The type of the computer call. Always `computer_call`.

Reasoning object
A description of the chain of thought used by a reasoning model while generating a response. Be sure to include these items in your `input` to the Responses API for subsequent turns of a

conversation if you are manually managing context.

Hide properties

id string
The unique identifier of the reasoning content.

summary array
Reasoning text contents.

Hide properties

text string
A short summary of the reasoning used by the model when generating the response.

type string
The type of the object. Always `summary_text`.

type string
The type of the object. Always `reasoning`.

encrypted_content string or null
The encrypted content of the reasoning item - populated when a response is generated with `reasoning.encrypted_content` in the `include` parameter.

status string
The status of the item. One of `in_progress`, `completed`, or `incomplete`. Populated when items are returned via API.

Image generation call object
An image generation request made by the model.

Hide properties

id string
The unique ID of the image generation call.

result string or null
The generated image encoded in base64.

status string
The status of the image generation call.

type string
The type of the image generation call. Always `image_generation_call`.

Code interpreter tool call object
A tool call to run code.

Hide properties

code string or null
The code to run, or null if not available.

container_id string
The ID of the container used to run the code.

id string
The unique ID of the code interpreter tool call.

outputs array or null
The outputs generated by the code interpreter, such as logs or images. Can be null if no outputs are available.

Hide possible types

Code interpreter output logs object
The logs output from the code interpreter.

Hide properties

logs string
The logs output from the code interpreter.

type string
The type of the output. Always 'logs'.

Code interpreter output image object
The image output from the code interpreter.

Hide properties

type string
The type of the output. Always 'image'.

url string
The URL of the image output from the code interpreter.

status string
The status of the code interpreter tool call.

type string
The type of the code interpreter tool call. Always `code_interpreter_call`.

Local shell call object
A tool call to run a command on the local shell.

Hide properties

action object
Execute a shell command on the server.

Hide properties

command array
The command to run.

env object
Environment variables to set for the command.

type string
The type of the local shell action. Always `exec`.

timeout_ms integer or null
Optional timeout in milliseconds for the command.

user string or null
Optional user to run the command as.

working_directory string or null
Optional working directory to run the command in.

call_id string
The unique ID of the local shell tool call generated by the model.

id string
The unique ID of the local shell call.

status string
The status of the local shell call.

type string
The type of the local shell call. Always `local_shell_call` .

MCP tool call object
An invocation of a tool on an MCP server.

Hide properties

arguments string
A JSON string of the arguments passed to the tool.

id string
The unique ID of the tool call.

name string
The name of the tool that was run.

server_label string
The label of the MCP server running the tool.

type string
The type of the item. Always `mcp_call` .

error string or null
The error from the tool call, if any.

output string or null
The output from the tool call.

MCP list tools object
A list of tools available on an MCP server.

Hide properties

id string
The unique ID of the list.

server_label string
The label of the MCP server.

tools array
The tools available on the server.

Hide properties

input_schema object
The JSON schema describing the tool's input.

name string
The name of the tool.

annotations object or null
Additional annotations about the tool.

description string or null
The description of the tool.

type string
The type of the item. Always `mcp_list_tools`.

error string or null
Error message if the server could not list tools.

MCP approval request object
A request for human approval of a tool invocation.

Hide properties

arguments string
A JSON string of arguments for the tool.

id string
The unique ID of the approval request.

name string
The name of the tool to run.

server_label string
The label of the MCP server making the request.

type string
The type of the item. Always `mcp_approval_request`.

parallel_tool_calls boolean
Whether to allow the model to run tool calls in parallel.

temperature number or null

What sampling temperature to use, between 0 and 2. Higher values like 0.8 will make the output more random, while lower values like 0.2 will make it more focused and deterministic. We generally recommend altering this or `top_p` but not both.

tool_choice string or object

How the model should select which tool (or tools) to use when generating a response. See the `tools` parameter to see how to specify which tools the model can call.

Hide possible types

Tool choice mode string

Controls which (if any) tool is called by the model.

`none` means the model will not call any tool and instead generates a message.

`auto` means the model can pick between generating a message or calling one or more tools.

`required` means the model must call one or more tools.

Hosted tool object

Indicates that the model should use a built-in tool to generate a response. Learn more about built-in tools.

Hide properties

type string

The type of hosted tool the model should to use. Learn more about built-in tools.

Allowed values are:

`file_search`

`web_search_preview`

`computer_use_preview`

`code_interpreter`

`image_generation`

Function tool object

Use this option to force the model to call a specific function.

Hide properties

name string

The name of the function to call.

type string

For function calling, the type is always `function`.

MCP tool object

Use this option to force the model to call a specific tool on a remote MCP server.

Hide properties

server_label string

The label of the MCP server to use.

type string

For MCP tools, the type is always `mcp`.

name string or null

The name of the tool to call on the server.

tools array

An array of tools the model may call while generating a response. You can specify which tool to use by setting the `tool_choice` parameter.

The two categories of tools you can provide the model are:

Built-in tools: Tools that are provided by OpenAI that extend the model's capabilities, like web search or file search. Learn more about built-in tools.

Function calls (custom tools): Functions that are defined by you, enabling the model to call your own code. Learn more about function calling.

Hide possible types

Function object

Defines a function in your own code the model can choose to call. Learn more about function calling.

Hide properties

name string

The name of the function to call.

parameters object

A JSON schema object describing the parameters of the function.

strict boolean

Whether to enforce strict parameter validation. Default `true`.

type string
The type of the function tool. Always `function`.

description string
A description of the function. Used by the model to determine whether or not to call the function.

File search object
A tool that searches for relevant content from uploaded files. Learn more about the file search tool.

Hide properties

type string
The type of the file search tool. Always `file_search`.

vector_store_ids array
The IDs of the vector stores to search.

filters object
A filter to apply.

Hide possible types

Comparison Filter object
A filter used to compare a specified attribute key to a given value using a defined comparison operation.

Hide properties

key string
The key to compare against the value.

type string
Specifies the comparison operator: `eq`, `ne`, `gt`, `gte`, `lt`, `lte`.

- `eq`: equals
- `ne`: not equal
- `gt`: greater than
- `gte`: greater than or equal
- `lt`: less than
- `lte`: less than or equal

value string / number / boolean
The value to compare against the attribute key; supports string, number, or boolean

types.

Compound Filter object

Combine multiple filters using `and` or `or`.

Hide properties

filters array

Array of filters to combine. Items can be `ComparisonFilter` or `CompoundFilter`.

Hide possible types

Comparison Filter object

A filter used to compare a specified attribute key to a given value using a defined comparison operation.

Hide properties

key string

The key to compare against the value.

type string

Specifies the comparison operator: `eq`, `ne`, `gt`, `gte`, `lt`, `lte`.

- `eq`: equals
- `ne`: not equal
- `gt`: greater than
- `gte`: greater than or equal
- `lt`: less than
- `lte`: less than or equal

value string / number / boolean

The value to compare against the attribute key; supports string, number, or boolean types.

type string

Type of operation: `and` or `or`.

max_num_results integer

The maximum number of results to return. This number should be between 1 and 50 inclusive.

ranking_options object
Ranking options for search.

Hide properties

ranker string
The ranker to use for the file search.

score_threshold number
The score threshold for the file search, a number between 0 and 1. Numbers closer to 1 will attempt to return only the most relevant results, but may return fewer results.

Web search preview object
This tool searches the web for relevant results to use in a response. Learn more about the web search tool.

Hide properties

type string
The type of the web search tool. One of `web_search_preview` or `web_search_preview_2025_03_11`.

search_context_size string
High level guidance for the amount of context window space to use for the search. One of `low`, `medium`, or `high`. `medium` is the default.

user_location object
The user's location.

Hide properties

type string
The type of location approximation. Always `approximate`.

city string
Free text input for the city of the user, e.g. `San Francisco`.

country string
The two-letter ISO country code of the user, e.g. `US`.

region string
Free text input for the region of the user, e.g. `California`.

timezone string

The IANA timezone of the user, e.g. `America/Los_Angeles` .

Computer use preview object

A tool that controls a virtual computer. Learn more about the computer tool.

Hide properties

`display_height` integer

The height of the computer display.

`display_width` integer

The width of the computer display.

`environment` string

The type of computer environment to control.

`type` string

The type of the computer use tool. Always `computer_use_preview` .

MCP tool object

Give the model access to additional tools via remote Model Context Protocol (MCP) servers. Learn more about MCP.

Hide properties

`server_label` string

A label for this MCP server, used to identify it in tool calls.

`server_url` string

The URL for the MCP server.

`type` string

The type of the MCP tool. Always `mcp` .

`allowed_tools` array or object

List of allowed tool names or a filter object.

Hide possible types

MCP allowed tools array

A string array of allowed tool names

MCP allowed tools filter object

A filter object to specify which tools are allowed.

Hide properties

tool_names array
List of allowed tool names.

headers object or null
Optional HTTP headers to send to the MCP server. Use for authentication or other purposes.

require_approval object or string
Specify which of the MCP server's tools require approval.

Hide possible types

MCP tool approval filter object

Hide properties

always object
A list of tools that always require approval.

Hide properties

tool_names array
List of tools that require approval.

never object
A list of tools that never require approval.

Hide properties

tool_names array
List of tools that do not require approval.

MCP tool approval setting string
Specify a single approval policy for all tools. One of `always` or `never`. When set to `always`, all tools will require approval. When set to `never`, all tools will not require approval.

server_description string
Optional description of the MCP server, used to provide more context.

Code interpreter object
A tool that runs Python code to help generate a response to a prompt.

Hide properties

container string or object
The code interpreter container. Can be a container ID or an object that specifies uploaded file IDs to make available to your code.

Hide possible types

string
The container ID.

CodeInterpreterContainerAuto object
Configuration for a code interpreter container. Optionally specify the IDs of the files to run the code on.

Hide properties

type string
Always `auto`.

file_ids array
An optional list of uploaded files to make available to your code.

type string
The type of the code interpreter tool. Always `code_interpreter`.

Image generation tool object
A tool that generates images using a model like `gpt-image-1`.

Hide properties

type string
The type of the image generation tool. Always `image_generation`.

background string
Background type for the generated image. One of `transparent`, `opaque`, or `auto`. Default: `auto`.

input_image_mask object
Optional mask for inpainting. Contains `image_url` (string, optional) and `file_id` (string, optional).

Hide properties

file_id string

File ID for the mask image.

image_url string
Base64-encoded mask image.

model string
The image generation model to use. Default: `gpt-image-1`.

moderation string
Moderation level for the generated image. Default: `auto`.

output_compression integer
Compression level for the output image. Default: 100.

output_format string
The output format of the generated image. One of `png`, `webp`, or `jpeg`. Default: `png`.

partial_images integer
Number of partial images to generate in streaming mode, from 0 (default value) to 3.

quality string
The quality of the generated image. One of `low`, `medium`, `high`, or `auto`. Default: `auto`.

size string
The size of the generated image. One of `1024x1024`, `1024x1536`, `1536x1024`, or `auto`. Default: `auto`.

Local shell tool object
A tool that allows the model to execute shell commands in a local environment.

Hide properties

type string
The type of the local shell tool. Always `local_shell`.

top_p number or null
An alternative to sampling with temperature, called nucleus sampling, where the model considers the results of the tokens with top_p probability mass. So 0.1 means only the tokens comprising the top 10% probability mass are considered.

We generally recommend altering this or `temperature` but not both.

background boolean or null
Whether to run the model response in the background. Learn more.

max_output_tokens integer or null
An upper bound for the number of tokens that can be generated for a response, including visible output tokens and reasoning tokens.

max_tool_calls integer or null
The maximum number of total calls to built-in tools that can be processed in a response. This maximum number applies across all built-in tool calls, not per individual tool. Any further attempts to call a tool by the model will be ignored.

output_text string or null SDK Only
SDK-only convenience property that contains the aggregated text output from all `output_text` items in the `output` array, if any are present. Supported in the Python and JavaScript SDKs.

previous_response_id string or null
The unique ID of the previous response to the model. Use this to create multi-turn conversations. Learn more about conversation state.

prompt object or null
Reference to a prompt template and its variables. Learn more.

Hide properties

id string
The unique identifier of the prompt template to use.

variables map
Optional map of values to substitute in for variables in your prompt. The substitution values can either be strings, or other Response input types like images or files.

version string or null
Optional version of the prompt template.

reasoning object or null
o-series models only

Configuration options for reasoning models.

Hide properties

effort string or null
o-series models only

Constrains effort on reasoning for reasoning models. Currently supported values are `low`, `medium`, and `high`. Reducing reasoning effort can result in faster responses and fewer tokens used on reasoning in a response.

`generate_summary` Deprecated string or null

Deprecated: use `summary` instead.

A summary of the reasoning performed by the model. This can be useful for debugging and understanding the model's reasoning process. One of `auto`, `concise`, or `detailed`.

`summary` string or null

A summary of the reasoning performed by the model. This can be useful for debugging and understanding the model's reasoning process. One of `auto`, `concise`, or `detailed`.

`service_tier` string or null

Specifies the processing type used for serving the request.

> If set to 'auto', then the request will be processed with the service tier configured in the Project settings. Unless otherwise configured, the Project will use 'default'.
>
> If set to 'default', then the requset will be processed with the standard pricing and performance for the selected model.
>
> If set to 'flex' or 'priority', then the request will be processed with the corresponding service tier. Contact sales to learn more about Priority processing.
>
> When not set, the default behavior is 'auto'.

When the `service_tier` parameter is set, the response body will include the `service_tier` value based on the processing mode actually used to serve the request. This response value may be different from the value set in the parameter.

`status` string

The status of the response generation. One of `completed`, `failed`, `in_progress`, `cancelled`, `queued`, or `incomplete`.

`text` object

Configuration options for a text response from the model. Can be plain text or structured JSON data. Learn more:

> Text inputs and outputs
>
> Structured Outputs

^ Hide properties

`format` object

An object specifying the format that the model must output.

Configuring `{ "type": "json_schema" }` enables Structured Outputs, which ensures the model will match your supplied JSON schema. Learn more in the Structured Outputs guide.

The default format is `{ "type": "text" }` with no additional options.

Not recommended for gpt-4o and newer models:

Setting to `{ "type": "json_object" }` enables the older JSON mode, which ensures the message the model generates is valid JSON. Using `json_schema` is preferred for models that support it.

Hide possible types

Text object
Default response format. Used to generate text responses.

Hide properties

type string
The type of response format being defined. Always `text`.

JSON schema object
JSON Schema response format. Used to generate structured JSON responses. Learn more about Structured Outputs.

Hide properties

name string
The name of the response format. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

schema object
The schema for the response format, described as a JSON Schema object. Learn how to build JSON schemas here.

type string
The type of response format being defined. Always `json_schema`.

description string
A description of what the response format is for, used by the model to determine how to respond in the format.

strict boolean or null
Whether to enable strict schema adherence when generating the output. If set to true, the model will always follow the exact schema defined in the `schema` field. Only a subset of JSON Schema is supported when `strict` is `true`. To learn more, read the Structured Outputs guide.

JSON object object

JSON object response format. An older method of generating JSON responses. Using `json_schema` is recommended for models that support it. Note that the model will not generate JSON without a system or user message instructing it to do so.

Hide properties

type string

The type of response format being defined. Always `json_object`.

top_logprobs integer or null

An integer between 0 and 20 specifying the number of most likely tokens to return at each token position, each with an associated log probability.

truncation string or null

The truncation strategy to use for the model response.

`auto`: If the context of this response and previous ones exceeds the model's context window size, the model will truncate the response to fit the context window by dropping input items in the middle of the conversation.

`disabled` (default): If a model response will exceed the context window size for a model, the request will fail with a 400 error.

usage object

Represents token usage details including input tokens, output tokens, a breakdown of output tokens, and the total tokens used.

Hide properties

input_tokens integer

The number of input tokens.

input_tokens_details object

A detailed breakdown of the input tokens.

Hide properties

cached_tokens integer

The number of tokens that were retrieved from the cache. More on prompt caching.

output_tokens integer

The number of output tokens.

output_tokens_details object
A detailed breakdown of the output tokens.

Hide properties

reasoning_tokens integer
The number of reasoning tokens.

total_tokens integer
The total number of tokens used.

user string
A stable identifier for your end-users. Used to boost cache hit rates by better bucketing similar requests and to help OpenAI detect and prevent abuse. Learn more.

sequence_number integer
The sequence number for this event.

type string
The type of the event. Always 'response.queued'.

OBJECT response.queued

```
{
  "type": "response.queued",
  "response": {
    "id": "res_123",
    "status": "queued",
    "created_at": "2021-01-01T00:00:00Z",
    "updated_at": "2021-01-01T00:00:00Z"
  },
  "sequence_number": 1
}
```

response.reasoning.delta

Emitted when there is a delta (partial update) to the reasoning content.

content_index integer
The index of the reasoning content part within the output item.

delta object
The partial update to the reasoning content.

item_id string
The unique identifier of the item for which reasoning is being updated.

output_index integer
The index of the output item in the response's output array.

sequence_number integer
The sequence number of this event.

type string
The type of the event. Always 'response.reasoning.delta'.

OBJECT response.reasoning.delta

```
{
  "type": "response.reasoning.delta",
  "item_id": "item-abc",
  "output_index": 0,
  "content_index": 0,
  "delta": {
    "text": "This is a test delta"
  },
  "sequence_number": 1
}
```

response.reasoning.done

Emitted when the reasoning content is finalized for an item.

content_index integer
The index of the reasoning content part within the output item.

item_id string

The unique identifier of the item for which reasoning is finalized.

output_index integer
The index of the output item in the response's output array.

sequence_number integer
The sequence number of this event.

text string
The finalized reasoning text.

type string
The type of the event. Always 'response.reasoning.done'.

OBJECT response.reasoning.done

```
{
  "type": "response.reasoning.done",
  "item_id": "item-abc",
  "output_index": 0,
  "content_index": 0,
  "text": "This is a test reasoning",
  "sequence_number": 1
}
```

response.reasoning_summary.delta

Emitted when there is a delta (partial update) to the reasoning summary content.

delta object
The partial update to the reasoning summary content.

item_id string
The unique identifier of the item for which the reasoning summary is being updated.

output_index integer
The index of the output item in the response's output array.

sequence_number integer
The sequence number of this event.

summary_index integer
The index of the summary part within the output item.

type string
The type of the event. Always 'response.reasoning_summary.delta'.

OBJECT response.reasoning_summary.delta

```
{
  "type": "response.reasoning_summary.delta",
  "item_id": "item-abc",
  "output_index": 0,
  "summary_index": 0,
  "delta": {
    "text": "delta text"
  },
  "sequence_number": 1
}
```

response.reasoning_summary.done

Emitted when the reasoning summary content is finalized for an item.

item_id string
The unique identifier of the item for which the reasoning summary is finalized.

output_index integer
The index of the output item in the response's output array.

sequence_number integer
The sequence number of this event.

summary_index integer
The index of the summary part within the output item.

text string

The finalized reasoning summary text.

type string
The type of the event. Always 'response.reasoning_summary.done'.

OBJECT response.reasoning_summary.done

```
{
  "type": "response.reasoning_summary.done",
  "item_id": "item-abc",
  "output_index": 0,
  "summary_index": 0,
  "text": "This is a test reasoning summary",
  "sequence_number": 1
}
```

error

Emitted when an error occurs.

code string or null
The error code.

message string
The error message.

param string or null
The error parameter.

sequence_number integer
The sequence number of this event.

type string
The type of the event. Always `error`.

OBJECT error

```
{
  "type": "error",
  "code": "ERR_SOMETHING",
```

```
  "message": "Something went wrong",
  "param": null,
  "sequence_number": 1
}
```

Chat Completions

The Chat Completions API endpoint will generate a model response from a list of messages comprising a conversation.

Related guides:

- Quickstart
- Text inputs and outputs
- Image inputs
- Audio inputs and outputs
- Structured Outputs
- Function calling
- Conversation state

Starting a new project? We recommend trying Responses to take advantage of the latest OpenAI platform features. Compare Chat Completions with Responses.

Create chat completion

```
POST https://api.openai.com/v1/chat/completions
```

Starting a new project? We recommend trying Responses to take advantage of the latest OpenAI platform features. Compare Chat Completions with Responses.

Creates a model response for the given chat conversation. Learn more in the text generation, vision, and audio guides.
Parameter support can differ depending on the model used to generate the response, particularly for newer reasoning models. Parameters that are only supported for reasoning models are noted below. For the current state of unsupported parameters in reasoning

models, refer to the reasoning guide.

Request body

messages array Required

A list of messages comprising the conversation so far. Depending on the model you use, different message types (modalities) are supported, like text, images, and audio.

Hide possible types

Developer message object

Developer-provided instructions that the model should follow, regardless of messages sent by the user. With o1 models and newer, `developer` messages replace the previous `system` messages.

Hide properties

content string or array Required

The contents of the developer message.

Hide possible types

Text content string

The contents of the developer message.

Array of content parts array

An array of content parts with a defined type. For developer messages, only type `text` is supported.

Hide properties

text string Required

The text content.

type string Required

The type of the content part.

role string Required

The role of the messages author, in this case `developer`.

name string Optional

An optional name for the participant. Provides the model information to differentiate between participants of the same role.

System message object

Developer-provided instructions that the model should follow, regardless of messages sent by the user. With o1 models and newer, use `developer` messages for this purpose instead.

Hide properties

content string or array Required
The contents of the system message.

Hide possible types

Text content string
The contents of the system message.

Array of content parts array
An array of content parts with a defined type. For system messages, only type `text` is supported.

Hide possible types

Text content part object
Learn about text inputs.

Hide properties

text string Required
The text content.

type string Required
The type of the content part.

role string Required
The role of the messages author, in this case `system`.

name string Optional
An optional name for the participant. Provides the model information to differentiate between participants of the same role.

User message object
Messages sent by an end user, containing prompts or additional context information.

Hide properties

content string or array Required
The contents of the user message.

Hide possible types

Text content string
The text contents of the message.

Array of content parts array
An array of content parts with a defined type. Supported options differ based on the model being used to generate the response. Can contain text, image, or audio inputs.

Hide possible types

Text content part object
Learn about text inputs.

Hide properties

text string Required
The text content.

type string Required
The type of the content part.

Image content part object
Learn about image inputs.

Hide properties

image_url object Required

Hide properties

url string Required
Either a URL of the image or the base64 encoded image data.

detail string Optional Defaults to auto
Specifies the detail level of the image. Learn more in the Vision guide.

type string Required
The type of the content part.

Audio content part object
Learn about audio inputs.

Hide properties

input_audio object Required

Hide properties

data string Required
Base64 encoded audio data.

format string Required
The format of the encoded audio data. Currently supports "wav" and "mp3".

type string Required
The type of the content part. Always `input_audio` .

File content part object
Learn about file inputs for text generation.

Hide properties

file object Required

Hide properties

file_data string Optional
The base64 encoded file data, used when passing the file to the model as a string.

file_id string Optional
The ID of an uploaded file to use as input.

filename string Optional
The name of the file, used when passing the file to the model as a string.

type string Required
The type of the content part. Always `file` .

role string Required
The role of the messages author, in this case `user` .

name string Optional
An optional name for the participant. Provides the model information to differentiate between participants of the same role.

Assistant message object
Messages sent by the model in response to user messages.

Hide properties

role string Required
The role of the messages author, in this case `assistant`.

audio object or null Optional
Data about a previous audio response from the model. Learn more.

Hide properties

id string Required
Unique identifier for a previous audio response from the model.

content string or array Optional
The contents of the assistant message. Required unless `tool_calls` or `function_call` is specified.

Hide possible types

Text content string
The contents of the assistant message.

Array of content parts array
An array of content parts with a defined type. Can be one or more of type `text`, or exactly one of type `refusal`.

Hide possible types

Text content part object
Learn about text inputs.

Hide properties

text string Required
The text content.

type string Required
The type of the content part.

Refusal content part object

Hide properties

refusal string Required
The refusal message generated by the model.

type string Required
The type of the content part.

function_call Deprecated object or null Optional
Deprecated and replaced by `tool_calls`. The name and arguments of a function that should be called, as generated by the model.

Hide properties

arguments string Required
The arguments to call the function with, as generated by the model in JSON format. Note that the model does not always generate valid JSON, and may hallucinate parameters not defined by your function schema. Validate the arguments in your code before calling your function.

name string Required
The name of the function to call.

name string Optional
An optional name for the participant. Provides the model information to differentiate between participants of the same role.

refusal string or null Optional
The refusal message by the assistant.

tool_calls array Optional
The tool calls generated by the model, such as function calls.

Hide properties

function object Required
The function that the model called.

Hide properties

arguments string Required
The arguments to call the function with, as generated by the model in JSON format. Note that the model does not always generate valid JSON, and may hallucinate parameters not

defined by your function schema. Validate the arguments in your code before calling your function.

name string Required
The name of the function to call.

id string Required
The ID of the tool call.

type string Required
The type of the tool. Currently, only `function` is supported.

Tool message object

Hide properties

content string or array Required
The contents of the tool message.

Hide possible types

Text content string
The contents of the tool message.

Array of content parts array
An array of content parts with a defined type. For tool messages, only type `text` is supported.

Hide possible types

Text content part object
Learn about text inputs.

Hide properties

text string Required
The text content.

type string Required
The type of the content part.

role string Required
The role of the messages author, in this case `tool`.

tool_call_id string Required
Tool call that this message is responding to.

Function message object Deprecated

Hide properties

content string or null Required
The contents of the function message.

name string Required
The name of the function to call.

role string Required
The role of the messages author, in this case `function`.

model string Required
Model ID used to generate the response, like `gpt-4o` or `o3`. OpenAI offers a wide range of models with different capabilities, performance characteristics, and price points. Refer to the model guide to browse and compare available models.

audio object or null Optional
Parameters for audio output. Required when audio output is requested with `modalities: ["audio"]`. Learn more.

Hide properties

format string Required
Specifies the output audio format. Must be one of `wav`, `mp3`, `flac`, `opus`, or `pcm16`.

voice string Required
The voice the model uses to respond. Supported voices are `alloy`, `ash`, `ballad`, `coral`, `echo`, `fable`, `nova`, `onyx`, `sage`, and `shimmer`.

frequency_penalty number or null Optional Defaults to 0
Number between -2.0 and 2.0. Positive values penalize new tokens based on their existing frequency in the text so far, decreasing the model's likelihood to repeat the same line verbatim.

function_call Deprecated string or object Optional
Deprecated in favor of `tool_choice`.

Controls which (if any) function is called by the model.

`none` means the model will not call a function and instead generates a message.

`auto` means the model can pick between generating a message or calling a function.

Specifying a particular function via `{"name": "my_function"}` forces the model to call that function.

`none` is the default when no functions are present. `auto` is the default if functions are present.

Hide possible types

string

`none` means the model will not call a function and instead generates a message. `auto` means the model can pick between generating a message or calling a function.

object

Specifying a particular function via `{"name": "my_function"}` forces the model to call that function.

Hide properties

name string Required

The name of the function to call.

functions Deprecated array Optional

Deprecated in favor of `tools`.

A list of functions the model may generate JSON inputs for.

Hide properties

name string Required

The name of the function to be called. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

description string Optional

A description of what the function does, used by the model to choose when and how to call the function.

parameters object Optional

The parameters the functions accepts, described as a JSON Schema object. See the guide for examples, and the JSON Schema reference for documentation about the format.

Omitting `parameters` defines a function with an empty parameter list.

logit_bias map Optional Defaults to null

Modify the likelihood of specified tokens appearing in the completion.

Accepts a JSON object that maps tokens (specified by their token ID in the tokenizer) to an associated bias value from -100 to 100. Mathematically, the bias is added to the logits generated by the model prior to sampling. The exact effect will vary per model, but values between -1 and 1 should decrease or increase

likelihood of selection; values like -100 or 100 should result in a ban or exclusive selection of the relevant token.

`logprobs` boolean or null Optional Defaults to false
Whether to return log probabilities of the output tokens or not. If true, returns the log probabilities of each output token returned in the `content` of `message`.

`max_completion_tokens` integer or null Optional
An upper bound for the number of tokens that can be generated for a completion, including visible output tokens and reasoning tokens.

`max_tokens` Deprecated integer or null Optional
The maximum number of tokens that can be generated in the chat completion. This value can be used to control costs for text generated via API.

This value is now deprecated in favor of `max_completion_tokens`, and is not compatible with o-series models.

`metadata` map Optional
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

`modalities` array or null Optional
Output types that you would like the model to generate. Most models are capable of generating text, which is the default:

`["text"]`

The `gpt-4o-audio-preview` model can also be used to generate audio. To request that this model generate both text and audio responses, you can use:

`["text", "audio"]`

`n` integer or null Optional Defaults to 1
How many chat completion choices to generate for each input message. Note that you will be charged based on the number of generated tokens across all of the choices. Keep `n` as `1` to minimize costs.

`parallel_tool_calls` boolean Optional Defaults to true
Whether to enable parallel function calling during tool use.

`prediction` object Optional

Configuration for a Predicted Output, which can greatly improve response times when large parts of the model response are known ahead of time. This is most common when you are regenerating a file with only minor changes to most of the content.

Hide possible types

Static Content object
Static predicted output content, such as the content of a text file that is being regenerated.

Hide properties

content string or array Required
The content that should be matched when generating a model response. If generated tokens would match this content, the entire model response can be returned much more quickly.

Hide possible types

Text content string
The content used for a Predicted Output. This is often the text of a file you are regenerating with minor changes.

Array of content parts array
An array of content parts with a defined type. Supported options differ based on the model being used to generate the response. Can contain text inputs.

Hide properties

text string Required
The text content.

type string Required
The type of the content part.

type string Required
The type of the predicted content you want to provide. This type is currently always `content`.

presence_penalty number or null Optional Defaults to 0
Number between -2.0 and 2.0. Positive values penalize new tokens based on whether they appear in the text so far, increasing the model's likelihood to talk about new topics.

reasoning_effort string or null Optional Defaults to medium
o-series models only

Constrains effort on reasoning for reasoning models. Currently supported values are `low`, `medium`, and `high`. Reducing reasoning effort can result in faster responses and fewer tokens used on reasoning in a

response.

response_format object Optional
An object specifying the format that the model must output.

Setting to `{ "type": "json_schema", "json_schema": {...} }` enables Structured Outputs which ensures the model will match your supplied JSON schema. Learn more in the Structured Outputs guide.

Setting to `{ "type": "json_object" }` enables the older JSON mode, which ensures the message the model generates is valid JSON. Using `json_schema` is preferred for models that support it.

Hide possible types

Text object
Default response format. Used to generate text responses.

Hide properties

type string Required
The type of response format being defined. Always `text`.

JSON schema object
JSON Schema response format. Used to generate structured JSON responses. Learn more about Structured Outputs.

Hide properties

json_schema object Required
Structured Outputs configuration options, including a JSON Schema.

Hide properties

name string Required
The name of the response format. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

description string Optional
A description of what the response format is for, used by the model to determine how to respond in the format.

schema object Optional
The schema for the response format, described as a JSON Schema object. Learn how to build JSON schemas here.

strict boolean or null Optional Defaults to false
Whether to enable strict schema adherence when generating the output. If set to true, the model will always follow the exact schema defined in the `schema` field. Only a subset of JSON Schema

is supported when `strict` is `true`. To learn more, read the Structured Outputs guide.

type string Required
The type of response format being defined. Always `json_schema`.

JSON object object
JSON object response format. An older method of generating JSON responses. Using `json_schema` is recommended for models that support it. Note that the model will not generate JSON without a system or user message instructing it to do so.

Hide properties

type string Required
The type of response format being defined. Always `json_object`.

seed integer or null Optional
This feature is in Beta. If specified, our system will make a best effort to sample deterministically, such that repeated requests with the same `seed` and parameters should return the same result. Determinism is not guaranteed, and you should refer to the `system_fingerprint` response parameter to monitor changes in the backend.

service_tier string or null Optional Defaults to auto
Specifies the processing type used for serving the request.

If set to 'auto', then the request will be processed with the service tier configured in the Project settings. Unless otherwise configured, the Project will use 'default'.

If set to 'default', then the requset will be processed with the standard pricing and performance for the selected model.

If set to 'flex' or 'priority', then the request will be processed with the corresponding service tier. Contact sales to learn more about Priority processing.

When not set, the default behavior is 'auto'.

When the `service_tier` parameter is set, the response body will include the `service_tier` value based on the processing mode actually used to serve the request. This response value may be different from the value set in the parameter.

stop string / array / null Optional Defaults to null
Not supported with latest reasoning models `o3` and `o4-mini`.

Up to 4 sequences where the API will stop generating further tokens. The returned text will not contain the stop sequence.

store boolean or null Optional Defaults to false

Whether or not to store the output of this chat completion request for use in our model distillation or evals products.

Supports text and image inputs. Note: image inputs over 10MB will be dropped.

stream boolean or null Optional Defaults to false

If set to true, the model response data will be streamed to the client as it is generated using server-sent events. See the Streaming section below for more information, along with the streaming responses guide for more information on how to handle the streaming events.

stream_options object or null Optional Defaults to null

Options for streaming response. Only set this when you set `stream: true`.

Hide properties

include_usage boolean Optional

If set, an additional chunk will be streamed before the `data: [DONE]` message. The `usage` field on this chunk shows the token usage statistics for the entire request, and the `choices` field will always be an empty array.

All other chunks will also include a `usage` field, but with a null value. **NOTE:** If the stream is interrupted, you may not receive the final usage chunk which contains the total token usage for the request.

temperature number or null Optional Defaults to 1

What sampling temperature to use, between 0 and 2. Higher values like 0.8 will make the output more random, while lower values like 0.2 will make it more focused and deterministic. We generally recommend altering this or `top_p` but not both.

tool_choice string or object Optional

Controls which (if any) tool is called by the model. `none` means the model will not call any tool and instead generates a message. `auto` means the model can pick between generating a message or calling one or more tools. `required` means the model must call one or more tools. Specifying a particular tool via `{"type": "function", "function": {"name": "my_function"}}` forces the model to call that tool.

`none` is the default when no tools are present. `auto` is the default if tools are present.

Hide possible types

string

`none` means the model will not call any tool and instead generates a message. `auto` means the model can pick between generating a message or calling one or more tools. `required` means the model must call one or more tools.

object

Specifies a tool the model should use. Use to force the model to call a specific function.

Hide properties

function object Required

Hide properties

name string Required

The name of the function to call.

type string Required

The type of the tool. Currently, only `function` is supported.

tools array Optional

A list of tools the model may call. Currently, only functions are supported as a tool. Use this to provide a list of functions the model may generate JSON inputs for. A max of 128 functions are supported.

Hide properties

function object Required

Hide properties

name string Required

The name of the function to be called. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

description string Optional

A description of what the function does, used by the model to choose when and how to call the function.

parameters object Optional

The parameters the functions accepts, described as a JSON Schema object. See the guide for examples, and the JSON Schema reference for documentation about the format.

Omitting `parameters` defines a function with an empty parameter list.

strict boolean or null Optional Defaults to false

Whether to enable strict schema adherence when generating the function call. If set to true, the model will follow the exact schema defined in the `parameters` field. Only a subset of JSON Schema is supported when `strict` is `true`. Learn more about Structured Outputs in the function calling guide.

type string Required

The type of the tool. Currently, only `function` is supported.

`top_logprobs` integer or null Optional
An integer between 0 and 20 specifying the number of most likely tokens to return at each token position, each with an associated log probability. `logprobs` must be set to `true` if this parameter is used.

`top_p` number or null Optional Defaults to 1
An alternative to sampling with temperature, called nucleus sampling, where the model considers the results of the tokens with top_p probability mass. So 0.1 means only the tokens comprising the top 10% probability mass are considered.

We generally recommend altering this or `temperature` but not both.

`user` string Optional
A stable identifier for your end-users. Used to boost cache hit rates by better bucketing similar requests and to help OpenAI detect and prevent abuse. Learn more.

`web_search_options` object Optional
This tool searches the web for relevant results to use in a response. Learn more about the web search tool.

^ Hide properties

`search_context_size` string Optional Defaults to medium
High level guidance for the amount of context window space to use for the search. One of `low`, `medium`, or `high`. `medium` is the default.

`user_location` object or null Optional
Approximate location parameters for the search.

^ Hide properties

`approximate` object Required
Approximate location parameters for the search.

^ Hide properties

`city` string Optional
Free text input for the city of the user, e.g. `San Francisco`.

`country` string Optional
The two-letter ISO country code of the user, e.g. `US`.

`region` string Optional
Free text input for the region of the user, e.g. `California`.

timezone string Optional
The IANA timezone of the user, e.g. `America/Los_Angeles` .

type string Required
The type of location approximation. Always `approximate` .

Returns

Returns a chat completion object, or a streamed sequence of chat completion chunk objects if the request is streamed.

Default | Image input | Streaming | Functions | Logprobs

Example request gpt-4.1 curl

```
curl https://api.openai.com/v1/chat/completions \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -d '{
    "model": "gpt-4.1",
    "messages": [
      {
        "role": "developer",
        "content": "You are a helpful assistant."
      },
      {
        "role": "user",
        "content": "Hello!"
      }
    ]
  }'
```

Response

```
{
  "id": "chatcmpl-B9MBs8CjcvOU2jLn4n570S5qMJKcT",
  "object": "chat.completion",
  "created": 1741569952,
  "model": "gpt-4.1-2025-04-14",
  "choices": [
```

```
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "Hello! How can I assist you today?",
        "refusal": null,
        "annotations": []
      },
      "logprobs": null,
      "finish_reason": "stop"
    }
  ],
  "usage": {
    "prompt_tokens": 19,
    "completion_tokens": 10,
    "total_tokens": 29,
    "prompt_tokens_details": {
      "cached_tokens": 0,
      "audio_tokens": 0
    },
    "completion_tokens_details": {
      "reasoning_tokens": 0,
      "audio_tokens": 0,
      "accepted_prediction_tokens": 0,
      "rejected_prediction_tokens": 0
    }
  },
  "service_tier": "default"
}
```

Get chat completion

`GET https://api.openai.com/v1/chat/completions/{completion_id}`

Get a stored chat completion. Only Chat Completions that have been created with the `store` parameter set to `true` will be returned.

Path parameters

completion_id string Required
The ID of the chat completion to retrieve.

Returns

The ChatCompletion object matching the specified ID.

Example request curl

```curl
curl https://api.openai.com/v1/chat/completions/chatcmpl-abc123 \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json"
```

Response


```json
{
  "object": "chat.completion",
  "id": "chatcmpl-abc123",
  "model": "gpt-4o-2024-08-06",
  "created": 1738960610,
  "request_id": "req_ded8ab984ec4bf840f37566c1011c417",
  "tool_choice": null,
  "usage": {
    "total_tokens": 31,
    "completion_tokens": 18,
    "prompt_tokens": 13
  },
  "seed": 4944116822809979520,
  "top_p": 1.0,
  "temperature": 1.0,
  "presence_penalty": 0.0,
  "frequency_penalty": 0.0,
  "system_fingerprint": "fp_50cad350e4",
  "input_user": null,
  "service_tier": "default",
  "tools": null,
  "metadata": {},
  "choices": [
    {
      "index": 0,
      "message": {
        "content": "Mind of circuits hum,  \nLearning patterns in silence—  \nFuture's qui
        "role": "assistant",
        "tool_calls": null,
        "function_call": null
      },
      "finish_reason": "stop",
```

```
      "logprobs": null
    }
  ],
  "response_format": null
}
```

Get chat messages

`GET https://api.openai.com/v1/chat/completions/{completion_id}/messages`

Get the messages in a stored chat completion. Only Chat Completions that have been created with the `store` parameter set to `true` will be returned.

Path parameters

`completion_id` string Required
The ID of the chat completion to retrieve messages from.

Query parameters

`after` string Optional
Identifier for the last message from the previous pagination request.

`limit` integer Optional Defaults to 20
Number of messages to retrieve.

`order` string Optional Defaults to asc
Sort order for messages by timestamp. Use `asc` for ascending order or `desc` for descending order. Defaults to `asc`.

Returns

A list of messages for the specified chat completion.

Example request curl

```
curl https://api.openai.com/v1/chat/completions/chat_abc123/messages \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
  "object": "list",
  "data": [
    {
      "id": "chatcmpl-AyPNinnUqUDYo9SAdA52NobMflmj2-0",
      "role": "user",
      "content": "write a haiku about ai",
      "name": null,
      "content_parts": null
    }
  ],
  "first_id": "chatcmpl-AyPNinnUqUDYo9SAdA52NobMflmj2-0",
  "last_id": "chatcmpl-AyPNinnUqUDYo9SAdA52NobMflmj2-0",
  "has_more": false
}
```

List Chat Completions

`GET https://api.openai.com/v1/chat/completions`

List stored Chat Completions. Only Chat Completions that have been stored with the `store` parameter set to `true` will be returned.

Query parameters

`after` string Optional
Identifier for the last chat completion from the previous pagination request.

`limit` integer Optional Defaults to 20
Number of Chat Completions to retrieve.

`metadata` map Optional
A list of metadata keys to filter the Chat Completions by. Example:

`metadata[key1]=value1&metadata[key2]=value2`

model string Optional
The model used to generate the Chat Completions.

order string Optional Defaults to asc
Sort order for Chat Completions by timestamp. Use `asc` for ascending order or `desc` for descending order. Defaults to `asc`.

Returns

A list of Chat Completions matching the specified filters.

Example request curl

```
curl https://api.openai.com/v1/chat/completions \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
  "object": "list",
  "data": [
    {
      "object": "chat.completion",
      "id": "chatcmpl-AyPNinnUqUDYo9SAdA52NobMflmj2",
      "model": "gpt-4.1-2025-04-14",
      "created": 1738960610,
      "request_id": "req_ded8ab984ec4bf840f37566c1011c417",
      "tool_choice": null,
      "usage": {
        "total_tokens": 31,
        "completion_tokens": 18,
        "prompt_tokens": 13
      },
      "seed": 4944116822809979520,
      "top_p": 1.0,
      "temperature": 1.0,
      "presence_penalty": 0.0,
      "frequency_penalty": 0.0,
      "system_fingerprint": "fp_50cad350e4",
```

```
      "input_user": null,
      "service_tier": "default",
      "tools": null,
      "metadata": {},
      "choices": [
        {
          "index": 0,
          "message": {
            "content": "Mind of circuits hum,  \nLearning patterns in silence—  \nFuture's
            "role": "assistant",
            "tool_calls": null,
            "function_call": null
          },
          "finish_reason": "stop",
          "logprobs": null
        }
      ],
      "response_format": null
    }
  ],
  "first_id": "chatcmpl-AyPNinnUqUDYo9SAdA52NobMflmj2",
  "last_id": "chatcmpl-AyPNinnUqUDYo9SAdA52NobMflmj2",
  "has_more": false
}
```


Update chat completion

`POST https://api.openai.com/v1/chat/completions/{completion_id}`

Modify a stored chat completion. Only Chat Completions that have been created with the `store` parameter set to `true` can be modified. Currently, the only supported modification is to update the `metadata` field.

Path parameters

`completion_id` string Required
The ID of the chat completion to update.

Request body

metadata map Required

Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

Returns

The ChatCompletion object matching the specified ID.

Example request curl

```
curl -X POST https://api.openai.com/v1/chat/completions/chat_abc123 \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{"metadata": {"foo": "bar"}}'
```

Response

```
{
  "object": "chat.completion",
  "id": "chatcmpl-AyPNinnUqUDYo9SAdA52NobMflmj2",
  "model": "gpt-4o-2024-08-06",
  "created": 1738960610,
  "request_id": "req_ded8ab984ec4bf840f37566c1011c417",
  "tool_choice": null,
  "usage": {
    "total_tokens": 31,
    "completion_tokens": 18,
    "prompt_tokens": 13
  },
  "seed": 4944116822809979520,
  "top_p": 1.0,
  "temperature": 1.0,
  "presence_penalty": 0.0,
  "frequency_penalty": 0.0,
  "system_fingerprint": "fp_50cad350e4",
  "input_user": null,
  "service_tier": "default",
  "tools": null,
  "metadata": {
    "foo": "bar"
```

```
  },
  "choices": [
    {
      "index": 0,
      "message": {
        "content": "Mind of circuits hum,  \nLearning patterns in silence—  \nFuture's qui
        "role": "assistant",
        "tool_calls": null,
        "function_call": null
      },
      "finish_reason": "stop",
      "logprobs": null
    }
  ],
  "response_format": null
}
```

Delete chat completion

`DELETE https://api.openai.com/v1/chat/completions/{completion_id}`

Delete a stored chat completion. Only Chat Completions that have been created with the `store` parameter set to `true` can be deleted.

Path parameters

completion_id string Required
The ID of the chat completion to delete.

Returns

A deletion confirmation object.

Example request curl

```curl
curl -X DELETE https://api.openai.com/v1/chat/completions/chat_abc123 \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
  "object": "chat.completion.deleted",
  "id": "chatcmpl-AyPNinnUqUDYo9SAdA52NobMflmj2",
  "deleted": true
}
```

The chat completion object

Represents a chat completion response returned by model, based on the provided input.

choices array

A list of chat completion choices. Can be more than one if `n` is greater than 1.

Hide properties

finish_reason string

The reason the model stopped generating tokens. This will be `stop` if the model hit a natural stop point or a provided stop sequence, `length` if the maximum number of tokens specified in the request was reached, `content_filter` if content was omitted due to a flag from our content filters, `tool_calls` if the model called a tool, or `function_call` (deprecated) if the model called a function.

index integer

The index of the choice in the list of choices.

logprobs object or null

Log probability information for the choice.

Hide properties

content array or null

A list of message content tokens with log probability information.

Hide properties

bytes array or null

A list of integers representing the UTF-8 bytes representation of the token. Useful in instances where characters are represented by multiple tokens and their byte representations must be combined to generate the correct text representation. Can be `null` if there is no bytes representation for the token.

logprob number

The log probability of this token, if it is within the top 20 most likely tokens. Otherwise, the value `-9999.0` is used to signify that the token is very unlikely.

token string
The token.

top_logprobs array
List of the most likely tokens and their log probability, at this token position. In rare cases, there may be fewer than the number of requested `top_logprobs` returned.

Hide properties

bytes array or null
A list of integers representing the UTF-8 bytes representation of the token. Useful in instances where characters are represented by multiple tokens and their byte representations must be combined to generate the correct text representation. Can be `null` if there is no bytes representation for the token.

logprob number
The log probability of this token, if it is within the top 20 most likely tokens. Otherwise, the value `-9999.0` is used to signify that the token is very unlikely.

token string
The token.

refusal array or null
A list of message refusal tokens with log probability information.

Hide properties

bytes array or null
A list of integers representing the UTF-8 bytes representation of the token. Useful in instances where characters are represented by multiple tokens and their byte representations must be combined to generate the correct text representation. Can be `null` if there is no bytes representation for the token.

logprob number
The log probability of this token, if it is within the top 20 most likely tokens. Otherwise, the value `-9999.0` is used to signify that the token is very unlikely.

token string
The token.

top_logprobs array

List of the most likely tokens and their log probability, at this token position. In rare cases, there may be fewer than the number of requested `top_logprobs` returned.

Hide properties

bytes array or null
A list of integers representing the UTF-8 bytes representation of the token. Useful in instances where characters are represented by multiple tokens and their byte representations must be combined to generate the correct text representation. Can be `null` if there is no bytes representation for the token.

logprob number
The log probability of this token, if it is within the top 20 most likely tokens. Otherwise, the value `-9999.0` is used to signify that the token is very unlikely.

token string
The token.

message object
A chat completion message generated by the model.

Hide properties

content string or null
The contents of the message.

refusal string or null
The refusal message generated by the model.

role string
The role of the author of this message.

annotations array
Annotations for the message, when applicable, as when using the web search tool.

Hide properties

type string
The type of the URL citation. Always `url_citation`.

url_citation object
A URL citation when using web search.

Hide properties

end_index integer
The index of the last character of the URL citation in the message.

start_index integer
The index of the first character of the URL citation in the message.

title string
The title of the web resource.

url string
The URL of the web resource.

audio object or null
If the audio output modality is requested, this object contains data about the audio response from the model. Learn more.

Hide properties

data string
Base64 encoded audio bytes generated by the model, in the format specified in the request.

expires_at integer
The Unix timestamp (in seconds) for when this audio response will no longer be accessible on the server for use in multi-turn conversations.

id string
Unique identifier for this audio response.

transcript string
Transcript of the audio generated by the model.

function_call Deprecated object
Deprecated and replaced by `tool_calls`. The name and arguments of a function that should be called, as generated by the model.

Hide properties

arguments string
The arguments to call the function with, as generated by the model in JSON format. Note that the model does not always generate valid JSON, and may hallucinate parameters not defined by your function schema. Validate the arguments in your code before calling your function.

name string

The name of the function to call.

tool_calls array

The tool calls generated by the model, such as function calls.

Hide properties

function object

The function that the model called.

Hide properties

arguments string

The arguments to call the function with, as generated by the model in JSON format. Note that the model does not always generate valid JSON, and may hallucinate parameters not defined by your function schema. Validate the arguments in your code before calling your function.

name string

The name of the function to call.

id string

The ID of the tool call.

type string

The type of the tool. Currently, only `function` is supported.

created integer

The Unix timestamp (in seconds) of when the chat completion was created.

id string

A unique identifier for the chat completion.

model string

The model used for the chat completion.

object string

The object type, which is always `chat.completion`.

service_tier string or null

Specifies the processing type used for serving the request.

If set to 'auto', then the request will be processed with the service tier configured in the Project settings. Unless otherwise configured, the Project will use 'default'.

If set to 'default', then the requset will be processed with the standard pricing and performance for the selected model.

If set to 'flex' or 'priority', then the request will be processed with the corresponding service tier. Contact sales to learn more about Priority processing.

When not set, the default behavior is 'auto'.

When the `service_tier` parameter is set, the response body will include the `service_tier` value based on the processing mode actually used to serve the request. This response value may be different from the value set in the parameter.

`system_fingerprint` string
This fingerprint represents the backend configuration that the model runs with.

Can be used in conjunction with the `seed` request parameter to understand when backend changes have been made that might impact determinism.

`usage` object
Usage statistics for the completion request.

^ Hide properties

`completion_tokens` integer
Number of tokens in the generated completion.

`prompt_tokens` integer
Number of tokens in the prompt.

`total_tokens` integer
Total number of tokens used in the request (prompt + completion).

`completion_tokens_details` object
Breakdown of tokens used in a completion.

^ Hide properties

`accepted_prediction_tokens` integer
When using Predicted Outputs, the number of tokens in the prediction that appeared in the completion.

`audio_tokens` integer
Audio input tokens generated by the model.

`reasoning_tokens` integer

Tokens generated by the model for reasoning.

rejected_prediction_tokens integer
When using Predicted Outputs, the number of tokens in the prediction that did not appear in the completion. However, like reasoning tokens, these tokens are still counted in the total completion tokens for purposes of billing, output, and context window limits.

prompt_tokens_details object
Breakdown of tokens used in the prompt.

Hide properties

audio_tokens integer
Audio input tokens present in the prompt.

cached_tokens integer
Cached tokens present in the prompt.

OBJECT The chat completion object


```
{
  "id": "chatcmpl-B9MHDbslfkBeAs8l4bebGdFOJ6PeG",
  "object": "chat.completion",
  "created": 1741570283,
  "model": "gpt-4o-2024-08-06",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "The image shows a wooden boardwalk path running through a lush green f
        "refusal": null,
        "annotations": []
      },
      "logprobs": null,
      "finish_reason": "stop"
    }
  ],
  "usage": {
    "prompt_tokens": 1117,
    "completion_tokens": 46,
    "total_tokens": 1163,
    "prompt_tokens_details": {
      "cached_tokens": 0,
```

```
      "audio_tokens": 0
    },
    "completion_tokens_details": {
      "reasoning_tokens": 0,
      "audio_tokens": 0,
      "accepted_prediction_tokens": 0,
      "rejected_prediction_tokens": 0
    }
  },
  "service_tier": "default",
  "system_fingerprint": "fp_fc9f1d7035"
}
```

The chat completion list object

An object representing a list of Chat Completions.

data array

An array of chat completion objects.

Hide properties

choices array

A list of chat completion choices. Can be more than one if `n` is greater than 1.

Hide properties

finish_reason string

The reason the model stopped generating tokens. This will be `stop` if the model hit a natural stop point or a provided stop sequence, `length` if the maximum number of tokens specified in the request was reached, `content_filter` if content was omitted due to a flag from our content filters, `tool_calls` if the model called a tool, or `function_call` (deprecated) if the model called a function.

index integer

The index of the choice in the list of choices.

logprobs object or null

Log probability information for the choice.

Hide properties

content array or null

A list of message content tokens with log probability information.

Hide properties

bytes array or null
A list of integers representing the UTF-8 bytes representation of the token. Useful in instances where characters are represented by multiple tokens and their byte representations must be combined to generate the correct text representation. Can be `null` if there is no bytes representation for the token.

logprob number
The log probability of this token, if it is within the top 20 most likely tokens. Otherwise, the value `-9999.0` is used to signify that the token is very unlikely.

token string
The token.

top_logprobs array
List of the most likely tokens and their log probability, at this token position. In rare cases, there may be fewer than the number of requested `top_logprobs` returned.

Hide properties

bytes array or null
A list of integers representing the UTF-8 bytes representation of the token. Useful in instances where characters are represented by multiple tokens and their byte representations must be combined to generate the correct text representation. Can be `null` if there is no bytes representation for the token.

logprob number
The log probability of this token, if it is within the top 20 most likely tokens. Otherwise, the value `-9999.0` is used to signify that the token is very unlikely.

token string
The token.

refusal array or null
A list of message refusal tokens with log probability information.

Hide properties

bytes array or null
A list of integers representing the UTF-8 bytes representation of the token. Useful in instances where characters are represented by multiple tokens and their byte representations must be combined to generate the correct text representation. Can be `null` if there is no bytes representation for the token.

logprob number
The log probability of this token, if it is within the top 20 most likely tokens. Otherwise, the value `-9999.0` is used to signify that the token is very unlikely.

token string
The token.

top_logprobs array
List of the most likely tokens and their log probability, at this token position. In rare cases, there may be fewer than the number of requested `top_logprobs` returned.

Hide properties

bytes array or null
A list of integers representing the UTF-8 bytes representation of the token. Useful in instances where characters are represented by multiple tokens and their byte representations must be combined to generate the correct text representation. Can be `null` if there is no bytes representation for the token.

logprob number
The log probability of this token, if it is within the top 20 most likely tokens. Otherwise, the value `-9999.0` is used to signify that the token is very unlikely.

token string
The token.

message object
A chat completion message generated by the model.

Hide properties

content string or null
The contents of the message.

refusal string or null
The refusal message generated by the model.

role string
The role of the author of this message.

annotations array
Annotations for the message, when applicable, as when using the web search tool.

Hide properties

type string
The type of the URL citation. Always `url_citation`.

url_citation object
A URL citation when using web search.

Hide properties

end_index integer
The index of the last character of the URL citation in the message.

start_index integer
The index of the first character of the URL citation in the message.

title string
The title of the web resource.

url string
The URL of the web resource.

audio object or null
If the audio output modality is requested, this object contains data about the audio response from the model. Learn more.

Hide properties

data string
Base64 encoded audio bytes generated by the model, in the format specified in the request.

expires_at integer
The Unix timestamp (in seconds) for when this audio response will no longer be accessible on the server for use in multi-turn conversations.

id string
Unique identifier for this audio response.

transcript string
Transcript of the audio generated by the model.

function_call Deprecated object
Deprecated and replaced by `tool_calls`. The name and arguments of a function that should be called, as generated by the model.

Hide properties

arguments string
The arguments to call the function with, as generated by the model in JSON format. Note that the model does not always generate valid JSON, and may hallucinate parameters not defined by your function schema. Validate the arguments in your code before calling your function.

name string
The name of the function to call.

tool_calls array
The tool calls generated by the model, such as function calls.

Hide properties

function object
The function that the model called.

Hide properties

arguments string
The arguments to call the function with, as generated by the model in JSON format. Note that the model does not always generate valid JSON, and may hallucinate parameters not defined by your function schema. Validate the arguments in your code before calling your function.

name string
The name of the function to call.

id string
The ID of the tool call.

type string
The type of the tool. Currently, only `function` is supported.

`created` integer
The Unix timestamp (in seconds) of when the chat completion was created.

`id` string
A unique identifier for the chat completion.

`model` string
The model used for the chat completion.

`object` string
The object type, which is always `chat.completion`.

`service_tier` string or null
Specifies the processing type used for serving the request.

> If set to 'auto', then the request will be processed with the service tier configured in the Project settings. Unless otherwise configured, the Project will use 'default'.
>
> If set to 'default', then the requset will be processed with the standard pricing and performance for the selected model.
>
> If set to 'flex' or 'priority', then the request will be processed with the corresponding service tier. Contact sales to learn more about Priority processing.
>
> When not set, the default behavior is 'auto'.

When the `service_tier` parameter is set, the response body will include the `service_tier` value based on the processing mode actually used to serve the request. This response value may be different from the value set in the parameter.

`system_fingerprint` string
This fingerprint represents the backend configuration that the model runs with.

Can be used in conjunction with the `seed` request parameter to understand when backend changes have been made that might impact determinism.

`usage` object
Usage statistics for the completion request.

^ Hide properties

`completion_tokens` integer
Number of tokens in the generated completion.

`prompt_tokens` integer
Number of tokens in the prompt.

total_tokens integer
Total number of tokens used in the request (prompt + completion).

completion_tokens_details object
Breakdown of tokens used in a completion.

Hide properties

accepted_prediction_tokens integer
When using Predicted Outputs, the number of tokens in the prediction that appeared in the completion.

audio_tokens integer
Audio input tokens generated by the model.

reasoning_tokens integer
Tokens generated by the model for reasoning.

rejected_prediction_tokens integer
When using Predicted Outputs, the number of tokens in the prediction that did not appear in the completion. However, like reasoning tokens, these tokens are still counted in the total completion tokens for purposes of billing, output, and context window limits.

prompt_tokens_details object
Breakdown of tokens used in the prompt.

Hide properties

audio_tokens integer
Audio input tokens present in the prompt.

cached_tokens integer
Cached tokens present in the prompt.

first_id string
The identifier of the first chat completion in the data array.

has_more boolean
Indicates whether there are more Chat Completions available.

last_id string

The identifier of the last chat completion in the data array.

object string

The type of this object. It is always set to "list".

OBJECT The chat completion list object


```
{
  "object": "list",
  "data": [
    {
      "object": "chat.completion",
      "id": "chatcmpl-AyPNinnUqUDYo9SAdA52NobMflmj2",
      "model": "gpt-4o-2024-08-06",
      "created": 1738960610,
      "request_id": "req_ded8ab984ec4bf840f37566c1011c417",
      "tool_choice": null,
      "usage": {
        "total_tokens": 31,
        "completion_tokens": 18,
        "prompt_tokens": 13
      },
      "seed": 4944116822809979520,
      "top_p": 1.0,
      "temperature": 1.0,
      "presence_penalty": 0.0,
      "frequency_penalty": 0.0,
      "system_fingerprint": "fp_50cad350e4",
      "input_user": null,
      "service_tier": "default",
      "tools": null,
      "metadata": {},
      "choices": [
        {
          "index": 0,
          "message": {
            "content": "Mind of circuits hum,  \nLearning patterns in silence—  \nFuture's
            "role": "assistant",
            "tool_calls": null,
            "function_call": null
          },
          "finish_reason": "stop",
          "logprobs": null
        }
```

```
      ],
      "response_format": null
    }
  ],
  "first_id": "chatcmpl-AyPNinnUqUDYo9SAdA52NobMflmj2",
  "last_id": "chatcmpl-AyPNinnUqUDYo9SAdA52NobMflmj2",
  "has_more": false
}
```

The chat completion message list object

An object representing a list of chat completion messages.

data array

An array of chat completion message objects.

Hide properties

content string or null

The contents of the message.

id string

The identifier of the chat message.

refusal string or null

The refusal message generated by the model.

role string

The role of the author of this message.

annotations array

Annotations for the message, when applicable, as when using the web search tool.

Hide properties

type string

The type of the URL citation. Always `url_citation`.

url_citation object

A URL citation when using web search.

Hide properties

end_index integer

The index of the last character of the URL citation in the message.

start_index integer
The index of the first character of the URL citation in the message.

title string
The title of the web resource.

url string
The URL of the web resource.

audio object or null
If the audio output modality is requested, this object contains data about the audio response from the model. Learn more.

Hide properties

data string
Base64 encoded audio bytes generated by the model, in the format specified in the request.

expires_at integer
The Unix timestamp (in seconds) for when this audio response will no longer be accessible on the server for use in multi-turn conversations.

id string
Unique identifier for this audio response.

transcript string
Transcript of the audio generated by the model.

function_call Deprecated object
Deprecated and replaced by `tool_calls`. The name and arguments of a function that should be called, as generated by the model.

Hide properties

arguments string
The arguments to call the function with, as generated by the model in JSON format. Note that the model does not always generate valid JSON, and may hallucinate parameters not defined by your function schema. Validate the arguments in your code before calling your function.

name string
The name of the function to call.

tool_calls array
The tool calls generated by the model, such as function calls.

Hide properties

function object
The function that the model called.

Hide properties

arguments string
The arguments to call the function with, as generated by the model in JSON format. Note that the model does not always generate valid JSON, and may hallucinate parameters not defined by your function schema. Validate the arguments in your code before calling your function.

name string
The name of the function to call.

id string
The ID of the tool call.

type string
The type of the tool. Currently, only `function` is supported.

first_id string
The identifier of the first chat message in the data array.

has_more boolean
Indicates whether there are more chat messages available.

last_id string
The identifier of the last chat message in the data array.

object string
The type of this object. It is always set to "list".

```
OBJECT The chat completion message list object
```

```
{
  "object": "list",
  "data": [
    {
      "id": "chatcmpl-AyPNinnUqUDYo9SAdA52NobMflmj2-0",
      "role": "user",
      "content": "write a haiku about ai",
      "name": null,
      "content_parts": null
    }
  ],
  "first_id": "chatcmpl-AyPNinnUqUDYo9SAdA52NobMflmj2-0",
  "last_id": "chatcmpl-AyPNinnUqUDYo9SAdA52NobMflmj2-0",
  "has_more": false
}
```

Streaming

Stream Chat Completions in real time. Receive chunks of completions returned from the model using server-sent events. Learn more.

The chat completion chunk object

Represents a streamed chunk of a chat completion response returned by the model, based on the provided input. Learn more.

choices array

A list of chat completion choices. Can contain more than one elements if `n` is greater than 1. Can also be empty for the last chunk if you set `stream_options: {"include_usage": true}`.

Hide properties

delta object

A chat completion delta generated by streamed model responses.

Hide properties

content string or null

The contents of the chunk message.

function_call Deprecated object

Deprecated and replaced by `tool_calls`. The name and arguments of a function that should be called, as generated by the model.

Hide properties

arguments string
The arguments to call the function with, as generated by the model in JSON format. Note that the model does not always generate valid JSON, and may hallucinate parameters not defined by your function schema. Validate the arguments in your code before calling your function.

name string
The name of the function to call.

refusal string or null
The refusal message generated by the model.

role string
The role of the author of this message.

tool_calls array

Hide properties

index integer

function object

Hide properties

arguments string
The arguments to call the function with, as generated by the model in JSON format. Note that the model does not always generate valid JSON, and may hallucinate parameters not defined by your function schema. Validate the arguments in your code before calling your function.

name string
The name of the function to call.

id string
The ID of the tool call.

type string
The type of the tool. Currently, only `function` is supported.

finish_reason string or null
The reason the model stopped generating tokens. This will be `stop` if the model hit a natural stop point or a provided stop sequence, `length` if the maximum number of tokens specified in the request was reached, `content_filter` if content was omitted due to a flag from our content filters, `tool_calls` if the model called a tool, or `function_call` (deprecated) if the model called a function.

index integer
The index of the choice in the list of choices.

logprobs object or null
Log probability information for the choice.

Hide properties

content array or null
A list of message content tokens with log probability information.

Hide properties

bytes array or null
A list of integers representing the UTF-8 bytes representation of the token. Useful in instances where characters are represented by multiple tokens and their byte representations must be combined to generate the correct text representation. Can be `null` if there is no bytes representation for the token.

logprob number
The log probability of this token, if it is within the top 20 most likely tokens. Otherwise, the value `-9999.0` is used to signify that the token is very unlikely.

token string
The token.

top_logprobs array
List of the most likely tokens and their log probability, at this token position. In rare cases, there may be fewer than the number of requested `top_logprobs` returned.

Hide properties

bytes array or null
A list of integers representing the UTF-8 bytes representation of the token. Useful in instances where characters are represented by multiple tokens and their byte representations must be combined to generate the correct text representation. Can be `null` if there is no bytes representation for the token.

logprob number

The log probability of this token, if it is within the top 20 most likely tokens. Otherwise, the value `-9999.0` is used to signify that the token is very unlikely.

`token` string
The token.

`refusal` array or null
A list of message refusal tokens with log probability information.

Hide properties

`bytes` array or null
A list of integers representing the UTF-8 bytes representation of the token. Useful in instances where characters are represented by multiple tokens and their byte representations must be combined to generate the correct text representation. Can be `null` if there is no bytes representation for the token.

`logprob` number
The log probability of this token, if it is within the top 20 most likely tokens. Otherwise, the value `-9999.0` is used to signify that the token is very unlikely.

`token` string
The token.

`top_logprobs` array
List of the most likely tokens and their log probability, at this token position. In rare cases, there may be fewer than the number of requested `top_logprobs` returned.

Hide properties

`bytes` array or null
A list of integers representing the UTF-8 bytes representation of the token. Useful in instances where characters are represented by multiple tokens and their byte representations must be combined to generate the correct text representation. Can be `null` if there is no bytes representation for the token.

`logprob` number
The log probability of this token, if it is within the top 20 most likely tokens. Otherwise, the value `-9999.0` is used to signify that the token is very unlikely.

`token` string
The token.

`created` integer
The Unix timestamp (in seconds) of when the chat completion was created. Each chunk has the same timestamp.

`id` string
A unique identifier for the chat completion. Each chunk has the same ID.

`model` string
The model to generate the completion.

`object` string
The object type, which is always `chat.completion.chunk`.

`service_tier` string or null
Specifies the processing type used for serving the request.

> If set to 'auto', then the request will be processed with the service tier configured in the Project settings. Unless otherwise configured, the Project will use 'default'.
>
> If set to 'default', then the requset will be processed with the standard pricing and performance for the selected model.
>
> If set to 'flex' or 'priority', then the request will be processed with the corresponding service tier. Contact sales to learn more about Priority processing.
>
> When not set, the default behavior is 'auto'.

When the `service_tier` parameter is set, the response body will include the `service_tier` value based on the processing mode actually used to serve the request. This response value may be different from the value set in the parameter.

`system_fingerprint` string
This fingerprint represents the backend configuration that the model runs with. Can be used in conjunction with the `seed` request parameter to understand when backend changes have been made that might impact determinism.

`usage` object or null
Usage statistics for the completion request.

Hide properties

`completion_tokens` integer
Number of tokens in the generated completion.

prompt_tokens integer
Number of tokens in the prompt.

total_tokens integer
Total number of tokens used in the request (prompt + completion).

completion_tokens_details object
Breakdown of tokens used in a completion.

^ Hide properties

accepted_prediction_tokens integer
When using Predicted Outputs, the number of tokens in the prediction that appeared in the completion.

audio_tokens integer
Audio input tokens generated by the model.

reasoning_tokens integer
Tokens generated by the model for reasoning.

rejected_prediction_tokens integer
When using Predicted Outputs, the number of tokens in the prediction that did not appear in the completion. However, like reasoning tokens, these tokens are still counted in the total completion tokens for purposes of billing, output, and context window limits.

prompt_tokens_details object
Breakdown of tokens used in the prompt.

^ Hide properties

audio_tokens integer
Audio input tokens present in the prompt.

cached_tokens integer
Cached tokens present in the prompt.

OBJECT The chat completion chunk object


```
{"id":"chatcmpl-123","object":"chat.completion.chunk","created":1694268190,"model":"gpt-4o-

{"id":"chatcmpl-123","object":"chat.completion.chunk","created":1694268190,"model":"gpt-4o-
```

```
....

{"id":"chatcmpl-123","object":"chat.completion.chunk","created":1694268190,"model":"gpt-4o-
```

Webhook Events

Webhooks are HTTP requests sent by OpenAI to a URL you specify when certain events happen during the course of API usage.

Learn more about webhooks.

response.completed

Sent when a background response has been completed.

created_at integer
The Unix timestamp (in seconds) of when the model response was completed.

data object
Event data payload.

Hide properties

id string
The unique ID of the model response.

id string
The unique ID of the event.

object string
The object of the event. Always `event`.

type string
The type of the event. Always `response.completed`.

OBJECT response.completed

```
{
  "id": "evt_abc123",
  "type": "response.completed",
  "created_at": 1719168000,
  "data": {
    "id": "resp_abc123"
  }
}
```

response.cancelled

Sent when a background response has been cancelled.

created_at integer
The Unix timestamp (in seconds) of when the model response was cancelled.

data object
Event data payload.

^ Hide properties

id string
The unique ID of the model response.

id string
The unique ID of the event.

object string
The object of the event. Always `event`.

type string
The type of the event. Always `response.cancelled`.

OBJECT response.cancelled

```
{
  "id": "evt_abc123",
  "type": "response.cancelled",
```

```
  "created_at": 1719168000,
  "data": {
    "id": "resp_abc123"
  }
}
```

response.failed

Sent when a background response has failed.

created_at integer
The Unix timestamp (in seconds) of when the model response failed.

data object
Event data payload.

Hide properties

id string
The unique ID of the model response.

id string
The unique ID of the event.

object string
The object of the event. Always `event`.

type string
The type of the event. Always `response.failed`.

OBJECT response.failed

```
{
  "id": "evt_abc123",
  "type": "response.failed",
  "created_at": 1719168000,
  "data": {
    "id": "resp_abc123"
  }
}
```

response.incomplete

Sent when a background response has been interrupted.

created_at integer
The Unix timestamp (in seconds) of when the model response was interrupted.

data object
Event data payload.

Hide properties

id string
The unique ID of the model response.

id string
The unique ID of the event.

object string
The object of the event. Always `event`.

type string
The type of the event. Always `response.incomplete`.

OBJECT response.incomplete

```
{
  "id": "evt_abc123",
  "type": "response.incomplete",
  "created_at": 1719168000,
  "data": {
    "id": "resp_abc123"
  }
}
```

batch.completed

Sent when a batch API request has been completed.

created_at integer
The Unix timestamp (in seconds) of when the batch API request was completed.

data object
Event data payload.

Hide properties

id string
The unique ID of the batch API request.

id string
The unique ID of the event.

object string
The object of the event. Always `event`.

type string
The type of the event. Always `batch.completed`.

OBJECT batch.completed

```
{
  "id": "evt_abc123",
  "type": "batch.completed",
  "created_at": 1719168000,
  "data": {
    "id": "batch_abc123"
  }
}
```

batch.cancelled

Sent when a batch API request has been cancelled.

created_at integer
The Unix timestamp (in seconds) of when the batch API request was cancelled.

data object
Event data payload.

Hide properties

id string

The unique ID of the batch API request.

id string
The unique ID of the event.

object string
The object of the event. Always `event`.

type string
The type of the event. Always `batch.cancelled`.

OBJECT batch.cancelled

```
{
  "id": "evt_abc123",
  "type": "batch.cancelled",
  "created_at": 1719168000,
  "data": {
    "id": "batch_abc123"
  }
}
```

batch.expired

Sent when a batch API request has expired.

created_at integer
The Unix timestamp (in seconds) of when the batch API request expired.

data object
Event data payload.

Hide properties

id string
The unique ID of the batch API request.

id string
The unique ID of the event.

object string
The object of the event. Always `event`.

type string
The type of the event. Always `batch.expired`.

OBJECT batch.expired

```
{
  "id": "evt_abc123",
  "type": "batch.expired",
  "created_at": 1719168000,
  "data": {
    "id": "batch_abc123"
  }
}
```

batch.failed

Sent when a batch API request has failed.

created_at integer
The Unix timestamp (in seconds) of when the batch API request failed.

data object
Event data payload.

Hide properties

id string
The unique ID of the batch API request.

id string
The unique ID of the event.

object string
The object of the event. Always `event`.

type string
The type of the event. Always `batch.failed`.

OBJECT batch.failed

```
{
  "id": "evt_abc123",
  "type": "batch.failed",
  "created_at": 1719168000,
  "data": {
    "id": "batch_abc123"
  }
}
```

fine_tuning.job.succeeded

Sent when a fine-tuning job has succeeded.

created_at integer
The Unix timestamp (in seconds) of when the fine-tuning job succeeded.

data object
Event data payload.

Hide properties

id string
The unique ID of the fine-tuning job.

id string
The unique ID of the event.

object string
The object of the event. Always `event`.

type string

The type of the event. Always `fine_tuning.job.succeeded`.

OBJECT `fine_tuning.job.succeeded`

```
{
  "id": "evt_abc123",
  "type": "fine_tuning.job.succeeded",
  "created_at": 1719168000,
  "data": {
    "id": "ftjob_abc123"
  }
}
```

fine_tuning.job.failed

Sent when a fine-tuning job has failed.

created_at integer
The Unix timestamp (in seconds) of when the fine-tuning job failed.

data object
Event data payload.

Hide properties

id string
The unique ID of the fine-tuning job.

id string
The unique ID of the event.

object string
The object of the event. Always `event`.

type string
The type of the event. Always `fine_tuning.job.failed`.

OBJECT `fine_tuning.job.failed`

```
{
```

```
  "id": "evt_abc123",
  "type": "fine_tuning.job.failed",
  "created_at": 1719168000,
  "data": {
    "id": "ftjob_abc123"
  }
}
```

fine_tuning.job.cancelled

Sent when a fine-tuning job has been cancelled.

created_at integer
The Unix timestamp (in seconds) of when the fine-tuning job was cancelled.

data object
Event data payload.

Hide properties

id string
The unique ID of the fine-tuning job.

id string
The unique ID of the event.

object string
The object of the event. Always `event`.

type string
The type of the event. Always `fine_tuning.job.cancelled`.

OBJECT fine_tuning.job.cancelled

```
{
  "id": "evt_abc123",
  "type": "fine_tuning.job.cancelled",
  "created_at": 1719168000,
  "data": {
    "id": "ftjob_abc123"
  }
```

```
}
```

eval.run.succeeded

Sent when an eval run has succeeded.

created_at integer
The Unix timestamp (in seconds) of when the eval run succeeded.

data object
Event data payload.

Hide properties

id string
The unique ID of the eval run.

id string
The unique ID of the event.

object string
The object of the event. Always `event`.

type string
The type of the event. Always `eval.run.succeeded`.

OBJECT eval.run.succeeded

```
{
  "id": "evt_abc123",
  "type": "eval.run.succeeded",
  "created_at": 1719168000,
  "data": {
    "id": "evalrun_abc123"
  }
}
```

eval.run.failed

Sent when an eval run has failed.

created_at integer
The Unix timestamp (in seconds) of when the eval run failed.

data object
Event data payload.

Hide properties

id string
The unique ID of the eval run.

id string
The unique ID of the event.

object string
The object of the event. Always `event`.

type string
The type of the event. Always `eval.run.failed`.

OBJECT eval.run.failed

```
{
  "id": "evt_abc123",
  "type": "eval.run.failed",
  "created_at": 1719168000,
  "data": {
    "id": "evalrun_abc123"
  }
}
```

eval.run.canceled

Sent when an eval run has been canceled.

created_at integer
The Unix timestamp (in seconds) of when the eval run was canceled.

data object
Event data payload.

Hide properties

id string
The unique ID of the eval run.

id string
The unique ID of the event.

object string
The object of the event. Always `event`.

type string
The type of the event. Always `eval.run.canceled`.

OBJECT eval.run.canceled

```
{
  "id": "evt_abc123",
  "type": "eval.run.canceled",
  "created_at": 1719168000,
  "data": {
    "id": "evalrun_abc123"
  }
}
```

Realtime Beta

Communicate with a GPT-4o class model in real time using WebRTC or WebSockets. Supports text and audio inputs and ouputs, along with audio transcriptions. Learn more about the Realtime API.

Session tokens

REST API endpoint to generate ephemeral session tokens for use in client-side applications.

Create session

```
POST https://api.openai.com/v1/realtime/sessions
```

Create an ephemeral API token for use in client-side applications with the Realtime API. Can be configured with the same session parameters as the `session.update` client event.
It responds with a session object, plus a `client_secret` key which contains a usable ephemeral API token that can be used to authenticate browser clients for the Realtime API.

Request body

`client_secret` object Optional
Configuration options for the generated client secret.

Hide properties

`expires_after` object Optional
Configuration for the ephemeral token expiration.

Hide properties

`anchor` string Required
The anchor point for the ephemeral token expiration. Only `created_at` is currently supported.

`seconds` integer Optional Defaults to 600
The number of seconds from the anchor point to the expiration. Select a value between `10` and `7200`.

`input_audio_format` string Optional Defaults to pcm16
The format of input audio. Options are `pcm16`, `g711_ulaw`, or `g711_alaw`. For `pcm16`, input audio must be 16-bit PCM at a 24kHz sample rate, single channel (mono), and little-endian byte order.

`input_audio_noise_reduction` object Optional Defaults to null
Configuration for input audio noise reduction. This can be set to `null` to turn off. Noise reduction filters audio added to the input audio buffer before it is sent to VAD and the model. Filtering the audio can improve VAD and turn detection accuracy (reducing false positives) and model performance by improving perception of the input audio.

Hide properties

`type` string Optional
Type of noise reduction. `near_field` is for close-talking microphones such as headphones, `far_field` is for far-field microphones such as laptop or conference room microphones.

input_audio_transcription object Optional

Configuration for input audio transcription, defaults to off and can be set to `null` to turn off once on. Input audio transcription is not native to the model, since the model consumes audio directly. Transcription runs asynchronously through the /audio/transcriptions endpoint and should be treated as guidance of input audio content rather than precisely what the model heard. The client can optionally set the language and prompt for transcription, these offer additional guidance to the transcription service.

^ Hide properties

language string Optional

The language of the input audio. Supplying the input language in ISO-639-1 (e.g. `en`) format will improve accuracy and latency.

model string Optional

The model to use for transcription, current options are `gpt-4o-transcribe` , `gpt-4o-mini-transcribe` , and `whisper-1` .

prompt string Optional

An optional text to guide the model's style or continue a previous audio segment. For `whisper-1` , the prompt is a list of keywords. For `gpt-4o-transcribe` models, the prompt is a free text string, for example "expect words related to technology".

instructions string Optional

The default system instructions (i.e. system message) prepended to model calls. This field allows the client to guide the model on desired responses. The model can be instructed on response content and format, (e.g. "be extremely succinct", "act friendly", "here are examples of good responses") and on audio behavior (e.g. "talk quickly", "inject emotion into your voice", "laugh frequently"). The instructions are not guaranteed to be followed by the model, but they provide guidance to the model on the desired behavior.

Note that the server sets default instructions which will be used if this field is not set and are visible in the `session.created` event at the start of the session.

max_response_output_tokens integer or "inf" Optional

Maximum number of output tokens for a single assistant response, inclusive of tool calls. Provide an integer between 1 and 4096 to limit output tokens, or `inf` for the maximum available tokens for a given model. Defaults to `inf` .

modalities Optional

The set of modalities the model can respond with. To disable audio, set this to ["text"].

model string Optional

The Realtime model used for this session.

output_audio_format string Optional Defaults to pcm16
The format of output audio. Options are `pcm16`, `g711_ulaw`, or `g711_alaw`. For `pcm16`, output audio is sampled at a rate of 24kHz.

speed number Optional Defaults to 1
The speed of the model's spoken response. 1.0 is the default speed. 0.25 is the minimum speed. 1.5 is the maximum speed. This value can only be changed in between model turns, not while a response is in progress.

temperature number Optional Defaults to 0.8
Sampling temperature for the model, limited to [0.6, 1.2]. For audio models a temperature of 0.8 is highly recommended for best performance.

tool_choice string Optional Defaults to auto
How the model chooses tools. Options are `auto`, `none`, `required`, or specify a function.

tools array Optional
Tools (functions) available to the model.

Hide properties

description string Optional
The description of the function, including guidance on when and how to call it, and guidance about what to tell the user when calling (if anything).

name string Optional
The name of the function.

parameters object Optional
Parameters of the function in JSON Schema.

type string Optional
The type of the tool, i.e. `function`.

tracing "auto" or object Optional
Configuration options for tracing. Set to null to disable tracing. Once tracing is enabled for a session, the configuration cannot be modified.

`auto` will create a trace for the session with default values for the workflow name, group id, and metadata.

Hide possible types

string

Default tracing mode for the session.

Tracing Configuration object
Granular configuration for tracing.

Hide properties

group_id string Optional
The group id to attach to this trace to enable filtering and grouping in the traces dashboard.

metadata object Optional
The arbitrary metadata to attach to this trace to enable filtering in the traces dashboard.

workflow_name string Optional
The name of the workflow to attach to this trace. This is used to name the trace in the traces dashboard.

turn_detection object Optional
Configuration for turn detection, ether Server VAD or Semantic VAD. This can be set to `null` to turn off, in which case the client must manually trigger model response. Server VAD means that the model will detect the start and end of speech based on audio volume and respond at the end of user speech. Semantic VAD is more advanced and uses a turn detection model (in conjuction with VAD) to semantically estimate whether the user has finished speaking, then dynamically sets a timeout based on this probability. For example, if user audio trails off with "uhhm", the model will score a low probability of turn end and wait longer for the user to continue speaking. This can be useful for more natural conversations, but may have a higher latency.

Hide properties

create_response boolean Optional Defaults to true
Whether or not to automatically generate a response when a VAD stop event occurs.

eagerness string Optional Defaults to auto
Used only for `semantic_vad` mode. The eagerness of the model to respond. `low` will wait longer for the user to continue speaking, `high` will respond more quickly. `auto` is the default and is equivalent to `medium`.

interrupt_response boolean Optional Defaults to true
Whether or not to automatically interrupt any ongoing response with output to the default conversation (i.e. `conversation` of `auto`) when a VAD start event occurs.

prefix_padding_ms integer Optional
Used only for `server_vad` mode. Amount of audio to include before the VAD detected speech (in milliseconds). Defaults to 300ms.

silence_duration_ms integer Optional
Used only for `server_vad` mode. Duration of silence to detect speech stop (in milliseconds). Defaults to 500ms. With shorter values the model will respond more quickly, but may jump in on short pauses from the user.

threshold number Optional
Used only for `server_vad` mode. Activation threshold for VAD (0.0 to 1.0), this defaults to 0.5. A higher threshold will require louder audio to activate the model, and thus might perform better in noisy environments.

type string Optional Defaults to server_vad
Type of turn detection.

voice string Optional
The voice the model uses to respond. Voice cannot be changed during the session once the model has responded with audio at least once. Current voice options are `alloy`, `ash`, `ballad`, `coral`, `echo`, `fable`, `onyx`, `nova`, `sage`, `shimmer`, and `verse`.

Returns

The created Realtime session object, plus an ephemeral key

Example request curl

```
curl -X POST https://api.openai.com/v1/realtime/sessions \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "model": "gpt-4o-realtime-preview",
    "modalities": ["audio", "text"],
    "instructions": "You are a friendly assistant."
  }'
```

Response

```
{
  "id": "sess_001",
  "object": "realtime.session",
  "model": "gpt-4o-realtime-preview",
  "modalities": ["audio", "text"],
```

```
  "instructions": "You are a friendly assistant.",
  "voice": "alloy",
  "input_audio_format": "pcm16",
  "output_audio_format": "pcm16",
  "input_audio_transcription": {
      "model": "whisper-1"
  },
  "turn_detection": null,
  "tools": [],
  "tool_choice": "none",
  "temperature": 0.7,
  "max_response_output_tokens": 200,
  "speed": 1.1,
  "tracing": "auto",
  "client_secret": {
    "value": "ek_abc123",
    "expires_at": 1234567890
  }
}
```

Create transcription session

`POST https://api.openai.com/v1/realtime/transcription_sessions`

Create an ephemeral API token for use in client-side applications with the Realtime API specifically for realtime transcriptions. Can be configured with the same session parameters as the `transcription_session.update` client event.
It responds with a session object, plus a `client_secret` key which contains a usable ephemeral API token that can be used to authenticate browser clients for the Realtime API.

Request body

client_secret object Optional
Configuration options for the generated client secret.

Hide properties

expires_at object Optional
Configuration for the ephemeral token expiration.

Hide properties

anchor string Optional Defaults to created_at

The anchor point for the ephemeral token expiration. Only `created_at` is currently supported.

`seconds` integer Optional Defaults to 600
The number of seconds from the anchor point to the expiration. Select a value between `10` and `7200`.

`include` array Optional
The set of items to include in the transcription. Current available items are:

null.

`input_audio_format` string Optional Defaults to pcm16
The format of input audio. Options are `pcm16`, `g711_ulaw`, or `g711_alaw`. For `pcm16`, input audio must be 16-bit PCM at a 24kHz sample rate, single channel (mono), and little-endian byte order.

`input_audio_noise_reduction` object Optional Defaults to null
Configuration for input audio noise reduction. This can be set to `null` to turn off. Noise reduction filters audio added to the input audio buffer before it is sent to VAD and the model. Filtering the audio can improve VAD and turn detection accuracy (reducing false positives) and model performance by improving perception of the input audio.

Hide properties

`type` string Optional
Type of noise reduction. `near_field` is for close-talking microphones such as headphones, `far_field` is for far-field microphones such as laptop or conference room microphones.

`input_audio_transcription` object Optional
Configuration for input audio transcription. The client can optionally set the language and prompt for transcription, these offer additional guidance to the transcription service.

Hide properties

`language` string Optional
The language of the input audio. Supplying the input language in ISO-639-1 (e.g. `en`) format will improve accuracy and latency.

`model` string Optional
The model to use for transcription, current options are `gpt-4o-transcribe`, `gpt-4o-mini-transcribe`, and `whisper-1`.

`prompt` string Optional
An optional text to guide the model's style or continue a previous audio segment. For `whisper-1`, the

prompt is a list of keywords. For `gpt-4o-transcribe` models, the prompt is a free text string, for example "expect words related to technology".

`modalities` Optional
The set of modalities the model can respond with. To disable audio, set this to ["text"].

`turn_detection` object Optional
Configuration for turn detection, ether Server VAD or Semantic VAD. This can be set to `null` to turn off, in which case the client must manually trigger model response. Server VAD means that the model will detect the start and end of speech based on audio volume and respond at the end of user speech. Semantic VAD is more advanced and uses a turn detection model (in conjuction with VAD) to semantically estimate whether the user has finished speaking, then dynamically sets a timeout based on this probability. For example, if user audio trails off with "uhhm", the model will score a low probability of turn end and wait longer for the user to continue speaking. This can be useful for more natural conversations, but may have a higher latency.

Hide properties

`create_response` boolean Optional Defaults to true
Whether or not to automatically generate a response when a VAD stop event occurs. Not available for transcription sessions.

`eagerness` string Optional Defaults to auto
Used only for `semantic_vad` mode. The eagerness of the model to respond. `low` will wait longer for the user to continue speaking, `high` will respond more quickly. `auto` is the default and is equivalent to `medium`.

`interrupt_response` boolean Optional Defaults to true
Whether or not to automatically interrupt any ongoing response with output to the default conversation (i.e. `conversation` of `auto`) when a VAD start event occurs. Not available for transcription sessions.

`prefix_padding_ms` integer Optional
Used only for `server_vad` mode. Amount of audio to include before the VAD detected speech (in milliseconds). Defaults to 300ms.

`silence_duration_ms` integer Optional
Used only for `server_vad` mode. Duration of silence to detect speech stop (in milliseconds). Defaults to 500ms. With shorter values the model will respond more quickly, but may jump in on short pauses from the user.

threshold number Optional
Used only for `server_vad` mode. Activation threshold for VAD (0.0 to 1.0), this defaults to 0.5. A higher threshold will require louder audio to activate the model, and thus might perform better in noisy environments.

type string Optional Defaults to server_vad
Type of turn detection.

Returns

The created Realtime transcription session object, plus an ephemeral key

Example request curl

```
curl -X POST https://api.openai.com/v1/realtime/transcription_sessions \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{}'
```

Response

```
{
  "id": "sess_BBwZc7cFV3XizEyKGDCGL",
  "object": "realtime.transcription_session",
  "modalities": ["audio", "text"],
  "turn_detection": {
    "type": "server_vad",
    "threshold": 0.5,
    "prefix_padding_ms": 300,
    "silence_duration_ms": 200
  },
  "input_audio_format": "pcm16",
  "input_audio_transcription": {
    "model": "gpt-4o-transcribe",
    "language": null,
    "prompt": ""
  },
  "client_secret": null
}
```

The session object

A new Realtime session configuration, with an ephermeral key. Default TTL for keys is one minute.

client_secret object
Ephemeral key returned by the API.

Hide properties

expires_at integer
Timestamp for when the token expires. Currently, all tokens expire after one minute.

value string
Ephemeral key usable in client environments to authenticate connections to the Realtime API. Use this in client-side environments rather than a standard API token, which should only be used server-side.

input_audio_format string
The format of input audio. Options are `pcm16`, `g711_ulaw`, or `g711_alaw`.

input_audio_transcription object
Configuration for input audio transcription, defaults to off and can be set to `null` to turn off once on. Input audio transcription is not native to the model, since the model consumes audio directly. Transcription runs asynchronously and should be treated as rough guidance rather than the representation understood by the model.

Hide properties

model string
The model to use for transcription.

instructions string
The default system instructions (i.e. system message) prepended to model calls. This field allows the client to guide the model on desired responses. The model can be instructed on response content and format, (e.g. "be extremely succinct", "act friendly", "here are examples of good responses") and on audio behavior (e.g. "talk quickly", "inject emotion into your voice", "laugh frequently"). The instructions are not guaranteed to be followed by the model, but they provide guidance to the model on the desired behavior.

Note that the server sets default instructions which will be used if this field is not set and are visible in the `session.created` event at the start of the session.

max_response_output_tokens integer or "inf"
Maximum number of output tokens for a single assistant response, inclusive of tool calls. Provide an integer

between 1 and 4096 to limit output tokens, or `inf` for the maximum available tokens for a given model. Defaults to `inf`.

`modalities`
The set of modalities the model can respond with. To disable audio, set this to ["text"].

`output_audio_format` string
The format of output audio. Options are `pcm16`, `g711_ulaw`, or `g711_alaw`.

`speed` number
The speed of the model's spoken response. 1.0 is the default speed. 0.25 is the minimum speed. 1.5 is the maximum speed. This value can only be changed in between model turns, not while a response is in progress.

`temperature` number
Sampling temperature for the model, limited to [0.6, 1.2]. Defaults to 0.8.

`tool_choice` string
How the model chooses tools. Options are `auto`, `none`, `required`, or specify a function.

`tools` array
Tools (functions) available to the model.

Hide properties

`description` string
The description of the function, including guidance on when and how to call it, and guidance about what to tell the user when calling (if anything).

`name` string
The name of the function.

`parameters` object
Parameters of the function in JSON Schema.

`type` string
The type of the tool, i.e. `function`.

`tracing` "auto" or object
Configuration options for tracing. Set to null to disable tracing. Once tracing is enabled for a session, the configuration cannot be modified.

`auto` will create a trace for the session with default values for the workflow name, group id, and metadata.

Hide possible types

string
Default tracing mode for the session.

Tracing Configuration object
Granular configuration for tracing.

Hide properties

`group_id` string
The group id to attach to this trace to enable filtering and grouping in the traces dashboard.

`metadata` object
The arbitrary metadata to attach to this trace to enable filtering in the traces dashboard.

`workflow_name` string
The name of the workflow to attach to this trace. This is used to name the trace in the traces dashboard.

`turn_detection` object
Configuration for turn detection. Can be set to `null` to turn off. Server VAD means that the model will detect the start and end of speech based on audio volume and respond at the end of user speech.

Hide properties

`prefix_padding_ms` integer
Amount of audio to include before the VAD detected speech (in milliseconds). Defaults to 300ms.

`silence_duration_ms` integer
Duration of silence to detect speech stop (in milliseconds). Defaults to 500ms. With shorter values the model will respond more quickly, but may jump in on short pauses from the user.

`threshold` number
Activation threshold for VAD (0.0 to 1.0), this defaults to 0.5. A higher threshold will require louder audio to activate the model, and thus might perform better in noisy environments.

`type` string
Type of turn detection, only `server_vad` is currently supported.

`voice` string
The voice the model uses to respond. Voice cannot be changed during the session once the model has

responded with audio at least once. Current voice options are `alloy`, `ash`, `ballad`, `coral`, `echo` `sage`, `shimmer` and `verse`.

OBJECT The session object

```
{
  "id": "sess_001",
  "object": "realtime.session",
  "model": "gpt-4o-realtime-preview",
  "modalities": ["audio", "text"],
  "instructions": "You are a friendly assistant.",
  "voice": "alloy",
  "input_audio_format": "pcm16",
  "output_audio_format": "pcm16",
  "input_audio_transcription": {
      "model": "whisper-1"
  },
  "turn_detection": null,
  "tools": [],
  "tool_choice": "none",
  "temperature": 0.7,
  "speed": 1.1,
  "tracing": "auto",
  "max_response_output_tokens": 200,
  "client_secret": {
    "value": "ek_abc123",
    "expires_at": 1234567890
  }
}
```

The transcription session object

A new Realtime transcription session configuration.

When a session is created on the server via REST API, the session object also contains an ephemeral key. Default TTL for keys is 10 minutes. This property is not present when a session is updated via the WebSocket API.

`client_secret` object

Ephemeral key returned by the API. Only present when the session is created on the server via REST API.

^ Hide properties

expires_at integer
Timestamp for when the token expires. Currently, all tokens expire after one minute.

value string
Ephemeral key usable in client environments to authenticate connections to the Realtime API. Use this in client-side environments rather than a standard API token, which should only be used server-side.

input_audio_format string
The format of input audio. Options are `pcm16`, `g711_ulaw`, or `g711_alaw`.

input_audio_transcription object
Configuration of the transcription model.

Hide properties

language string
The language of the input audio. Supplying the input language in ISO-639-1 (e.g. `en`) format will improve accuracy and latency.

model string
The model to use for transcription. Can be `gpt-4o-transcribe`, `gpt-4o-mini-transcribe`, or `whisper-1`.

prompt string
An optional text to guide the model's style or continue a previous audio segment. The prompt should match the audio language.

modalities
The set of modalities the model can respond with. To disable audio, set this to ["text"].

turn_detection object
Configuration for turn detection. Can be set to `null` to turn off. Server VAD means that the model will detect the start and end of speech based on audio volume and respond at the end of user speech.

Hide properties

prefix_padding_ms integer
Amount of audio to include before the VAD detected speech (in milliseconds). Defaults to 300ms.

silence_duration_ms integer
Duration of silence to detect speech stop (in milliseconds). Defaults to 500ms. With shorter values the model will respond more quickly, but may jump in on short pauses from the user.

threshold number
Activation threshold for VAD (0.0 to 1.0), this defaults to 0.5. A higher threshold will require louder audio to activate the model, and thus might perform better in noisy environments.

type string
Type of turn detection, only `server_vad` is currently supported.

OBJECT The transcription session object

```
{
  "id": "sess_BBwZc7cFV3XizEyKGDCGL",
  "object": "realtime.transcription_session",
  "expires_at": 1742188264,
  "modalities": ["audio", "text"],
  "turn_detection": {
    "type": "server_vad",
    "threshold": 0.5,
    "prefix_padding_ms": 300,
    "silence_duration_ms": 200
  },
  "input_audio_format": "pcm16",
  "input_audio_transcription": {
    "model": "gpt-4o-transcribe",
    "language": null,
    "prompt": ""
  },
  "client_secret": null
}
```

Client events

These are events that the OpenAI Realtime WebSocket server will accept from the client.

session.update

Send this event to update the session's default configuration. The client may send this event at any time to update any field, except for `voice`. However, note that once a session has been

initialized with a particular `model`, it can't be changed to another model using `session.update`.

When the server receives a `session.update`, it will respond with a `session.updated` event showing the full, effective configuration. Only the fields that are present are updated. To clear a field like `instructions`, pass an empty string.

event_id string
Optional client-generated ID used to identify this event.

session object
Realtime session object configuration.

Hide properties

client_secret object
Configuration options for the generated client secret.

Hide properties

expires_after object
Configuration for the ephemeral token expiration.

Hide properties

anchor string
The anchor point for the ephemeral token expiration. Only `created_at` is currently supported.

seconds integer
The number of seconds from the anchor point to the expiration. Select a value between `10` and `7200`.

input_audio_format string
The format of input audio. Options are `pcm16`, `g711_ulaw`, or `g711_alaw`. For `pcm16`, input audio must be 16-bit PCM at a 24kHz sample rate, single channel (mono), and little-endian byte order.

input_audio_noise_reduction object
Configuration for input audio noise reduction. This can be set to `null` to turn off. Noise reduction filters audio added to the input audio buffer before it is sent to VAD and the model. Filtering the audio can improve VAD and turn detection accuracy (reducing false positives) and model performance by improving perception of the input audio.

Hide properties

type string

Type of noise reduction. `near_field` is for close-talking microphones such as headphones, `far_field` is for far-field microphones such as laptop or conference room microphones.

`input_audio_transcription` object
Configuration for input audio transcription, defaults to off and can be set to `null` to turn off once on. Input audio transcription is not native to the model, since the model consumes audio directly. Transcription runs asynchronously through the /audio/transcriptions endpoint and should be treated as guidance of input audio content rather than precisely what the model heard. The client can optionally set the language and prompt for transcription, these offer additional guidance to the transcription service.

Hide properties

`language` string
The language of the input audio. Supplying the input language in ISO-639-1 (e.g. `en`) format will improve accuracy and latency.

`model` string
The model to use for transcription, current options are `gpt-4o-transcribe` , `gpt-4o-mini-transcribe` , and `whisper-1` .

`prompt` string
An optional text to guide the model's style or continue a previous audio segment. For `whisper-1` , the prompt is a list of keywords. For `gpt-4o-transcribe` models, the prompt is a free text string, for example "expect words related to technology".

`instructions` string
The default system instructions (i.e. system message) prepended to model calls. This field allows the client to guide the model on desired responses. The model can be instructed on response content and format, (e.g. "be extremely succinct", "act friendly", "here are examples of good responses") and on audio behavior (e.g. "talk quickly", "inject emotion into your voice", "laugh frequently"). The instructions are not guaranteed to be followed by the model, but they provide guidance to the model on the desired behavior.

Note that the server sets default instructions which will be used if this field is not set and are visible in the `session.created` event at the start of the session.

`max_response_output_tokens` integer or "inf"
Maximum number of output tokens for a single assistant response, inclusive of tool calls. Provide an integer between 1 and 4096 to limit output tokens, or `inf` for the maximum available tokens for a given model. Defaults to `inf` .

`modalities`
The set of modalities the model can respond with. To disable audio, set this to ["text"].

model string
The Realtime model used for this session.

output_audio_format string
The format of output audio. Options are `pcm16`, `g711_ulaw`, or `g711_alaw`. For `pcm16`, output audio is sampled at a rate of 24kHz.

speed number
The speed of the model's spoken response. 1.0 is the default speed. 0.25 is the minimum speed. 1.5 is the maximum speed. This value can only be changed in between model turns, not while a response is in progress.

temperature number
Sampling temperature for the model, limited to [0.6, 1.2]. For audio models a temperature of 0.8 is highly recommended for best performance.

tool_choice string
How the model chooses tools. Options are `auto`, `none`, `required`, or specify a function.

tools array
Tools (functions) available to the model.

Hide properties

description string
The description of the function, including guidance on when and how to call it, and guidance about what to tell the user when calling (if anything).

name string
The name of the function.

parameters object
Parameters of the function in JSON Schema.

type string
The type of the tool, i.e. `function`.

tracing "auto" or object
Configuration options for tracing. Set to null to disable tracing. Once tracing is enabled for a session, the configuration cannot be modified.

`auto` will create a trace for the session with default values for the workflow name, group id, and metadata.

Hide possible types

string
Default tracing mode for the session.

Tracing Configuration object
Granular configuration for tracing.

Hide properties

group_id string
The group id to attach to this trace to enable filtering and grouping in the traces dashboard.

metadata object
The arbitrary metadata to attach to this trace to enable filtering in the traces dashboard.

workflow_name string
The name of the workflow to attach to this trace. This is used to name the trace in the traces dashboard.

turn_detection object
Configuration for turn detection, ether Server VAD or Semantic VAD. This can be set to `null` to turn off, in which case the client must manually trigger model response. Server VAD means that the model will detect the start and end of speech based on audio volume and respond at the end of user speech. Semantic VAD is more advanced and uses a turn detection model (in conjuction with VAD) to semantically estimate whether the user has finished speaking, then dynamically sets a timeout based on this probability. For example, if user audio trails off with "uhhm", the model will score a low probability of turn end and wait longer for the user to continue speaking. This can be useful for more natural conversations, but may have a higher latency.

Hide properties

create_response boolean
Whether or not to automatically generate a response when a VAD stop event occurs.

eagerness string
Used only for `semantic_vad` mode. The eagerness of the model to respond. `low` will wait longer for the user to continue speaking, `high` will respond more quickly. `auto` is the default and is equivalent to `medium`.

interrupt_response boolean
Whether or not to automatically interrupt any ongoing response with output to the default conversation (i.e. `conversation` of `auto`) when a VAD start event occurs.

prefix_padding_ms integer
Used only for `server_vad` mode. Amount of audio to include before the VAD detected speech (in milliseconds). Defaults to 300ms.

silence_duration_ms integer
Used only for `server_vad` mode. Duration of silence to detect speech stop (in milliseconds). Defaults to 500ms. With shorter values the model will respond more quickly, but may jump in on short pauses from the user.

threshold number
Used only for `server_vad` mode. Activation threshold for VAD (0.0 to 1.0), this defaults to 0.5. A higher threshold will require louder audio to activate the model, and thus might perform better in noisy environments.

type string
Type of turn detection.

voice string
The voice the model uses to respond. Voice cannot be changed during the session once the model has responded with audio at least once. Current voice options are `alloy`, `ash`, `ballad`, `coral`, `echo`, `fable`, `onyx`, `nova`, `sage`, `shimmer`, and `verse`.

type string
The event type, must be `session.update`.

OBJECT session.update

```
{
    "event_id": "event_123",
    "type": "session.update",
    "session": {
        "modalities": ["text", "audio"],
        "instructions": "You are a helpful assistant.",
        "voice": "sage",
        "input_audio_format": "pcm16",
        "output_audio_format": "pcm16",
        "input_audio_transcription": {
            "model": "whisper-1"
        },
        "turn_detection": {
            "type": "server_vad",
            "threshold": 0.5,
```

```
            "prefix_padding_ms": 300,
            "silence_duration_ms": 500,
            "create_response": true
        },
        "tools": [
            {
                "type": "function",
                "name": "get_weather",
                "description": "Get the current weather...",
                "parameters": {
                    "type": "object",
                    "properties": {
                        "location": { "type": "string" }
                    },
                    "required": ["location"]
                }
            }
        ],
        "tool_choice": "auto",
        "temperature": 0.8,
        "max_response_output_tokens": "inf",
        "speed": 1.1,
        "tracing": "auto"
    }
}
```

input_audio_buffer.append

Send this event to append audio bytes to the input audio buffer. The audio buffer is temporary storage you can write to and later commit. In Server VAD mode, the audio buffer is used to detect speech and the server will decide when to commit. When Server VAD is disabled, you must commit the audio buffer manually.

The client may choose how much audio to place in each event up to a maximum of 15 MiB, for example streaming smaller chunks from the client may allow the VAD to be more responsive. Unlike made other client events, the server will not send a confirmation response to this event.

audio string

Base64-encoded audio bytes. This must be in the format specified by the `input_audio_format` field in the

session configuration.

event_id string

Optional client-generated ID used to identify this event.

type string

The event type, must be `input_audio_buffer.append` .

OBJECT input_audio_buffer.append

```
{
    "event_id": "event_456",
    "type": "input_audio_buffer.append",
    "audio": "Base64EncodedAudioData"
}
```

input_audio_buffer.commit

Send this event to commit the user input audio buffer, which will create a new user message item in the conversation. This event will produce an error if the input audio buffer is empty. When in Server VAD mode, the client does not need to send this event, the server will commit the audio buffer automatically.

Committing the input audio buffer will trigger input audio transcription (if enabled in session configuration), but it will not create a response from the model. The server will respond with an `input_audio_buffer.committed` event.

event_id string
Optional client-generated ID used to identify this event.

type string
The event type, must be `input_audio_buffer.commit`.

OBJECT input_audio_buffer.commit

```
{
    "event_id": "event_789",
    "type": "input_audio_buffer.commit"
}
```

input_audio_buffer.clear

Send this event to clear the audio bytes in the buffer. The server will respond with an `input_audio_buffer.cleared` event.

event_id string
Optional client-generated ID used to identify this event.

type string
The event type, must be `input_audio_buffer.clear`.

OBJECT input_audio_buffer.clear

```
{
```

```
2     "event_id": "event_012",
3     "type": "input_audio_buffer.clear"
4 }
```

conversation.item.create

Add a new Item to the Conversation's context, including messages, function calls, and function call responses. This event can be used both to populate a "history" of the conversation and to add new items mid-stream, but has the current limitation that it cannot populate assistant audio messages.

If successful, the server will respond with a `conversation.item.created` event, otherwise an `error` event will be sent.

`event_id` string
Optional client-generated ID used to identify this event.

`item` object
The item to add to the conversation.

^ Hide properties

`arguments` string
The arguments of the function call (for `function_call` items).

`call_id` string
The ID of the function call (for `function_call` and `function_call_output` items). If passed on a `function_call_output` item, the server will check that a `function_call` item with the same ID exists in the conversation history.

`content` array
The content of the message, applicable for `message` items.

Message items of role `system` support only `input_text` content

Message items of role `user` support `input_text` and `input_audio` content

Message items of role `assistant` support `text` content.

Hide properties

audio string
Base64-encoded audio bytes, used for `input_audio` content type.

id string
ID of a previous conversation item to reference (for `item_reference` content types in `response.create` events). These can reference both client and server created items.

text string
The text content, used for `input_text` and `text` content types.

transcript string
The transcript of the audio, used for `input_audio` content type.

type string
The content type (`input_text` , `input_audio` , `item_reference` , `text`).

id string
The unique ID of the item, this can be generated by the client to help manage server-side context, but is not required because the server will generate one if not provided.

name string
The name of the function being called (for `function_call` items).

object string
Identifier for the API object being returned - always `realtime.item` .

output string
The output of the function call (for `function_call_output` items).

role string
The role of the message sender (`user` , `assistant` , `system`), only applicable for `message` items.

status string
The status of the item (`completed` , `incomplete` , `in_progress`). These have no effect on the conversation, but are accepted for consistency with the `conversation.item.created` event.

type string
The type of the item (`message` , `function_call` , `function_call_output`).

previous_item_id string
The ID of the preceding item after which the new item will be inserted. If not set, the new item will be appended to the end of the conversation. If set to `root`, the new item will be added to the beginning of the conversation. If set to an existing ID, it allows an item to be inserted mid-conversation. If the ID cannot be found, an error will be returned and the item will not be added.

type string
The event type, must be `conversation.item.create`.

OBJECT conversation.item.create

```
{
    "event_id": "event_345",
    "type": "conversation.item.create",
    "previous_item_id": null,
    "item": {
        "id": "msg_001",
        "type": "message",
        "role": "user",
        "content": [
            {
                "type": "input_text",
                "text": "Hello, how are you?"
            }
        ]
    }
}
```

conversation.item.retrieve

Send this event when you want to retrieve the server's representation of a specific item in the conversation history. This is useful, for example, to inspect user audio after noise cancellation and VAD. The server will respond with a `conversation.item.retrieved` event, unless the item does not exist in the conversation history, in which case the server will respond with an error.

event_id string
Optional client-generated ID used to identify this event.

item_id string
The ID of the item to retrieve.

type string
The event type, must be `conversation.item.retrieve`.

OBJECT conversation.item.retrieve

```
{
    "event_id": "event_901",
    "type": "conversation.item.retrieve",
    "item_id": "msg_003"
}
```

conversation.item.truncate

Send this event to truncate a previous assistant message's audio. The server will produce audio faster than realtime, so this event is useful when the user interrupts to truncate audio that has already been sent to the client but not yet played. This will synchronize the server's understanding of the audio with the client's playback.

Truncating audio will delete the server-side text transcript to ensure there is not text in the context that hasn't been heard by the user.

If successful, the server will respond with a `conversation.item.truncated` event.

audio_end_ms integer
Inclusive duration up to which audio is truncated, in milliseconds. If the audio_end_ms is greater than the actual audio duration, the server will respond with an error.

content_index integer
The index of the content part to truncate. Set this to 0.

event_id string
Optional client-generated ID used to identify this event.

item_id string
The ID of the assistant message item to truncate. Only assistant message items can be truncated.

type string
The event type, must be `conversation.item.truncate`.

OBJECT conversation.item.truncate

```
{
    "event_id": "event_678",
    "type": "conversation.item.truncate",
    "item_id": "msg_002",
    "content_index": 0,
    "audio_end_ms": 1500
}
```

conversation.item.delete

Send this event when you want to remove any item from the conversation history. The server will respond with a `conversation.item.deleted` event, unless the item does not exist in the conversation history, in which case the server will respond with an error.

event_id string
Optional client-generated ID used to identify this event.

item_id string
The ID of the item to delete.

type string
The event type, must be `conversation.item.delete`.

OBJECT conversation.item.delete

```
{
    "event_id": "event_901",
    "type": "conversation.item.delete",
    "item_id": "msg_003"
}
```

response.create

This event instructs the server to create a Response, which means triggering model inference. When in Server VAD mode, the server will create Responses automatically.

A Response will include at least one Item, and may have two, in which case the second will be a function call. These Items will be appended to the conversation history.

The server will respond with a `response.created` event, events for Items and content created, and finally a `response.done` event to indicate the Response is complete.

The `response.create` event includes inference configuration like `instructions`, and `temperature`. These fields will override the Session's configuration for this Response only.

---

event_id string
Optional client-generated ID used to identify this event.

---

response object
Create a new Realtime response with these parameters

^ Hide properties

conversation string
Controls which conversation the response is added to. Currently supports `auto` and `none`, with `auto` as the default value. The `auto` value means that the contents of the response will be added to the default conversation. Set this to `none` to create an out-of-band response which will not add items to default conversation.

input array
Input items to include in the prompt for the model. Using this field creates a new context for this Response instead of using the default conversation. An empty array `[]` will clear the context for this Response. Note that this can include references to items from the default conversation.

^ Hide properties

arguments string
The arguments of the function call (for `function_call` items).

call_id string
The ID of the function call (for `function_call` and `function_call_output` items). If passed on a `function_call_output` item, the server will check that a `function_call` item with the same ID exists in the conversation history.

content array

The content of the message, applicable for `message` items.

Message items of role `system` support only `input_text` content

Message items of role `user` support `input_text` and `input_audio` content

Message items of role `assistant` support `text` content.

Hide properties

audio string
Base64-encoded audio bytes, used for `input_audio` content type.

id string
ID of a previous conversation item to reference (for `item_reference` content types in `response.create` events). These can reference both client and server created items.

text string
The text content, used for `input_text` and `text` content types.

transcript string
The transcript of the audio, used for `input_audio` content type.

type string
The content type (`input_text` , `input_audio` , `item_reference` , `text`).

id string
For an item of type (`message` | `function_call` | `function_call_output`) this field allows the client to assign the unique ID of the item. It is not required because the server will generate one if not provided.

For an item of type `item_reference` , this field is required and is a reference to any item that has previously existed in the conversation.

name string
The name of the function being called (for `function_call` items).

object string
Identifier for the API object being returned - always `realtime.item` .

output string
The output of the function call (for `function_call_output` items).

role string
The role of the message sender (`user` , `assistant` , `system`), only applicable for `message`

items.

status string
The status of the item (`completed` , `incomplete` , `in_progress`). These have no effect on the conversation, but are accepted for consistency with the `conversation.item.created` event.

type string
The type of the item (`message` , `function_call` , `function_call_output` , `item_reference`).

instructions string
The default system instructions (i.e. system message) prepended to model calls. This field allows the client to guide the model on desired responses. The model can be instructed on response content and format, (e.g. "be extremely succinct", "act friendly", "here are examples of good responses") and on audio behavior (e.g. "talk quickly", "inject emotion into your voice", "laugh frequently"). The instructions are not guaranteed to be followed by the model, but they provide guidance to the model on the desired behavior.

Note that the server sets default instructions which will be used if this field is not set and are visible in the `session.created` event at the start of the session.

max_response_output_tokens integer or "inf"
Maximum number of output tokens for a single assistant response, inclusive of tool calls. Provide an integer between 1 and 4096 to limit output tokens, or `inf` for the maximum available tokens for a given model. Defaults to `inf` .

metadata map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

modalities array
The set of modalities the model can respond with. To disable audio, set this to ["text"].

output_audio_format string
The format of output audio. Options are `pcm16` , `g711_ulaw` , or `g711_alaw` .

temperature number
Sampling temperature for the model, limited to [0.6, 1.2]. Defaults to 0.8.

tool_choice string
How the model chooses tools. Options are `auto` , `none` , `required` , or specify a function, like

`{"type": "function", "function": {"name": "my_function"}}`.

tools array
Tools (functions) available to the model.

Hide properties

description string
The description of the function, including guidance on when and how to call it, and guidance about what to tell the user when calling (if anything).

name string
The name of the function.

parameters object
Parameters of the function in JSON Schema.

type string
The type of the tool, i.e. `function`.

voice string
The voice the model uses to respond. Voice cannot be changed during the session once the model has responded with audio at least once. Current voice options are `alloy`, `ash`, `ballad`, `coral`, `echo`, `fable`, `onyx`, `nova`, `sage`, `shimmer`, and `verse`.

type string
The event type, must be `response.create`.

OBJECT response.create

```
{
    "event_id": "event_234",
    "type": "response.create",
    "response": {
        "modalities": ["text", "audio"],
        "instructions": "Please assist the user.",
        "voice": "sage",
        "output_audio_format": "pcm16",
        "tools": [
            {
                "type": "function",
                "name": "calculate_sum",
                "description": "Calculates the sum of two numbers.",

```

```
                "parameters": {
                    "type": "object",
                    "properties": {
                        "a": { "type": "number" },
                        "b": { "type": "number" }
                    },
                    "required": ["a", "b"]
                }
            }
        ],
        "tool_choice": "auto",
        "temperature": 0.8,
        "max_output_tokens": 1024
    }
}
```

response.cancel

Send this event to cancel an in-progress response. The server will respond with a `response.cancelled` event or an error if there is no response to cancel.

`event_id` string
Optional client-generated ID used to identify this event.

`response_id` string
A specific response ID to cancel - if not provided, will cancel an in-progress response in the default conversation.

`type` string
The event type, must be `response.cancel`.

OBJECT response.cancel

```
{
    "event_id": "event_567",
    "type": "response.cancel"
}
```

transcription_session.update

Send this event to update a transcription session.

event_id string
Optional client-generated ID used to identify this event.

session object
Realtime transcription session object configuration.

Hide properties

client_secret object
Configuration options for the generated client secret.

Hide properties

expires_at object
Configuration for the ephemeral token expiration.

Hide properties

anchor string
The anchor point for the ephemeral token expiration. Only `created_at` is currently supported.

seconds integer
The number of seconds from the anchor point to the expiration. Select a value between `10` and `7200`.

include array
The set of items to include in the transcription. Current available items are:

null.

input_audio_format string
The format of input audio. Options are `pcm16`, `g711_ulaw`, or `g711_alaw`. For `pcm16`, input audio must be 16-bit PCM at a 24kHz sample rate, single channel (mono), and little-endian byte order.

input_audio_noise_reduction object
Configuration for input audio noise reduction. This can be set to `null` to turn off. Noise reduction filters audio added to the input audio buffer before it is sent to VAD and the model. Filtering the audio can improve VAD and turn detection accuracy (reducing false positives) and model performance by improving perception of the input audio.

Hide properties

type string
Type of noise reduction. `near_field` is for close-talking microphones such as headphones, `far_field` is for far-field microphones such as laptop or conference room microphones.

input_audio_transcription object
Configuration for input audio transcription. The client can optionally set the language and prompt for transcription, these offer additional guidance to the transcription service.

Hide properties

language string
The language of the input audio. Supplying the input language in ISO-639-1 (e.g. `en`) format will improve accuracy and latency.

model string
The model to use for transcription, current options are `gpt-4o-transcribe` , `gpt-4o-mini-transcribe` , and `whisper-1` .

prompt string
An optional text to guide the model's style or continue a previous audio segment. For `whisper-1` , the prompt is a list of keywords. For `gpt-4o-transcribe` models, the prompt is a free text string, for example "expect words related to technology".

modalities
The set of modalities the model can respond with. To disable audio, set this to ["text"].

turn_detection object
Configuration for turn detection, ether Server VAD or Semantic VAD. This can be set to `null` to turn off, in which case the client must manually trigger model response. Server VAD means that the model will detect the start and end of speech based on audio volume and respond at the end of user speech. Semantic VAD is more advanced and uses a turn detection model (in conjuction with VAD) to semantically estimate whether the user has finished speaking, then dynamically sets a timeout based on this probability. For example, if user audio trails off with "uhhm", the model will score a low probability of turn end and wait longer for the user to continue speaking. This can be useful for more natural conversations, but may have a higher latency.

Hide properties

create_response boolean
Whether or not to automatically generate a response when a VAD stop event occurs. Not available for transcription sessions.

eagerness string

Used only for `semantic_vad` mode. The eagerness of the model to respond. `low` will wait longer for the user to continue speaking, `high` will respond more quickly. `auto` is the default and is equivalent to `medium`.

`interrupt_response` boolean
Whether or not to automatically interrupt any ongoing response with output to the default conversation (i.e. `conversation` of `auto`) when a VAD start event occurs. Not available for transcription sessions.

`prefix_padding_ms` integer
Used only for `server_vad` mode. Amount of audio to include before the VAD detected speech (in milliseconds). Defaults to 300ms.

`silence_duration_ms` integer
Used only for `server_vad` mode. Duration of silence to detect speech stop (in milliseconds). Defaults to 500ms. With shorter values the model will respond more quickly, but may jump in on short pauses from the user.

`threshold` number
Used only for `server_vad` mode. Activation threshold for VAD (0.0 to 1.0), this defaults to 0.5. A higher threshold will require louder audio to activate the model, and thus might perform better in noisy environments.

`type` string
Type of turn detection.

`type` string
The event type, must be `transcription_session.update`.

OBJECT transcription_session.update

```
{
  "type": "transcription_session.update",
  "session": {
    "input_audio_format": "pcm16",
    "input_audio_transcription": {
      "model": "gpt-4o-transcribe",
      "prompt": "",
      "language": ""
    },
    "turn_detection": {
      "type": "server_vad",
```

```
      "threshold": 0.5,
      "prefix_padding_ms": 300,
      "silence_duration_ms": 500,
      "create_response": true,
    },
    "input_audio_noise_reduction": {
      "type": "near_field"
    },
    "include": [
      "item.input_audio_transcription.logprobs",
    ]
  }
}
```

output_audio_buffer.clear

WebRTC Only: Emit to cut off the current audio response. This will trigger the server to stop generating audio and emit a `output_audio_buffer.cleared` event. This event should be preceded by a `response.cancel` client event to stop the generation of the current response. Learn more.

`event_id` string
The unique ID of the client event used for error handling.

`type` string
The event type, must be `output_audio_buffer.clear`.

OBJECT output_audio_buffer.clear

```
{
    "event_id": "optional_client_event_id",
    "type": "output_audio_buffer.clear"
}
```

Server events

These are events emitted from the OpenAI Realtime WebSocket server to the client.

error

Returned when an error occurs, which could be a client problem or a server problem. Most errors are recoverable and the session will stay open, we recommend to implementors to monitor and log error messages by default.

error object
Details of the error.

Hide properties

message string
A human-readable error message.

type string
The type of error (e.g., "invalid_request_error", "server_error").

code string or null
Error code, if any.

event_id string or null
The event_id of the client event that caused the error, if applicable.

param string or null
Parameter related to the error, if any.

event_id string
The unique ID of the server event.

type string
The event type, must be `error`.

OBJECT error

```
{
    "event_id": "event_890",
    "type": "error",
    "error": {

```

```
        "type": "invalid_request_error",
        "code": "invalid_event",
        "message": "The 'type' field is missing.",
        "param": null,
        "event_id": "event_567"
    }
}
```

session.created

Returned when a Session is created. Emitted automatically when a new connection is established as the first server event. This event will contain the default Session configuration.

event_id string
The unique ID of the server event.

session object
Realtime session object configuration.

Hide properties

id string
Unique identifier for the session that looks like `sess_1234567890abcdef`.

input_audio_format string
The format of input audio. Options are `pcm16`, `g711_ulaw`, or `g711_alaw`. For `pcm16`, input audio must be 16-bit PCM at a 24kHz sample rate, single channel (mono), and little-endian byte order.

input_audio_noise_reduction object
Configuration for input audio noise reduction. This can be set to `null` to turn off. Noise reduction filters audio added to the input audio buffer before it is sent to VAD and the model. Filtering the audio can improve VAD and turn detection accuracy (reducing false positives) and model performance by improving perception of the input audio.

Hide properties

type string
Type of noise reduction. `near_field` is for close-talking microphones such as headphones, `far_field` is for far-field microphones such as laptop or conference room microphones.

input_audio_transcription object

Configuration for input audio transcription, defaults to off and can be set to `null` to turn off once on. Input audio transcription is not native to the model, since the model consumes audio directly. Transcription runs asynchronously through the /audio/transcriptions endpoint and should be treated as guidance of input audio content rather than precisely what the model heard. The client can optionally set the language and prompt for transcription, these offer additional guidance to the transcription service.

Hide properties

language string

The language of the input audio. Supplying the input language in ISO-639-1 (e.g. `en`) format will improve accuracy and latency.

model string

The model to use for transcription, current options are `gpt-4o-transcribe` , `gpt-4o-mini-transcribe` , and `whisper-1` .

prompt string

An optional text to guide the model's style or continue a previous audio segment. For `whisper-1` , the prompt is a list of keywords. For `gpt-4o-transcribe` models, the prompt is a free text string, for example "expect words related to technology".

instructions string

The default system instructions (i.e. system message) prepended to model calls. This field allows the client to guide the model on desired responses. The model can be instructed on response content and format, (e.g. "be extremely succinct", "act friendly", "here are examples of good responses") and on audio behavior (e.g. "talk quickly", "inject emotion into your voice", "laugh frequently"). The instructions are not guaranteed to be followed by the model, but they provide guidance to the model on the desired behavior.

Note that the server sets default instructions which will be used if this field is not set and are visible in the `session.created` event at the start of the session.

max_response_output_tokens integer or "inf"

Maximum number of output tokens for a single assistant response, inclusive of tool calls. Provide an integer between 1 and 4096 to limit output tokens, or `inf` for the maximum available tokens for a given model. Defaults to `inf` .

modalities

The set of modalities the model can respond with. To disable audio, set this to ["text"].

model string

The Realtime model used for this session.

output_audio_format string
The format of output audio. Options are `pcm16`, `g711_ulaw`, or `g711_alaw`. For `pcm16`, output audio is sampled at a rate of 24kHz.

speed number
The speed of the model's spoken response. 1.0 is the default speed. 0.25 is the minimum speed. 1.5 is the maximum speed. This value can only be changed in between model turns, not while a response is in progress.

temperature number
Sampling temperature for the model, limited to [0.6, 1.2]. For audio models a temperature of 0.8 is highly recommended for best performance.

tool_choice string
How the model chooses tools. Options are `auto`, `none`, `required`, or specify a function.

tools array
Tools (functions) available to the model.

Hide properties

description string
The description of the function, including guidance on when and how to call it, and guidance about what to tell the user when calling (if anything).

name string
The name of the function.

parameters object
Parameters of the function in JSON Schema.

type string
The type of the tool, i.e. `function`.

tracing "auto" or object
Configuration options for tracing. Set to null to disable tracing. Once tracing is enabled for a session, the configuration cannot be modified.

`auto` will create a trace for the session with default values for the workflow name, group id, and metadata.

Hide possible types

string

Default tracing mode for the session.

Tracing Configuration object
Granular configuration for tracing.

Hide properties

group_id string
The group id to attach to this trace to enable filtering and grouping in the traces dashboard.

metadata object
The arbitrary metadata to attach to this trace to enable filtering in the traces dashboard.

workflow_name string
The name of the workflow to attach to this trace. This is used to name the trace in the traces dashboard.

turn_detection object
Configuration for turn detection, ether Server VAD or Semantic VAD. This can be set to `null` to turn off, in which case the client must manually trigger model response. Server VAD means that the model will detect the start and end of speech based on audio volume and respond at the end of user speech. Semantic VAD is more advanced and uses a turn detection model (in conjuction with VAD) to semantically estimate whether the user has finished speaking, then dynamically sets a timeout based on this probability. For example, if user audio trails off with "uhhm", the model will score a low probability of turn end and wait longer for the user to continue speaking. This can be useful for more natural conversations, but may have a higher latency.

Hide properties

create_response boolean
Whether or not to automatically generate a response when a VAD stop event occurs.

eagerness string
Used only for `semantic_vad` mode. The eagerness of the model to respond. `low` will wait longer for the user to continue speaking, `high` will respond more quickly. `auto` is the default and is equivalent to `medium`.

interrupt_response boolean
Whether or not to automatically interrupt any ongoing response with output to the default conversation (i.e. `conversation` of `auto`) when a VAD start event occurs.

prefix_padding_ms integer
Used only for `server_vad` mode. Amount of audio to include before the VAD detected speech (in milliseconds). Defaults to 300ms.

silence_duration_ms integer
Used only for `server_vad` mode. Duration of silence to detect speech stop (in milliseconds). Defaults to 500ms. With shorter values the model will respond more quickly, but may jump in on short pauses from the user.

threshold number
Used only for `server_vad` mode. Activation threshold for VAD (0.0 to 1.0), this defaults to 0.5. A higher threshold will require louder audio to activate the model, and thus might perform better in noisy environments.

type string
Type of turn detection.

voice string
The voice the model uses to respond. Voice cannot be changed during the session once the model has responded with audio at least once. Current voice options are `alloy`, `ash`, `ballad`, `coral`, `echo`, `fable`, `onyx`, `nova`, `sage`, `shimmer`, and `verse`.

type string
The event type, must be `session.created`.

OBJECT session.created

```
{
    "event_id": "event_1234",
    "type": "session.created",
    "session": {
        "id": "sess_001",
        "object": "realtime.session",
        "model": "gpt-4o-realtime-preview",
        "modalities": ["text", "audio"],
        "instructions": "...model instructions here...",
        "voice": "sage",
        "input_audio_format": "pcm16",
        "output_audio_format": "pcm16",
        "input_audio_transcription": null,
        "turn_detection": {
            "type": "server_vad",
```

```
            "threshold": 0.5,
            "prefix_padding_ms": 300,
            "silence_duration_ms": 200
        },
        "tools": [],
        "tool_choice": "auto",
        "temperature": 0.8,
        "max_response_output_tokens": "inf",
        "speed": 1.1,
        "tracing": "auto"
    }
}
```

session.updated

Returned when a session is updated with a `session.update` event, unless there is an error.

`event_id` string
The unique ID of the server event.

`session` object
Realtime session object configuration.

Hide properties

`id` string
Unique identifier for the session that looks like `sess_1234567890abcdef`.

`input_audio_format` string
The format of input audio. Options are `pcm16`, `g711_ulaw`, or `g711_alaw`. For `pcm16`, input audio must be 16-bit PCM at a 24kHz sample rate, single channel (mono), and little-endian byte order.

`input_audio_noise_reduction` object
Configuration for input audio noise reduction. This can be set to `null` to turn off. Noise reduction filters audio added to the input audio buffer before it is sent to VAD and the model. Filtering the audio can improve VAD and turn detection accuracy (reducing false positives) and model performance by improving perception of the input audio.

Hide properties

type string
Type of noise reduction. `near_field` is for close-talking microphones such as headphones, `far_field` is for far-field microphones such as laptop or conference room microphones.

input_audio_transcription object
Configuration for input audio transcription, defaults to off and can be set to `null` to turn off once on. Input audio transcription is not native to the model, since the model consumes audio directly. Transcription runs asynchronously through the /audio/transcriptions endpoint and should be treated as guidance of input audio content rather than precisely what the model heard. The client can optionally set the language and prompt for transcription, these offer additional guidance to the transcription service.

^ Hide properties

language string
The language of the input audio. Supplying the input language in ISO-639-1 (e.g. `en`) format will improve accuracy and latency.

model string
The model to use for transcription, current options are `gpt-4o-transcribe` , `gpt-4o-mini-transcribe` , and `whisper-1` .

prompt string
An optional text to guide the model's style or continue a previous audio segment. For `whisper-1` , the prompt is a list of keywords. For `gpt-4o-transcribe` models, the prompt is a free text string, for example "expect words related to technology".

instructions string
The default system instructions (i.e. system message) prepended to model calls. This field allows the client to guide the model on desired responses. The model can be instructed on response content and format, (e.g. "be extremely succinct", "act friendly", "here are examples of good responses") and on audio behavior (e.g. "talk quickly", "inject emotion into your voice", "laugh frequently"). The instructions are not guaranteed to be followed by the model, but they provide guidance to the model on the desired behavior.

Note that the server sets default instructions which will be used if this field is not set and are visible in the `session.created` event at the start of the session.

max_response_output_tokens integer or "inf"
Maximum number of output tokens for a single assistant response, inclusive of tool calls. Provide an integer between 1 and 4096 to limit output tokens, or `inf` for the maximum available tokens for a given model. Defaults to `inf` .

modalities

The set of modalities the model can respond with. To disable audio, set this to ["text"].

model string
The Realtime model used for this session.

output_audio_format string
The format of output audio. Options are `pcm16`, `g711_ulaw`, or `g711_alaw`. For `pcm16`, output audio is sampled at a rate of 24kHz.

speed number
The speed of the model's spoken response. 1.0 is the default speed. 0.25 is the minimum speed. 1.5 is the maximum speed. This value can only be changed in between model turns, not while a response is in progress.

temperature number
Sampling temperature for the model, limited to [0.6, 1.2]. For audio models a temperature of 0.8 is highly recommended for best performance.

tool_choice string
How the model chooses tools. Options are `auto`, `none`, `required`, or specify a function.

tools array
Tools (functions) available to the model.

Hide properties

description string
The description of the function, including guidance on when and how to call it, and guidance about what to tell the user when calling (if anything).

name string
The name of the function.

parameters object
Parameters of the function in JSON Schema.

type string
The type of the tool, i.e. `function`.

tracing "auto" or object
Configuration options for tracing. Set to null to disable tracing. Once tracing is enabled for a session, the configuration cannot be modified.

`auto` will create a trace for the session with default values for the workflow name, group id, and

metadata.

Hide possible types

string
Default tracing mode for the session.

Tracing Configuration object
Granular configuration for tracing.

Hide properties

group_id string
The group id to attach to this trace to enable filtering and grouping in the traces dashboard.

metadata object
The arbitrary metadata to attach to this trace to enable filtering in the traces dashboard.

workflow_name string
The name of the workflow to attach to this trace. This is used to name the trace in the traces dashboard.

turn_detection object
Configuration for turn detection, ether Server VAD or Semantic VAD. This can be set to `null` to turn off, in which case the client must manually trigger model response. Server VAD means that the model will detect the start and end of speech based on audio volume and respond at the end of user speech. Semantic VAD is more advanced and uses a turn detection model (in conjuction with VAD) to semantically estimate whether the user has finished speaking, then dynamically sets a timeout based on this probability. For example, if user audio trails off with "uhhm", the model will score a low probability of turn end and wait longer for the user to continue speaking. This can be useful for more natural conversations, but may have a higher latency.

Hide properties

create_response boolean
Whether or not to automatically generate a response when a VAD stop event occurs.

eagerness string
Used only for `semantic_vad` mode. The eagerness of the model to respond. `low` will wait longer for the user to continue speaking, `high` will respond more quickly. `auto` is the default and is equivalent to `medium`.

interrupt_response boolean
Whether or not to automatically interrupt any ongoing response with output to the default conversation (i.e. `conversation` of `auto`) when a VAD start event occurs.

prefix_padding_ms integer
Used only for `server_vad` mode. Amount of audio to include before the VAD detected speech (in milliseconds). Defaults to 300ms.

silence_duration_ms integer
Used only for `server_vad` mode. Duration of silence to detect speech stop (in milliseconds). Defaults to 500ms. With shorter values the model will respond more quickly, but may jump in on short pauses from the user.

threshold number
Used only for `server_vad` mode. Activation threshold for VAD (0.0 to 1.0), this defaults to 0.5. A higher threshold will require louder audio to activate the model, and thus might perform better in noisy environments.

type string
Type of turn detection.

voice string
The voice the model uses to respond. Voice cannot be changed during the session once the model has responded with audio at least once. Current voice options are `alloy`, `ash`, `ballad`, `coral`, `echo`, `fable`, `onyx`, `nova`, `sage`, `shimmer`, and `verse`.

type string
The event type, must be `session.updated`.

OBJECT session.updated

```
{
    "event_id": "event_5678",
    "type": "session.updated",
    "session": {
        "id": "sess_001",
        "object": "realtime.session",
        "model": "gpt-4o-realtime-preview",
        "modalities": ["text"],
        "instructions": "New instructions",
        "voice": "sage",
        "input_audio_format": "pcm16",
```

```
        "output_audio_format": "pcm16",
        "input_audio_transcription": {
            "model": "whisper-1"
        },
        "turn_detection": null,
        "tools": [],
        "tool_choice": "none",
        "temperature": 0.7,
        "max_response_output_tokens": 200,
        "speed": 1.1,
        "tracing": "auto"
    }
}
```

conversation.created

Returned when a conversation is created. Emitted right after session creation.

conversation object
The conversation resource.

Hide properties

id string
The unique ID of the conversation.

object string
The object type, must be `realtime.conversation`.

event_id string
The unique ID of the server event.

type string
The event type, must be `conversation.created`.

OBJECT conversation.created

```
{
    "event_id": "event_9101",
```

```
    "type": "conversation.created",
    "conversation": {
        "id": "conv_001",
        "object": "realtime.conversation"
    }
}
```

conversation.item.created

Returned when a conversation item is created. There are several scenarios that produce this event:

- The server is generating a Response, which if successful will produce either one or two Items, which will be of type `message` (role `assistant`) or type `function_call`.
- The input audio buffer has been committed, either by the client or the server (in `server_vad` mode). The server will take the content of the input audio buffer and add it to a new user message Item.
- The client has sent a `conversation.item.create` event to add a new Item to the Conversation.

event_id string
The unique ID of the server event.

item object
The item to add to the conversation.

Hide properties

arguments string
The arguments of the function call (for `function_call` items).

call_id string
The ID of the function call (for `function_call` and `function_call_output` items). If passed on a `function_call_output` item, the server will check that a `function_call` item with the same ID exists in the conversation history.

content array
The content of the message, applicable for `message` items.

Message items of role `system` support only `input_text` content

Message items of role `user` support `input_text` and `input_audio` content

Message items of role `assistant` support `text` content.

Hide properties

`audio` string
Base64-encoded audio bytes, used for `input_audio` content type.

`id` string
ID of a previous conversation item to reference (for `item_reference` content types in `response.create` events). These can reference both client and server created items.

`text` string
The text content, used for `input_text` and `text` content types.

`transcript` string
The transcript of the audio, used for `input_audio` content type.

`type` string
The content type (`input_text` , `input_audio` , `item_reference` , `text`).

`id` string
The unique ID of the item, this can be generated by the client to help manage server-side context, but is not required because the server will generate one if not provided.

`name` string
The name of the function being called (for `function_call` items).

`object` string
Identifier for the API object being returned - always `realtime.item` .

`output` string
The output of the function call (for `function_call_output` items).

`role` string
The role of the message sender (`user` , `assistant` , `system`), only applicable for `message` items.

`status` string
The status of the item (`completed` , `incomplete` , `in_progress`). These have no effect on the conversation, but are accepted for consistency with the `conversation.item.created` event.

type string
The type of the item (`message` , `function_call` , `function_call_output`).

previous_item_id string or null
The ID of the preceding item in the Conversation context, allows the client to understand the order of the conversation. Can be `null` if the item has no predecessor.

type string
The event type, must be `conversation.item.created` .

OBJECT conversation.item.created

```
{
    "event_id": "event_1920",
    "type": "conversation.item.created",
    "previous_item_id": "msg_002",
    "item": {
        "id": "msg_003",
        "object": "realtime.item",
        "type": "message",
        "status": "completed",
        "role": "user",
        "content": []
    }
}
```

conversation.item.retrieved

Returned when a conversation item is retrieved with `conversation.item.retrieve` .

event_id string
The unique ID of the server event.

item object
The item to add to the conversation.

Hide properties

arguments string
The arguments of the function call (for `function_call` items).

call_id string
The ID of the function call (for `function_call` and `function_call_output` items). If passed on a `function_call_output` item, the server will check that a `function_call` item with the same ID exists in the conversation history.

content array
The content of the message, applicable for `message` items.

Message items of role `system` support only `input_text` content

Message items of role `user` support `input_text` and `input_audio` content

Message items of role `assistant` support `text` content.

^ Hide properties

audio string
Base64-encoded audio bytes, used for `input_audio` content type.

id string
ID of a previous conversation item to reference (for `item_reference` content types in `response.create` events). These can reference both client and server created items.

text string
The text content, used for `input_text` and `text` content types.

transcript string
The transcript of the audio, used for `input_audio` content type.

type string
The content type (`input_text` , `input_audio` , `item_reference` , `text`).

id string
The unique ID of the item, this can be generated by the client to help manage server-side context, but is not required because the server will generate one if not provided.

name string
The name of the function being called (for `function_call` items).

object string
Identifier for the API object being returned - always `realtime.item` .

output string
The output of the function call (for `function_call_output` items).

role string
The role of the message sender (`user` , `assistant` , `system`), only applicable for `message` items.

status string
The status of the item (`completed` , `incomplete` , `in_progress`). These have no effect on the conversation, but are accepted for consistency with the `conversation.item.created` event.

type string
The type of the item (`message` , `function_call` , `function_call_output`).

type string
The event type, must be `conversation.item.retrieved` .

OBJECT conversation.item.retrieved

```
{
    "event_id": "event_1920",
    "type": "conversation.item.created",
    "previous_item_id": "msg_002",
    "item": {
        "id": "msg_003",
        "object": "realtime.item",
        "type": "message",
        "status": "completed",
        "role": "user",
        "content": [
            {
                "type": "input_audio",
                "transcript": "hello how are you",
                "audio": "base64encodedaudio=="
            }
        ]
    }
}
```

conversation.item.input_audio_transcription.completed

This event is the output of audio transcription for user audio written to the user audio buffer. Transcription begins when the input audio buffer is committed by the client or server (in `server_vad` mode). Transcription runs asynchronously with Response creation, so this event may come before or after the Response events.

Realtime API models accept audio natively, and thus input transcription is a separate process run on a separate ASR (Automatic Speech Recognition) model. The transcript may diverge somewhat from the model's interpretation, and should be treated as a rough guide.

`content_index` integer
The index of the content part containing the audio.

`event_id` string
The unique ID of the server event.

`item_id` string
The ID of the user message item containing the audio.

`logprobs` array or null
The log probabilities of the transcription.

^ Hide properties

`bytes` array
The bytes that were used to generate the log probability.

`logprob` number
The log probability of the token.

`token` string
The token that was used to generate the log probability.

`transcript` string
The transcribed text.

type string
The event type, must be `conversation.item.input_audio_transcription.completed`.

usage object
Usage statistics for the transcription.

Hide possible types

Token Usage object
Usage statistics for models billed by token usage.

Hide properties

input_tokens integer
Number of input tokens billed for this request.

output_tokens integer
Number of output tokens generated.

total_tokens integer
Total number of tokens used (input + output).

type string
The type of the usage object. Always `tokens` for this variant.

input_token_details object
Details about the input tokens billed for this request.

Hide properties

audio_tokens integer
Number of audio tokens billed for this request.

text_tokens integer
Number of text tokens billed for this request.

Duration Usage object
Usage statistics for models billed by audio input duration.

Hide properties

seconds number
Duration of the input audio in seconds.

type string

The type of the usage object. Always `duration` for this variant.

OBJECT conversation.item.input_audio_transcription.completed

```
{
    "event_id": "event_2122",
    "type": "conversation.item.input_audio_transcription.completed",
    "item_id": "msg_003",
    "content_index": 0,
    "transcript": "Hello, how are you?",
    "usage": {
      "type": "tokens",
      "total_tokens": 48,
      "input_tokens": 38,
      "input_token_details": {
        "text_tokens": 10,
        "audio_tokens": 28,
      },
      "output_tokens": 10,
    }
}
```

conversation.item.input_audio_transcription.delta

Returned when the text value of an input audio transcription content part is updated.

content_index integer
The index of the content part in the item's content array.

delta string
The text delta.

event_id string
The unique ID of the server event.

item_id string
The ID of the item.

logprobs array or null

The log probabilities of the transcription.

Hide properties

bytes array

The bytes that were used to generate the log probability.

logprob number

The log probability of the token.

token string

The token that was used to generate the log probability.

type string

The event type, must be `conversation.item.input_audio_transcription.delta`.

OBJECT conversation.item.input_audio_transcription.delta

```
{
  "type": "conversation.item.input_audio_transcription.delta",
  "event_id": "event_001",
  "item_id": "item_001",
  "content_index": 0,
  "delta": "Hello"
}
```

conversation.item.input_audio_transcription.failed

Returned when input audio transcription is configured, and a transcription request for a user message failed. These events are separate from other `error` events so that the client can identify the related Item.

content_index integer

The index of the content part containing the audio.

error object

Details of the transcription error.

Hide properties

code string
Error code, if any.

message string
A human-readable error message.

param string
Parameter related to the error, if any.

type string
The type of error.

event_id string
The unique ID of the server event.

item_id string
The ID of the user message item.

type string
The event type, must be `conversation.item.input_audio_transcription.failed`.

OBJECT conversation.item.input_audio_transcription.failed

```
{
    "event_id": "event_2324",
    "type": "conversation.item.input_audio_transcription.failed",
    "item_id": "msg_003",
    "content_index": 0,
    "error": {
        "type": "transcription_error",
        "code": "audio_unintelligible",
        "message": "The audio could not be transcribed.",
        "param": null
    }
}
```

conversation.item.truncated

Returned when an earlier assistant audio message item is truncated by the client with a `conversation.item.truncate` event. This event is used to synchronize the server's understanding of the audio with the client's playback.

This action will truncate the audio and remove the server-side text transcript to ensure there is no text in the context that hasn't been heard by the user.

`audio_end_ms` integer
The duration up to which the audio was truncated, in milliseconds.

`content_index` integer
The index of the content part that was truncated.

`event_id` string
The unique ID of the server event.

`item_id` string
The ID of the assistant message item that was truncated.

`type` string
The event type, must be `conversation.item.truncated`.

OBJECT conversation.item.truncated

```
{
    "event_id": "event_2526",
    "type": "conversation.item.truncated",
    "item_id": "msg_004",
    "content_index": 0,
    "audio_end_ms": 1500
}
```

conversation.item.deleted

Returned when an item in the conversation is deleted by the client with a `conversation.item.delete` event. This event is used to synchronize the server's understanding of the conversation history with the client's view.

`event_id` string

The unique ID of the server event.

item_id string
The ID of the item that was deleted.

type string
The event type, must be `conversation.item.deleted`.

OBJECT conversation.item.deleted

```
{
    "event_id": "event_2728",
    "type": "conversation.item.deleted",
    "item_id": "msg_005"
}
```

input_audio_buffer.committed

Returned when an input audio buffer is committed, either by the client or automatically in server VAD mode. The `item_id` property is the ID of the user message item that will be created, thus a `conversation.item.created` event will also be sent to the client.

event_id string
The unique ID of the server event.

item_id string
The ID of the user message item that will be created.

previous_item_id string or null
The ID of the preceding item after which the new item will be inserted. Can be `null` if the item has no predecessor.

type string
The event type, must be `input_audio_buffer.committed`.

OBJECT input_audio_buffer.committed

```
{
    "event_id": "event_1121",
    "type": "input_audio_buffer.committed",
    "previous_item_id": "msg_001",
    "item_id": "msg_002"
}
```

input_audio_buffer.cleared

Returned when the input audio buffer is cleared by the client with a `input_audio_buffer.clear` event.

event_id string
The unique ID of the server event.

type string
The event type, must be `input_audio_buffer.cleared`.

OBJECT input_audio_buffer.cleared

```
{
    "event_id": "event_1314",
    "type": "input_audio_buffer.cleared"
}
```

input_audio_buffer.speech_started

Sent by the server when in `server_vad` mode to indicate that speech has been detected in the audio buffer. This can happen any time audio is added to the buffer (unless speech is already detected). The client may want to use this event to interrupt audio playback or provide visual feedback to the user.

The client should expect to receive a `input_audio_buffer.speech_stopped` event when speech stops. The `item_id` property is the ID of the user message item that will be created when speech stops and will also be included in the `input_audio_buffer.speech_stopped` event (unless the client manually commits the audio buffer during VAD activation).

audio_start_ms integer
Milliseconds from the start of all audio written to the buffer during the session when speech was first detected. This will correspond to the beginning of audio sent to the model, and thus includes the `prefix_padding_ms` configured in the Session.

event_id string
The unique ID of the server event.

item_id string
The ID of the user message item that will be created when speech stops.

type string
The event type, must be `input_audio_buffer.speech_started`.

OBJECT input_audio_buffer.speech_started

```
{
    "event_id": "event_1516",
    "type": "input_audio_buffer.speech_started",
    "audio_start_ms": 1000,
    "item_id": "msg_003"
}
```

input_audio_buffer.speech_stopped

Returned in `server_vad` mode when the server detects the end of speech in the audio buffer. The server will also send an `conversation.item.created` event with the user message item that is created from the audio buffer.

audio_end_ms integer
Milliseconds since the session started when speech stopped. This will correspond to the end of audio sent to the model, and thus includes the `min_silence_duration_ms` configured in the Session.

event_id string
The unique ID of the server event.

item_id string
The ID of the user message item that will be created.

type string
The event type, must be `input_audio_buffer.speech_stopped`.

OBJECT input_audio_buffer.speech_stopped

```
{
    "event_id": "event_1718",
    "type": "input_audio_buffer.speech_stopped",
    "audio_end_ms": 2000,
    "item_id": "msg_003"
}
```

response.created

Returned when a new Response is created. The first event of response creation, where the response is in an initial state of `in_progress`.

event_id string
The unique ID of the server event.

response object
The response resource.

Hide properties

conversation_id string
Which conversation the response is added to, determined by the `conversation` field in the `response.create` event. If `auto`, the response will be added to the default conversation and the value of `conversation_id` will be an id like `conv_1234`. If `none`, the response will not be added to any conversation and the value of `conversation_id` will be `null`. If responses are being triggered by server VAD, the response will be added to the default conversation, thus the `conversation_id` will be an id like `conv_1234`.

id string
The unique ID of the response.

max_output_tokens integer or "inf"
Maximum number of output tokens for a single assistant response, inclusive of tool calls, that was used

in this response.

metadata map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

modalities array
The set of modalities the model used to respond. If there are multiple modalities, the model will pick one, for example if `modalities` is `["text", "audio"]`, the model could be responding in either text or audio.

object string
The object type, must be `realtime.response`.

output array
The list of output items generated by the response.

Hide properties

arguments string
The arguments of the function call (for `function_call` items).

call_id string
The ID of the function call (for `function_call` and `function_call_output` items). If passed on a `function_call_output` item, the server will check that a `function_call` item with the same ID exists in the conversation history.

content array
The content of the message, applicable for `message` items.

- Message items of role `system` support only `input_text` content
- Message items of role `user` support `input_text` and `input_audio` content
- Message items of role `assistant` support `text` content.

Hide properties

audio string
Base64-encoded audio bytes, used for `input_audio` content type.

id string
ID of a previous conversation item to reference (for `item_reference` content types in `response.create` events). These can reference both client and server created items.

text string
The text content, used for `input_text` and `text` content types.

transcript string
The transcript of the audio, used for `input_audio` content type.

type string
The content type (`input_text` , `input_audio` , `item_reference` , `text`).

id string
The unique ID of the item, this can be generated by the client to help manage server-side context, but is not required because the server will generate one if not provided.

name string
The name of the function being called (for `function_call` items).

object string
Identifier for the API object being returned - always `realtime.item` .

output string
The output of the function call (for `function_call_output` items).

role string
The role of the message sender (`user` , `assistant` , `system`), only applicable for `message` items.

status string
The status of the item (`completed` , `incomplete` , `in_progress`). These have no effect on the conversation, but are accepted for consistency with the `conversation.item.created` event.

type string
The type of the item (`message` , `function_call` , `function_call_output`).

output_audio_format string
The format of output audio. Options are `pcm16` , `g711_ulaw` , or `g711_alaw` .

status string

The final status of the response (`completed` , `cancelled` , `failed` , or `incomplete` , `in_progress`).

`status_details` object
Additional details about the status.

Hide properties

`error` object
A description of the error that caused the response to fail, populated when the `status` is `failed` .

Hide properties

`code` string
Error code, if any.

`type` string
The type of error.

`reason` string
The reason the Response did not complete. For a `cancelled` Response, one of `turn_detected` (the server VAD detected a new start of speech) or `client_cancelled` (the client sent a cancel event). For an `incomplete` Response, one of `max_output_tokens` or `content_filter` (the server-side safety filter activated and cut off the response).

`type` string
The type of error that caused the response to fail, corresponding with the `status` field (`completed` , `cancelled` , `incomplete` , `failed`).

`temperature` number
Sampling temperature for the model, limited to [0.6, 1.2]. Defaults to 0.8.

`usage` object
Usage statistics for the Response, this will correspond to billing. A Realtime API session will maintain a conversation context and append new Items to the Conversation, thus output from previous turns (text and audio tokens) will become the input for later turns.

Hide properties

`input_token_details` object
Details about the input tokens used in the Response.

Hide properties

`audio_tokens` integer
The number of audio tokens used in the Response.

cached_tokens integer
The number of cached tokens used in the Response.

text_tokens integer
The number of text tokens used in the Response.

input_tokens integer
The number of input tokens used in the Response, including text and audio tokens.

output_token_details object
Details about the output tokens used in the Response.

Hide properties

audio_tokens integer
The number of audio tokens used in the Response.

text_tokens integer
The number of text tokens used in the Response.

output_tokens integer
The number of output tokens sent in the Response, including text and audio tokens.

total_tokens integer
The total number of tokens in the Response including input and output text and audio tokens.

voice string
The voice the model used to respond. Current voice options are `alloy`, `ash`, `ballad`, `coral`, `echo`, `fable`, `onyx`, `nova`, `sage`, `shimmer`, and `verse`.

type string
The event type, must be `response.created`.

OBJECT response.created

```
{
    "event_id": "event_2930",
    "type": "response.created",
    "response": {
        "id": "resp_001",
        "object": "realtime.response",
```

```
        "status": "in_progress",
        "status_details": null,
        "output": [],
        "usage": null
    }
}
```

response.done

Returned when a Response is done streaming. Always emitted, no matter the final state. The Response object included in the `response.done` event will include all output Items in the Response but will omit the raw audio data.

event_id string
The unique ID of the server event.

response object
The response resource.

Hide properties

conversation_id string
Which conversation the response is added to, determined by the `conversation` field in the `response.create` event. If `auto`, the response will be added to the default conversation and the value of `conversation_id` will be an id like `conv_1234`. If `none`, the response will not be added to any conversation and the value of `conversation_id` will be `null`. If responses are being triggered by server VAD, the response will be added to the default conversation, thus the `conversation_id` will be an id like `conv_1234`.

id string
The unique ID of the response.

max_output_tokens integer or "inf"
Maximum number of output tokens for a single assistant response, inclusive of tool calls, that was used in this response.

metadata map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

modalities array
The set of modalities the model used to respond. If there are multiple modalities, the model will pick one, for example if `modalities` is `["text", "audio"]`, the model could be responding in either text or audio.

object string
The object type, must be `realtime.response`.

output array
The list of output items generated by the response.

Hide properties

arguments string
The arguments of the function call (for `function_call` items).

call_id string
The ID of the function call (for `function_call` and `function_call_output` items). If passed on a `function_call_output` item, the server will check that a `function_call` item with the same ID exists in the conversation history.

content array
The content of the message, applicable for `message` items.

Message items of role `system` support only `input_text` content

Message items of role `user` support `input_text` and `input_audio` content

Message items of role `assistant` support `text` content.

Hide properties

audio string
Base64-encoded audio bytes, used for `input_audio` content type.

id string
ID of a previous conversation item to reference (for `item_reference` content types in `response.create` events). These can reference both client and server created items.

text string
The text content, used for `input_text` and `text` content types.

transcript string
The transcript of the audio, used for `input_audio` content type.

type string
The content type (`input_text` , `input_audio` , `item_reference` , `text`).

id string
The unique ID of the item, this can be generated by the client to help manage server-side context, but is not required because the server will generate one if not provided.

name string
The name of the function being called (for `function_call` items).

object string
Identifier for the API object being returned - always `realtime.item` .

output string
The output of the function call (for `function_call_output` items).

role string
The role of the message sender (`user` , `assistant` , `system`), only applicable for `message` items.

status string
The status of the item (`completed` , `incomplete` , `in_progress`). These have no effect on the conversation, but are accepted for consistency with the `conversation.item.created` event.

type string
The type of the item (`message` , `function_call` , `function_call_output`).

output_audio_format string
The format of output audio. Options are `pcm16` , `g711_ulaw` , or `g711_alaw` .

status string
The final status of the response (`completed` , `cancelled` , `failed` , or `incomplete` , `in_progress`).

status_details object
Additional details about the status.

Hide properties

error object
A description of the error that caused the response to fail, populated when the `status` is `failed`.

Hide properties

code string
Error code, if any.

type string
The type of error.

reason string
The reason the Response did not complete. For a `cancelled` Response, one of `turn_detected` (the server VAD detected a new start of speech) or `client_cancelled` (the client sent a cancel event). For an `incomplete` Response, one of `max_output_tokens` or `content_filter` (the server-side safety filter activated and cut off the response).

type string
The type of error that caused the response to fail, corresponding with the `status` field (`completed` , `cancelled` , `incomplete` , `failed`).

temperature number
Sampling temperature for the model, limited to [0.6, 1.2]. Defaults to 0.8.

usage object
Usage statistics for the Response, this will correspond to billing. A Realtime API session will maintain a conversation context and append new Items to the Conversation, thus output from previous turns (text and audio tokens) will become the input for later turns.

Hide properties

input_token_details object
Details about the input tokens used in the Response.

Hide properties

audio_tokens integer
The number of audio tokens used in the Response.

cached_tokens integer
The number of cached tokens used in the Response.

text_tokens integer

The number of text tokens used in the Response.

input_tokens integer
The number of input tokens used in the Response, including text and audio tokens.

output_token_details object
Details about the output tokens used in the Response.

Hide properties

audio_tokens integer
The number of audio tokens used in the Response.

text_tokens integer
The number of text tokens used in the Response.

output_tokens integer
The number of output tokens sent in the Response, including text and audio tokens.

total_tokens integer
The total number of tokens in the Response including input and output text and audio tokens.

voice string
The voice the model used to respond. Current voice options are `alloy`, `ash`, `ballad`, `coral`, `echo`, `fable`, `onyx`, `nova`, `sage`, `shimmer`, and `verse`.

type string
The event type, must be `response.done`.

OBJECT response.done

```
{
    "event_id": "event_3132",
    "type": "response.done",
    "response": {
        "id": "resp_001",
        "object": "realtime.response",
        "status": "completed",
        "status_details": null,
        "output": [
            {
                "id": "msg_006",
```

```
                "object": "realtime.item",
                "type": "message",
                "status": "completed",
                "role": "assistant",
                "content": [
                    {
                        "type": "text",
                        "text": "Sure, how can I assist you today?"
                    }
                ]
            }
        ],
        "usage": {
            "total_tokens":275,
            "input_tokens":127,
            "output_tokens":148,
            "input_token_details": {
                "cached_tokens":384,
                "text_tokens":119,
                "audio_tokens":8,
                "cached_tokens_details": {
                    "text_tokens": 128,
                    "audio_tokens": 256
                }
            },
            "output_token_details": {
              "text_tokens":36,
              "audio_tokens":112
            }
        }
    }
}
```

response.output_item.added

Returned when a new Item is created during Response generation.

event_id string

The unique ID of the server event.

item object
The item to add to the conversation.

Hide properties

arguments string
The arguments of the function call (for `function_call` items).

call_id string
The ID of the function call (for `function_call` and `function_call_output` items). If passed on a `function_call_output` item, the server will check that a `function_call` item with the same ID exists in the conversation history.

content array
The content of the message, applicable for `message` items.

Message items of role `system` support only `input_text` content

Message items of role `user` support `input_text` and `input_audio` content

Message items of role `assistant` support `text` content.

Hide properties

audio string
Base64-encoded audio bytes, used for `input_audio` content type.

id string
ID of a previous conversation item to reference (for `item_reference` content types in `response.create` events). These can reference both client and server created items.

text string
The text content, used for `input_text` and `text` content types.

transcript string
The transcript of the audio, used for `input_audio` content type.

type string
The content type (`input_text` , `input_audio` , `item_reference` , `text`).

id string
The unique ID of the item, this can be generated by the client to help manage server-side context, but is not required because the server will generate one if not provided.

name string

The name of the function being called (for `function_call` items).

object string
Identifier for the API object being returned - always `realtime.item`.

output string
The output of the function call (for `function_call_output` items).

role string
The role of the message sender (`user`, `assistant`, `system`), only applicable for `message` items.

status string
The status of the item (`completed`, `incomplete`, `in_progress`). These have no effect on the conversation, but are accepted for consistency with the `conversation.item.created` event.

type string
The type of the item (`message`, `function_call`, `function_call_output`).

output_index integer
The index of the output item in the Response.

response_id string
The ID of the Response to which the item belongs.

type string
The event type, must be `response.output_item.added`.

OBJECT response.output_item.added

```
{
    "event_id": "event_3334",
    "type": "response.output_item.added",
    "response_id": "resp_001",
    "output_index": 0,
    "item": {
        "id": "msg_007",
        "object": "realtime.item",
        "type": "message",
        "status": "in_progress",
        "role": "assistant",
        "content": []
    }

```

```
}
```

response.output_item.done

Returned when an Item is done streaming. Also emitted when a Response is interrupted, incomplete, or cancelled.

event_id string
The unique ID of the server event.

item object
The item to add to the conversation.

Hide properties

arguments string
The arguments of the function call (for `function_call` items).

call_id string
The ID of the function call (for `function_call` and `function_call_output` items). If passed on a `function_call_output` item, the server will check that a `function_call` item with the same ID exists in the conversation history.

content array
The content of the message, applicable for `message` items.

Message items of role `system` support only `input_text` content

Message items of role `user` support `input_text` and `input_audio` content

Message items of role `assistant` support `text` content.

Hide properties

audio string
Base64-encoded audio bytes, used for `input_audio` content type.

id string
ID of a previous conversation item to reference (for `item_reference` content types in `response.create` events). These can reference both client and server created items.

text string
The text content, used for `input_text` and `text` content types.

transcript string
The transcript of the audio, used for `input_audio` content type.

type string
The content type (`input_text` , `input_audio` , `item_reference` , `text`).

id string
The unique ID of the item, this can be generated by the client to help manage server-side context, but is not required because the server will generate one if not provided.

name string
The name of the function being called (for `function_call` items).

object string
Identifier for the API object being returned - always `realtime.item` .

output string
The output of the function call (for `function_call_output` items).

role string
The role of the message sender (`user` , `assistant` , `system`), only applicable for `message` items.

status string
The status of the item (`completed` , `incomplete` , `in_progress`). These have no effect on the conversation, but are accepted for consistency with the `conversation.item.created` event.

type string
The type of the item (`message` , `function_call` , `function_call_output`).

output_index integer
The index of the output item in the Response.

response_id string
The ID of the Response to which the item belongs.

type string
The event type, must be `response.output_item.done` .

```
OBJECT response.output_item.done
```

```
{
    "event_id": "event_3536",
    "type": "response.output_item.done",
    "response_id": "resp_001",
    "output_index": 0,
    "item": {
        "id": "msg_007",
        "object": "realtime.item",
        "type": "message",
        "status": "completed",
        "role": "assistant",
        "content": [
            {
                "type": "text",
                "text": "Sure, I can help with that."
            }
        ]
    }
}
```

response.content_part.added

Returned when a new content part is added to an assistant message item during response generation.

content_index integer
The index of the content part in the item's content array.

event_id string
The unique ID of the server event.

item_id string
The ID of the item to which the content part was added.

output_index integer
The index of the output item in the response.

part object

The content part that was added.

Hide properties

audio string
Base64-encoded audio data (if type is "audio").

text string
The text content (if type is "text").

transcript string
The transcript of the audio (if type is "audio").

type string
The content type ("text", "audio").

response_id string
The ID of the response.

type string
The event type, must be `response.content_part.added`.

OBJECT response.content_part.added

```
{
    "event_id": "event_3738",
    "type": "response.content_part.added",
    "response_id": "resp_001",
    "item_id": "msg_007",
    "output_index": 0,
    "content_index": 0,
    "part": {
        "type": "text",
        "text": ""
    }
}
```

response.content_part.done

Returned when a content part is done streaming in an assistant message item. Also emitted when a Response is interrupted, incomplete, or cancelled.

content_index integer
The index of the content part in the item's content array.

event_id string
The unique ID of the server event.

item_id string
The ID of the item.

output_index integer
The index of the output item in the response.

part object
The content part that is done.

Hide properties

audio string
Base64-encoded audio data (if type is "audio").

text string
The text content (if type is "text").

transcript string
The transcript of the audio (if type is "audio").

type string
The content type ("text", "audio").

response_id string
The ID of the response.

type string
The event type, must be `response.content_part.done`.

OBJECT response.content_part.done

```
{
    "event_id": "event_3940",
    "type": "response.content_part.done",
    "response_id": "resp_001",
    "item_id": "msg_007",
```

```
    "output_index": 0,
    "content_index": 0,
    "part": {
        "type": "text",
        "text": "Sure, I can help with that."
    }
}
```

response.text.delta

Returned when the text value of a "text" content part is updated.

content_index integer
The index of the content part in the item's content array.

delta string
The text delta.

event_id string
The unique ID of the server event.

item_id string
The ID of the item.

output_index integer
The index of the output item in the response.

response_id string
The ID of the response.

type string
The event type, must be `response.text.delta`.

OBJECT response.text.delta

```
{
    "event_id": "event_4142",
```

```
    "type": "response.text.delta",
    "response_id": "resp_001",
    "item_id": "msg_007",
    "output_index": 0,
    "content_index": 0,
    "delta": "Sure, I can h"
}
```

response.text.done

Returned when the text value of a "text" content part is done streaming. Also emitted when a Response is interrupted, incomplete, or cancelled.

content_index integer
The index of the content part in the item's content array.

event_id string
The unique ID of the server event.

item_id string
The ID of the item.

output_index integer
The index of the output item in the response.

response_id string
The ID of the response.

text string
The final text content.

type string
The event type, must be `response.text.done`.

OBJECT response.text.done

```
{
    "event_id": "event_4344",
    "type": "response.text.done",

```

```
    "response_id": "resp_001",
    "item_id": "msg_007",
    "output_index": 0,
    "content_index": 0,
    "text": "Sure, I can help with that."
}
```

response.audio_transcript.delta

Returned when the model-generated transcription of audio output is updated.

content_index integer
The index of the content part in the item's content array.

delta string
The transcript delta.

event_id string
The unique ID of the server event.

item_id string
The ID of the item.

output_index integer
The index of the output item in the response.

response_id string
The ID of the response.

type string
The event type, must be `response.audio_transcript.delta`.

OBJECT response.audio_transcript.delta

```
{
    "event_id": "event_4546",
    "type": "response.audio_transcript.delta",
```

```
    "response_id": "resp_001",
    "item_id": "msg_008",
    "output_index": 0,
    "content_index": 0,
    "delta": "Hello, how can I a"
}
```

response.audio_transcript.done

Returned when the model-generated transcription of audio output is done streaming. Also emitted when a Response is interrupted, incomplete, or cancelled.

content_index integer
The index of the content part in the item's content array.

event_id string
The unique ID of the server event.

item_id string
The ID of the item.

output_index integer
The index of the output item in the response.

response_id string
The ID of the response.

transcript string
The final transcript of the audio.

type string
The event type, must be `response.audio_transcript.done`.

OBJECT response.audio_transcript.done

```
{
    "event_id": "event_4748",
    "type": "response.audio_transcript.done",
    "response_id": "resp_001",

```

```
    "item_id": "msg_008",
    "output_index": 0,
    "content_index": 0,
    "transcript": "Hello, how can I assist you today?"
}
```

response.audio.delta

Returned when the model-generated audio is updated.

content_index integer
The index of the content part in the item's content array.

delta string
Base64-encoded audio data delta.

event_id string
The unique ID of the server event.

item_id string
The ID of the item.

output_index integer
The index of the output item in the response.

response_id string
The ID of the response.

type string
The event type, must be `response.audio.delta`.

OBJECT response.audio.delta

```
{
    "event_id": "event_4950",
    "type": "response.audio.delta",
    "response_id": "resp_001",
```

```
5     "item_id": "msg_008",
6     "output_index": 0,
7     "content_index": 0,
8     "delta": "Base64EncodedAudioDelta"
9 }
```

response.audio.done

Returned when the model-generated audio is done. Also emitted when a Response is interrupted, incomplete, or cancelled.

content_index integer
The index of the content part in the item's content array.

event_id string
The unique ID of the server event.

item_id string
The ID of the item.

output_index integer
The index of the output item in the response.

response_id string
The ID of the response.

type string
The event type, must be `response.audio.done`.

OBJECT response.audio.done

```
{
    "event_id": "event_5152",
    "type": "response.audio.done",
    "response_id": "resp_001",
    "item_id": "msg_008",
    "output_index": 0,
    "content_index": 0
}
```

response.function_call_arguments.delta

Returned when the model-generated function call arguments are updated.

call_id string
The ID of the function call.

delta string
The arguments delta as a JSON string.

event_id string
The unique ID of the server event.

item_id string
The ID of the function call item.

output_index integer
The index of the output item in the response.

response_id string
The ID of the response.

type string
The event type, must be `response.function_call_arguments.delta`.

OBJECT response.function_call_arguments.delta

```
{
    "event_id": "event_5354",
    "type": "response.function_call_arguments.delta",
    "response_id": "resp_002",
    "item_id": "fc_001",
    "output_index": 0,
    "call_id": "call_001",
    "delta": "{\"location\": \"San\""
}
```

response.function_call_arguments.done

Returned when the model-generated function call arguments are done streaming. Also emitted when a Response is interrupted, incomplete, or cancelled.

arguments string

The final arguments as a JSON string.

call_id string
The ID of the function call.

event_id string
The unique ID of the server event.

item_id string
The ID of the function call item.

output_index integer
The index of the output item in the response.

response_id string
The ID of the response.

type string
The event type, must be `response.function_call_arguments.done`.

OBJECT response.function_call_arguments.done

```
{
    "event_id": "event_5556",
    "type": "response.function_call_arguments.done",
    "response_id": "resp_002",
    "item_id": "fc_001",
    "output_index": 0,
    "call_id": "call_001",
    "arguments": "{\"location\": \"San Francisco\"}"
}
```

transcription_session.updated

Returned when a transcription session is updated with a `transcription_session.update` event, unless there is an error.

event_id string
The unique ID of the server event.

session object
A new Realtime transcription session configuration.

When a session is created on the server via REST API, the session object also contains an ephemeral key. Default TTL for keys is 10 minutes. This property is not present when a session is updated via the WebSocket API.

Hide properties

client_secret object
Ephemeral key returned by the API. Only present when the session is created on the server via REST API.

Hide properties

expires_at integer
Timestamp for when the token expires. Currently, all tokens expire after one minute.

value string
Ephemeral key usable in client environments to authenticate connections to the Realtime API. Use this in client-side environments rather than a standard API token, which should only be used server-side.

input_audio_format string
The format of input audio. Options are `pcm16`, `g711_ulaw`, or `g711_alaw`.

input_audio_transcription object
Configuration of the transcription model.

Hide properties

language string
The language of the input audio. Supplying the input language in ISO-639-1 (e.g. `en`) format will improve accuracy and latency.

model string
The model to use for transcription. Can be `gpt-4o-transcribe`, `gpt-4o-mini-transcribe`, or `whisper-1`.

prompt string
An optional text to guide the model's style or continue a previous audio segment. The prompt should match the audio language.

modalities
The set of modalities the model can respond with. To disable audio, set this to ["text"].

turn_detection object
Configuration for turn detection. Can be set to `null` to turn off. Server VAD means that the model will detect the start and end of speech based on audio volume and respond at the end of user speech.

Hide properties

prefix_padding_ms integer
Amount of audio to include before the VAD detected speech (in milliseconds). Defaults to 300ms.

silence_duration_ms integer
Duration of silence to detect speech stop (in milliseconds). Defaults to 500ms. With shorter values the model will respond more quickly, but may jump in on short pauses from the user.

threshold number
Activation threshold for VAD (0.0 to 1.0), this defaults to 0.5. A higher threshold will require louder audio to activate the model, and thus might perform better in noisy environments.

type string
Type of turn detection, only `server_vad` is currently supported.

type string
The event type, must be `transcription_session.updated`.

OBJECT transcription_session.updated

```
{
  "event_id": "event_5678",
  "type": "transcription_session.updated",
  "session": {
    "id": "sess_001",
    "object": "realtime.transcription_session",
    "input_audio_format": "pcm16",
    "input_audio_transcription": {
      "model": "gpt-4o-transcribe",
      "prompt": "",
      "language": ""
    },
    "turn_detection": {
      "type": "server_vad",
      "threshold": 0.5,
```

```
      "prefix_padding_ms": 300,
      "silence_duration_ms": 500,
      "create_response": true,
      // "interrupt_response": false  -- this will NOT be returned
    },
    "input_audio_noise_reduction": {
      "type": "near_field"
    },
    "include": [
      "item.input_audio_transcription.avg_logprob",
    ],
  }
}
```

rate_limits.updated

Emitted at the beginning of a Response to indicate the updated rate limits. When a Response is created some tokens will be "reserved" for the output tokens, the rate limits shown here reflect that reservation, which is then adjusted accordingly once the Response is completed.

event_id string
The unique ID of the server event.

rate_limits array

List of rate limit information.

Hide properties

limit integer

The maximum allowed value for the rate limit.

name string

The name of the rate limit (`requests` , `tokens`).

remaining integer

The remaining value before the limit is reached.

reset_seconds number

Seconds until the rate limit resets.

type string

The event type, must be `rate_limits.updated` .

OBJECT rate_limits.updated

```
{
    "event_id": "event_5758",
    "type": "rate_limits.updated",
    "rate_limits": [
        {
            "name": "requests",
            "limit": 1000,
            "remaining": 999,
            "reset_seconds": 60
        },
        {
            "name": "tokens",
            "limit": 50000,
            "remaining": 49950,
            "reset_seconds": 60
        }
    ]
}
```

output_audio_buffer.started

WebRTC Only: Emitted when the server begins streaming audio to the client. This event is emitted after an audio content part has been added (`response.content_part.added`) to the response. Learn more.

event_id string
The unique ID of the server event.

response_id string
The unique ID of the response that produced the audio.

type string
The event type, must be `output_audio_buffer.started` .

OBJECT output_audio_buffer.started

```
{
    "event_id": "event_abc123",
    "type": "output_audio_buffer.started",
    "response_id": "resp_abc123"
}
```

output_audio_buffer.stopped

WebRTC Only: Emitted when the output audio buffer has been completely drained on the server, and no more audio is forthcoming. This event is emitted after the full response data has been sent to the client (`response.done`). Learn more.

event_id string
The unique ID of the server event.

response_id string
The unique ID of the response that produced the audio.

type string
The event type, must be `output_audio_buffer.stopped` .

OBJECT output_audio_buffer.stopped

```
{
    "event_id": "event_abc123",
    "type": "output_audio_buffer.stopped",
    "response_id": "resp_abc123"
}
```

output_audio_buffer.cleared

WebRTC Only: Emitted when the output audio buffer is cleared. This happens either in VAD mode when the user has interrupted (`input_audio_buffer.speech_started`), or when the client has emitted the `output_audio_buffer.clear` event to manually cut off the current audio response. Learn more.

event_id string
The unique ID of the server event.

response_id string
The unique ID of the response that produced the audio.

type string
The event type, must be `output_audio_buffer.cleared` .

OBJECT output_audio_buffer.cleared

```
{
    "event_id": "event_abc123",
    "type": "output_audio_buffer.cleared",
    "response_id": "resp_abc123"
}
```

Audio

Learn how to turn audio into text or text into audio.

Related guide: Speech to text

Create speech

```
POST https://api.openai.com/v1/audio/speech
```

Generates audio from the input text.

Request body

`input` string Required
The text to generate audio for. The maximum length is 4096 characters.

`model` string Required
One of the available TTS models: `tts-1`, `tts-1-hd` or `gpt-4o-mini-tts`.

`voice` string Required
The voice to use when generating the audio. Supported voices are `alloy`, `ash`, `ballad`, `coral`, `echo`, `fable`, `onyx`, `nova`, `sage`, `shimmer`, and `verse`. Previews of the voices are available in the Text to speech guide.

`instructions` string Optional
Control the voice of your generated audio with additional instructions. Does not work with `tts-1` or `tts-1-hd`.

`response_format` string Optional Defaults to mp3
The format to audio in. Supported formats are `mp3`, `opus`, `aac`, `flac`, `wav`, and `pcm`.

`speed` number Optional Defaults to 1
The speed of the generated audio. Select a value from `0.25` to `4.0`. `1.0` is the default.

`stream_format` string Optional Defaults to audio
The format to stream the audio in. Supported formats are `sse` and `audio`. `sse` is not supported for `tts-1` or `tts-1-hd`.

Returns

The audio file content or a stream of audio events.

Default SSE Stream Format

Example request curl

```
curl https://api.openai.com/v1/audio/speech \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "model": "gpt-4o-mini-tts",
    "input": "The quick brown fox jumped over the lazy dog.",
    "voice": "alloy"
  }' \
  --output speech.mp3
```

Create transcription

`POST https://api.openai.com/v1/audio/transcriptions`

Transcribes audio into the input language.

Request body

`file` file Required
The audio file object (not file name) to transcribe, in one of these formats: flac, mp3, mp4, mpeg, mpga, m4a, ogg, wav, or webm.

`model` string Required
ID of the model to use. The options are `gpt-4o-transcribe`, `gpt-4o-mini-transcribe`, and `whisper-1` (which is powered by our open source Whisper V2 model).

`chunking_strategy` "auto" or object Optional
Controls how the audio is cut into chunks. When set to `"auto"`, the server first normalizes loudness and then uses voice activity detection (VAD) to choose boundaries. `server_vad` object can be provided to tweak VAD detection parameters manually. If unset, the audio is transcribed as a single block.

Hide possible types

string

Automatically set chunking parameters based on the audio. Must be set to `"auto"`.

object

Hide properties

`type` string Required

Must be set to `server_vad` to enable manual chunking using server side VAD.

`prefix_padding_ms` integer Optional Defaults to 300

Amount of audio to include before the VAD detected speech (in milliseconds).

`silence_duration_ms` integer Optional Defaults to 200

Duration of silence to detect speech stop (in milliseconds). With shorter values the model will respond more quickly, but may jump in on short pauses from the user.

`threshold` number Optional Defaults to 0.5

Sensitivity threshold (0.0 to 1.0) for voice activity detection. A higher threshold will require louder audio to activate the model, and thus might perform better in noisy environments.

`include[]` array Optional

Additional information to include in the transcription response. `logprobs` will return the log probabilities of the tokens in the response to understand the model's confidence in the transcription. `logprobs` only works with response_format set to `json` and only with the models `gpt-4o-transcribe` and `gpt-4o-mini-transcribe`.

`language` string Optional

The language of the input audio. Supplying the input language in ISO-639-1 (e.g. `en`) format will improve accuracy and latency.

`prompt` string Optional

An optional text to guide the model's style or continue a previous audio segment. The prompt should match the audio language.

`response_format` string Optional Defaults to json

The format of the output, in one of these options: `json`, `text`, `srt`, `verbose_json`, or `vtt`. For `gpt-4o-transcribe` and `gpt-4o-mini-transcribe`, the only supported format is `json`.

`stream` boolean or null Optional Defaults to false

If set to true, the model response data will be streamed to the client as it is generated using server-sent

events. See the Streaming section of the Speech-to-Text guide for more information.

Note: Streaming is not supported for the `whisper-1` model and will be ignored.

`temperature` number Optional Defaults to 0

The sampling temperature, between 0 and 1. Higher values like 0.8 will make the output more random, while lower values like 0.2 will make it more focused and deterministic. If set to 0, the model will use log probability to automatically increase the temperature until certain thresholds are hit.

`timestamp_granularities[]` array Optional Defaults to segment

The timestamp granularities to populate for this transcription. `response_format` must be set `verbose_json` to use timestamp granularities. Either or both of these options are supported: `word`, or `segment`. Note: There is no additional latency for segment timestamps, but generating word timestamps incurs additional latency.

Returns

The transcription object, a verbose transcription object or a stream of transcript events.

Default | Streaming | Logprobs | Word timestamps | Segment timestamps

Example request curl

```
curl https://api.openai.com/v1/audio/transcriptions \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: multipart/form-data" \
  -F file="@/path/to/file/audio.mp3" \
  -F model="gpt-4o-transcribe"
```

Response

```
{
  "text": "Imagine the wildest idea that you've ever had, and you're curious about how it
  "usage": {
    "type": "tokens",
    "input_tokens": 14,
    "input_token_details": {
      "text_tokens": 0,
      "audio_tokens": 14
    },
    "output_tokens": 45,
    "total_tokens": 59
```

```
  }
}
```

Create translation

`POST https://api.openai.com/v1/audio/translations`

Translates audio into English.

Request body

`file` file Required
The audio file object (not file name) translate, in one of these formats: flac, mp3, mp4, mpeg, mpga, m4a, ogg, wav, or webm.

`model` string or "whisper-1" Required
ID of the model to use. Only `whisper-1` (which is powered by our open source Whisper V2 model) is currently available.

`prompt` string Optional
An optional text to guide the model's style or continue a previous audio segment. The prompt should be in English.

`response_format` string Optional Defaults to json
The format of the output, in one of these options: `json`, `text`, `srt`, `verbose_json`, or `vtt`.

`temperature` number Optional Defaults to 0
The sampling temperature, between 0 and 1. Higher values like 0.8 will make the output more random, while lower values like 0.2 will make it more focused and deterministic. If set to 0, the model will use log probability to automatically increase the temperature until certain thresholds are hit.

Returns

The translated text.

Example request curl

```
curl https://api.openai.com/v1/audio/translations \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: multipart/form-data" \
  -F file="@/path/to/file/german.m4a" \
  -F model="whisper-1"
```

Response

```
{
  "text": "Hello, my name is Wolfgang and I come from Germany. Where are you heading today?
}
```

The transcription object (JSON)

Represents a transcription response returned by model, based on the provided input.

logprobs array
The log probabilities of the tokens in the transcription. Only returned with the models `gpt-4o-transcribe` and `gpt-4o-mini-transcribe` if `logprobs` is added to the `include` array.

Hide properties

bytes array
The bytes of the token.

logprob number
The log probability of the token.

token string
The token in the transcription.

text string
The transcribed text.

usage object
Token usage statistics for the request.

Hide possible types

Token Usage object
Usage statistics for models billed by token usage.

Hide properties

input_tokens integer
Number of input tokens billed for this request.

output_tokens integer
Number of output tokens generated.

total_tokens integer
Total number of tokens used (input + output).

type string
The type of the usage object. Always `tokens` for this variant.

input_token_details object
Details about the input tokens billed for this request.

Hide properties

audio_tokens integer
Number of audio tokens billed for this request.

text_tokens integer
Number of text tokens billed for this request.

Duration Usage object
Usage statistics for models billed by audio input duration.

Hide properties

seconds number
Duration of the input audio in seconds.

type string
The type of the usage object. Always `duration` for this variant.

OBJECT The transcription object (JSON)

```
{
  "text": "Imagine the wildest idea that you've ever had, and you're curious about how it
  "usage": {
    "type": "tokens",
    "input_tokens": 14,
```

```
    "input_token_details": {
      "text_tokens": 10,
      "audio_tokens": 4
    },
    "output_tokens": 101,
    "total_tokens": 115
  }
}
```

The transcription object (Verbose JSON)

Represents a verbose json transcription response returned by model, based on the provided input.

`duration` number
The duration of the input audio.

`language` string
The language of the input audio.

`segments` array
Segments of the transcribed text and their corresponding details.

Hide properties

`avg_logprob` number
Average logprob of the segment. If the value is lower than -1, consider the logprobs failed.

`compression_ratio` number
Compression ratio of the segment. If the value is greater than 2.4, consider the compression failed.

`end` number
End time of the segment in seconds.

`id` integer
Unique identifier of the segment.

`no_speech_prob` number
Probability of no speech in the segment. If the value is higher than 1.0 and the `avg_logprob` is below -1, consider this segment silent.

seek integer
Seek offset of the segment.

start number
Start time of the segment in seconds.

temperature number
Temperature parameter used for generating the segment.

text string
Text content of the segment.

tokens array
Array of token IDs for the text content.

text string
The transcribed text.

usage object
Usage statistics for models billed by audio input duration.

Hide properties

seconds number
Duration of the input audio in seconds.

type string
The type of the usage object. Always `duration` for this variant.

words array

Extracted words and their corresponding timestamps.

Hide properties

end number

End time of the word in seconds.

start number

Start time of the word in seconds.

word string

The text content of the word.

OBJECT The transcription object (Verbose JSON)


```
{
  "task": "transcribe",
  "language": "english",
  "duration": 8.470000267028809,
  "text": "The beach was a popular spot on a hot summer day. People were swimming in the c
  "segments": [
    {
      "id": 0,
      "seek": 0,
      "start": 0.0,
      "end": 3.319999933242798,
      "text": " The beach was a popular spot on a hot summer day.",
      "tokens": [
        50364, 440, 7534, 390, 257, 3743, 4008, 322, 257, 2368, 4266, 786, 13, 50530
      ],
      "temperature": 0.0,
      "avg_logprob": -0.2860786020755768,
      "compression_ratio": 1.2363636493682861,
      "no_speech_prob": 0.00985979475080967
    },
    ...
  ],
  "usage": {
    "type": "duration",
    "seconds": 9
  }
}
```

Stream Event (speech.audio.delta)

Emitted for each chunk of audio data generated during speech synthesis.

audio string
A chunk of Base64-encoded audio data.

type string
The type of the event. Always `speech.audio.delta`.

OBJECT Stream Event (speech.audio.delta)

```
{
  "type": "speech.audio.delta",
  "audio": "base64-encoded-audio-data"
}
```

Stream Event (speech.audio.done)

Emitted when the speech synthesis is complete and all audio has been streamed.

type string
The type of the event. Always `speech.audio.done`.

usage object
Token usage statistics for the request.

Hide properties

input_tokens integer
Number of input tokens in the prompt.

output_tokens integer
Number of output tokens generated.

total_tokens integer
Total number of tokens used (input + output).

OBJECT Stream Event (speech.audio.done)

```
{
  "type": "speech.audio.done",
  "usage": {
    "input_tokens": 14,
    "output_tokens": 101,
    "total_tokens": 115
  }
}
```

Stream Event (transcript.text.delta)

Emitted when there is an additional text delta. This is also the first event emitted when the transcription starts. Only emitted when you create a transcription with the `Stream` parameter set to `true`.

delta string
The text delta that was additionally transcribed.

logprobs array
The log probabilities of the delta. Only included if you create a transcription with the `include[]` parameter set to `logprobs`.

Hide properties

bytes array
The bytes that were used to generate the log probability.

logprob number
The log probability of the token.

token string
The token that was used to generate the log probability.

type string
The type of the event. Always `transcript.text.delta`.

OBJECT Stream Event (transcript.text.delta)

```
{
  "type": "transcript.text.delta",
```

```
3   "delta": " wonderful"
4 }
```

Stream Event (transcript.text.done)

Emitted when the transcription is complete. Contains the complete transcription text. Only emitted when you create a transcription with the `Stream` parameter set to `true`.

logprobs array
The log probabilities of the individual tokens in the transcription. Only included if you create a transcription with the `include[]` parameter set to `logprobs`.

Hide properties

bytes array
The bytes that were used to generate the log probability.

logprob number
The log probability of the token.

token string
The token that was used to generate the log probability.

text string
The text that was transcribed.

type string
The type of the event. Always `transcript.text.done`.

usage object
Usage statistics for models billed by token usage.

Hide properties

input_tokens integer
Number of input tokens billed for this request.

output_tokens integer
Number of output tokens generated.

total_tokens integer

Total number of tokens used (input + output).

type string
The type of the usage object. Always `tokens` for this variant.

input_token_details object
Details about the input tokens billed for this request.

Hide properties

audio_tokens integer
Number of audio tokens billed for this request.

text_tokens integer
Number of text tokens billed for this request.

OBJECT Stream Event (transcript.text.done)

```
{
  "type": "transcript.text.done",
  "text": "I see skies of blue and clouds of white, the bright blessed days, the dark sacr
  "usage": {
    "type": "tokens",
    "input_tokens": 14,
    "input_token_details": {
      "text_tokens": 10,
      "audio_tokens": 4
    },
    "output_tokens": 31,
    "total_tokens": 45
  }
}
```

Images

Given a prompt and/or an input image, the model will generate a new image. Related guide: Image generation

Create image

```
POST https://api.openai.com/v1/images/generations
```

Creates an image given a prompt. Learn more.

Request body

prompt string Required
A text description of the desired image(s). The maximum length is 32000 characters for `gpt-image-1`, 1000 characters for `dall-e-2` and 4000 characters for `dall-e-3`.

background string or null Optional Defaults to auto
Allows to set transparency for the background of the generated image(s). This parameter is only supported for `gpt-image-1`. Must be one of `transparent`, `opaque` or `auto` (default value). When `auto` is used, the model will automatically determine the best background for the image.

If `transparent`, the output format needs to support transparency, so it should be set to either `png` (default value) or `webp`.

model string Optional Defaults to dall-e-2
The model to use for image generation. One of `dall-e-2`, `dall-e-3`, or `gpt-image-1`. Defaults to `dall-e-2` unless a parameter specific to `gpt-image-1` is used.

moderation string or null Optional Defaults to auto
Control the content-moderation level for images generated by `gpt-image-1`. Must be either `low` for less restrictive filtering or `auto` (default value).

n integer or null Optional Defaults to 1
The number of images to generate. Must be between 1 and 10. For `dall-e-3`, only `n=1` is supported.

output_compression integer or null Optional Defaults to 100
The compression level (0-100%) for the generated images. This parameter is only supported for `gpt-image-1` with the `webp` or `jpeg` output formats, and defaults to 100.

output_format string or null Optional Defaults to png
The format in which the generated images are returned. This parameter is only supported for `gpt-image-1`. Must be one of `png`, `jpeg`, or `webp`.

quality string or null Optional Defaults to auto
The quality of the image that will be generated.

- `auto` (default value) will automatically select the best quality for the given model.

`high`, `medium` and `low` are supported for `gpt-image-1`.

`hd` and `standard` are supported for `dall-e-3`.

`standard` is the only option for `dall-e-2`.

`response_format` string or null Optional Defaults to url

The format in which generated images with `dall-e-2` and `dall-e-3` are returned. Must be one of `url` or `b64_json`. URLs are only valid for 60 minutes after the image has been generated. This parameter isn't supported for `gpt-image-1` which will always return base64-encoded images.

`size` string or null Optional Defaults to auto

The size of the generated images. Must be one of `1024x1024`, `1536x1024` (landscape), `1024x1536` (portrait), or `auto` (default value) for `gpt-image-1`, one of `256x256`, `512x512`, or `1024x1024` for `dall-e-2`, and one of `1024x1024`, `1792x1024`, or `1024x1792` for `dall-e-3`.

`style` string or null Optional Defaults to vivid

The style of the generated images. This parameter is only supported for `dall-e-3`. Must be one of `vivid` or `natural`. Vivid causes the model to lean towards generating hyper-real and dramatic images. Natural causes the model to produce more natural, less hyper-real looking images.

`user` string Optional

A unique identifier representing your end-user, which can help OpenAI to monitor and detect abuse. Learn more.

Returns

Returns an image object.

Example request curl

```curl
curl https://api.openai.com/v1/images/generations \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -d '{
    "model": "gpt-image-1",
    "prompt": "A cute baby sea otter",
    "n": 1,
    "size": "1024x1024"
  }'
```

Response

```
{
  "created": 1713833628,
  "data": [
    {
      "b64_json": "..."
    }
  ],
  "usage": {
    "total_tokens": 100,
    "input_tokens": 50,
    "output_tokens": 50,
    "input_tokens_details": {
      "text_tokens": 10,
      "image_tokens": 40
    }
  }
}
```

Create image edit

`POST https://api.openai.com/v1/images/edits`

Creates an edited or extended image given one or more source images and a prompt. This endpoint only supports `gpt-image-1` and `dall-e-2`.

Request body

`image` string or array Required

The image(s) to edit. Must be a supported image file or an array of images.

For `gpt-image-1`, each image should be a `png`, `webp`, or `jpg` file less than 50MB. You can provide up to 16 images.

For `dall-e-2`, you can only provide one image, and it should be a square `png` file less than 4MB.

`prompt` string Required

A text description of the desired image(s). The maximum length is 1000 characters for `dall-e-2`, and 32000 characters for `gpt-image-1`.

background string or null Optional Defaults to auto

Allows to set transparency for the background of the generated image(s). This parameter is only supported for `gpt-image-1`. Must be one of `transparent`, `opaque` or `auto` (default value). When `auto` is used, the model will automatically determine the best background for the image.

If `transparent`, the output format needs to support transparency, so it should be set to either `png` (default value) or `webp`.

mask file Optional

An additional image whose fully transparent areas (e.g. where alpha is zero) indicate where `image` should be edited. If there are multiple images provided, the mask will be applied on the first image. Must be a valid PNG file, less than 4MB, and have the same dimensions as `image`.

model string Optional Defaults to dall-e-2

The model to use for image generation. Only `dall-e-2` and `gpt-image-1` are supported. Defaults to `dall-e-2` unless a parameter specific to `gpt-image-1` is used.

n integer or null Optional Defaults to 1

The number of images to generate. Must be between 1 and 10.

output_compression integer or null Optional Defaults to 100

The compression level (0-100%) for the generated images. This parameter is only supported for `gpt-image-1` with the `webp` or `jpeg` output formats, and defaults to 100.

output_format string or null Optional Defaults to png

The format in which the generated images are returned. This parameter is only supported for `gpt-image-1`. Must be one of `png`, `jpeg`, or `webp`. The default value is `png`.

quality string or null Optional Defaults to auto

The quality of the image that will be generated. `high`, `medium` and `low` are only supported for `gpt-image-1`. `dall-e-2` only supports `standard` quality. Defaults to `auto`.

response_format string or null Optional Defaults to url

The format in which the generated images are returned. Must be one of `url` or `b64_json`. URLs are only valid for 60 minutes after the image has been generated. This parameter is only supported for `dall-e-2`, as `gpt-image-1` will always return base64-encoded images.

size string or null Optional Defaults to 1024×1024

The size of the generated images. Must be one of `1024x1024`, `1536x1024` (landscape), `1024x1536` (portrait), or `auto` (default value) for `gpt-image-1`, and one of `256x256`, `512x512`, or `1024x1024` for `dall-e-2`.

user string Optional

A unique identifier representing your end-user, which can help OpenAI to monitor and detect abuse. Learn more.

Returns

Returns an image object.

Example request curl

```
curl -s -D >(grep -i x-request-id >&2) \
  -o >(jq -r '.data[0].b64_json' | base64 --decode > gift-basket.png) \
  -X POST "https://api.openai.com/v1/images/edits" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -F "model=gpt-image-1" \
  -F "image[]=@body-lotion.png" \
  -F "image[]=@bath-bomb.png" \
  -F "image[]=@incense-kit.png" \
  -F "image[]=@soap.png" \
  -F 'prompt=Create a lovely gift basket with these four items in it'
```

Response

```
{
  "created": 1713833628,
  "data": [
    {
      "b64_json": "..."
    }
  ],
  "usage": {
    "total_tokens": 100,
    "input_tokens": 50,
    "output_tokens": 50,
    "input_tokens_details": {
      "text_tokens": 10,

```

```
      "image_tokens": 40
    }
  }
}
```

Create image variation

`POST https://api.openai.com/v1/images/variations`

Creates a variation of a given image. This endpoint only supports `dall-e-2`.

Request body

`image` file Required
The image to use as the basis for the variation(s). Must be a valid PNG file, less than 4MB, and square.

`model` string or "dall-e-2" Optional Defaults to dall-e-2
The model to use for image generation. Only `dall-e-2` is supported at this time.

`n` integer or null Optional Defaults to 1
The number of images to generate. Must be between 1 and 10.

`response_format` string or null Optional Defaults to url
The format in which the generated images are returned. Must be one of `url` or `b64_json`. URLs are only valid for 60 minutes after the image has been generated.

`size` string or null Optional Defaults to 1024×1024
The size of the generated images. Must be one of `256x256`, `512x512`, or `1024x1024`.

`user` string Optional
A unique identifier representing your end-user, which can help OpenAI to monitor and detect abuse. Learn more.

Returns

Returns a list of image objects.

Example request curl

```
curl https://api.openai.com/v1/images/variations \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -F image="@otter.png" \
  -F n=2 \
  -F size="1024x1024"
```

Response

```
{
  "created": 1589478378,
  "data": [
    {
      "url": "https://..."
    },
    {
      "url": "https://..."
    }
  ]
}
```

The image generation response

The response from the image generation endpoint.

background string
The background parameter used for the image generation. Either `transparent` or `opaque`.

created integer
The Unix timestamp (in seconds) of when the image was created.

data array
The list of generated images.

Hide properties

b64_json string
The base64-encoded JSON of the generated image. Default value for `gpt-image-1`, and only present if `response_format` is set to `b64_json` for `dall-e-2` and `dall-e-3`.

revised_prompt string
For `dall-e-3` only, the revised prompt that was used to generate the image.

url string
When using `dall-e-2` or `dall-e-3`, the URL of the generated image if `response_format` is set to `url` (default value). Unsupported for `gpt-image-1`.

output_format string
The output format of the image generation. Either `png`, `webp`, or `jpeg`.

quality string
The quality of the image generated. Either `low`, `medium`, or `high`.

size string
The size of the image generated. Either `1024x1024`, `1024x1536`, or `1536x1024`.

usage object
For `gpt-image-1` only, the token usage information for the image generation.

Hide properties

input_tokens integer
The number of tokens (images and text) in the input prompt.

input_tokens_details object
The input tokens detailed information for the image generation.

Hide properties

image_tokens integer
The number of image tokens in the input prompt.

text_tokens integer
The number of text tokens in the input prompt.

output_tokens integer
The number of image tokens in the output image.

total_tokens integer
The total number of tokens (images and text) used for the image generation.

OBJECT The image generation response

```
{
  "created": 1713833628,
  "data": [
    {
      "b64_json": "..."
    }
  ],
  "background": "transparent",
  "output_format": "png",
  "size": "1024x1024",
  "quality": "high",
  "usage": {
    "total_tokens": 100,
    "input_tokens": 50,
    "output_tokens": 50,
    "input_tokens_details": {
      "text_tokens": 10,
      "image_tokens": 40
    }
  }
}
```

Embeddings

Get a vector representation of a given input that can be easily consumed by machine learning models and algorithms. Related guide: Embeddings

Create embeddings

POST https://api.openai.com/v1/embeddings

Creates an embedding vector representing the input text.

Request body

input string or array Required

Input text to embed, encoded as a string or array of tokens. To embed multiple inputs in a single request,

pass an array of strings or array of token arrays. The input must not exceed the max input tokens for the model (8192 tokens for all embedding models), cannot be an empty string, and any array must be 2048 dimensions or less. Example Python code for counting tokens. In addition to the per-input token limit, all embedding models enforce a maximum of 300,000 tokens summed across all inputs in a single request.

model string Required
ID of the model to use. You can use the List models API to see all of your available models, or see our Model overview for descriptions of them.

dimensions integer Optional
The number of dimensions the resulting output embeddings should have. Only supported in `text-embedding-3` and later models.

encoding_format string Optional Defaults to float
The format to return the embeddings in. Can be either `float` or `base64`.

user string Optional
A unique identifier representing your end-user, which can help OpenAI to monitor and detect abuse. Learn more.

Returns

A list of embedding objects.

Example request curl

```
curl https://api.openai.com/v1/embeddings \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "input": "The food was delicious and the waiter...",
    "model": "text-embedding-ada-002",
    "encoding_format": "float"
  }'
```

Response

```
{
  "object": "list",
  "data": [
```

```
    {
      "object": "embedding",
      "embedding": [
        0.0023064255,
        -0.009327292,
        .... (1536 floats total for ada-002)
        -0.0028842222,
      ],
      "index": 0
    }
  ],
  "model": "text-embedding-ada-002",
  "usage": {
    "prompt_tokens": 8,
    "total_tokens": 8
  }
}
```

The embedding object

Represents an embedding vector returned by embedding endpoint.

`embedding` array
The embedding vector, which is a list of floats. The length of vector depends on the model as listed in the embedding guide.

`index` integer
The index of the embedding in the list of embeddings.

`object` string
The object type, which is always "embedding".

OBJECT The embedding object

```
{
  "object": "embedding",
  "embedding": [
    0.0023064255,
    -0.009327292,
    .... (1536 floats total for ada-002)
    -0.0028842222,
  ],
```

```
9     "index": 0
10 }
```

Evals

Create, manage, and run evals in the OpenAI platform. Related guide: Evals

Create eval

`POST https://api.openai.com/v1/evals`

Create the structure of an evaluation that can be used to test a model's performance. An evaluation is a set of testing criteria and the config for a data source, which dictates the schema of the data used in the evaluation. After creating an evaluation, you can run it on different models and model parameters. We support several types of graders and datasources. For more information, see the Evals guide.

Request body

`data_source_config` object Required
The configuration for the data source used for the evaluation runs. Dictates the schema of the data used in the evaluation.

Hide possible types

CustomDataSourceConfig object
A CustomDataSourceConfig object that defines the schema for the data source used for the evaluation runs. This schema is used to define the shape of the data that will be:

Used to define your testing criteria and

What data is required when creating a run

Hide properties

`item_schema` object Required
The json schema for each row in the data source.

`type` string Required
The type of data source. Always `custom`.

`include_sample_schema` boolean Optional Defaults to false
Whether the eval should expect you to populate the sample namespace (ie, by generating responses off of your data source)

LogsDataSourceConfig object
A data source config which specifies the metadata property of your logs query. This is usually metadata like `usecase=chatbot` or `prompt-version=v2`, etc.

Hide properties

`type` string Required
The type of data source. Always `logs`.

`metadata` object Optional
Metadata filters for the logs data source.

StoredCompletionsDataSourceConfig object Deprecated
Deprecated in favor of LogsDataSourceConfig.

Hide properties

`type` string Required
The type of data source. Always `stored_completions`.

`metadata` object Optional
Metadata filters for the stored completions data source.

`testing_criteria` array Required
A list of graders for all eval runs in this group. Graders can reference variables in the data source using double curly braces notation, like `{{item.variable_name}}`. To reference the model's output, use the `sample` namespace (ie, `{{sample.output_text}}`).

Hide possible types

LabelModelGrader object
A LabelModelGrader object which uses a model to assign labels to each item in the evaluation.

Hide properties

`input` array Required
A list of chat messages forming the prompt or context. May include variable references to the `item` namespace, ie {{item.name}}.

Hide possible types

SimpleInputMessage object

Hide properties

content string Required

The content of the message.

role string Required

The role of the message (e.g. "system", "assistant", "user").

Eval message object object

A message input to the model with a role indicating instruction following hierarchy. Instructions given with the `developer` or `system` role take precedence over instructions given with the `user` role. Messages with the `assistant` role are presumed to have been generated by the model in previous interactions.

Hide properties

content string / object / array Required

Inputs to the model - can contain template strings.

Hide possible types

Text input string

A text input to the model.

Input text object

A text input to the model.

Hide properties

text string Required

The text input to the model.

type string Required

The type of the input item. Always `input_text`.

Output text object

A text output from the model.

Hide properties

text string Required

The text output from the model.

type string Required

The type of the output text. Always `output_text`.

Input image object

An image input to the model.

Hide properties

`image_url` string Required

The URL of the image input.

`type` string Required

The type of the image input. Always `input_image`.

`detail` string Optional

The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

An array of Input text and Input image array

A list of inputs, each of which may be either an input text or input image object.

`role` string Required

The role of the message input. One of `user`, `assistant`, `system`, or `developer`.

`type` string Optional

The type of the message input. Always `message`.

`labels` array Required

The labels to classify to each item in the evaluation.

`model` string Required

The model to use for the evaluation. Must support structured outputs.

`name` string Required

The name of the grader.

`passing_labels` array Required

The labels that indicate a passing result. Must be a subset of labels.

`type` string Required

The object type, which is always `label_model`.

StringCheckGrader object

A StringCheckGrader object that performs a string comparison between input and reference using a specified operation.

Hide properties

input string Required

The input text. This may include template strings.

name string Required

The name of the grader.

operation string Required

The string check operation to perform. One of `eq`, `ne`, `like`, or `ilike`.

reference string Required

The reference text. This may include template strings.

type string Required

The object type, which is always `string_check`.

TextSimilarityGrader object

A TextSimilarityGrader object which grades text based on similarity metrics.

Hide properties

evaluation_metric string Required

The evaluation metric to use. One of `fuzzy_match`, `bleu`, `gleu`, `meteor`, `rouge_1`, `rouge_2`, `rouge_3`, `rouge_4`, `rouge_5`, or `rouge_l`.

input string Required

The text being graded.

name string Required

The name of the grader.

pass_threshold number Required

The threshold for the score.

reference string Required

The text being graded against.

type string Required

The type of grader.

PythonGrader object

A PythonGrader object that runs a python script on the input.

Hide properties

name string Required

The name of the grader.

source string Required

The source code of the python script.

type string Required

The object type, which is always `python`.

image_tag string Optional

The image tag to use for the python script.

pass_threshold number Optional

The threshold for the score.

ScoreModelGrader object

A ScoreModelGrader object that uses a model to assign a score to the input.

Hide properties

input array Required

The input text. This may include template strings.

Hide properties

content string / object / array Required

Inputs to the model - can contain template strings.

Hide possible types

Text input string

A text input to the model.

Input text object

A text input to the model.

Hide properties

text string Required
The text input to the model.

type string Required
The type of the input item. Always `input_text` .

Output text object
A text output from the model.

^ Hide properties

text string Required
The text output from the model.

type string Required
The type of the output text. Always `output_text` .

Input image object
An image input to the model.

^ Hide properties

image_url string Required
The URL of the image input.

type string Required
The type of the image input. Always `input_image` .

detail string Optional
The detail level of the image to be sent to the model. One of `high` , `low` , or `auto` . Defaults to `auto` .

An array of Input text and Input image array
A list of inputs, each of which may be either an input text or input image object.

role string Required
The role of the message input. One of `user` , `assistant` , `system` , or `developer` .

type string Optional
The type of the message input. Always `message` .

model string Required
The model to use for the evaluation.

name string Required
The name of the grader.

type string Required
The object type, which is always `score_model`.

pass_threshold number Optional
The threshold for the score.

range array Optional
The range of the score. Defaults to `[0, 1]`.

sampling_params object Optional
The sampling parameters for the model.

metadata map Optional
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

name string Optional
The name of the evaluation.

Returns

The created Eval object.

Example request curl

```
curl https://api.openai.com/v1/evals \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
        "name": "Sentiment",
        "data_source_config": {
```

```
          "type": "stored_completions",
          "metadata": {
              "usecase": "chatbot"
          }
        },
        "testing_criteria": [
          {
            "type": "label_model",
            "model": "o3-mini",
            "input": [
              {
                "role": "developer",
                "content": "Classify the sentiment of the following statement as one of 'p
              },
              {
                "role": "user",
                "content": "Statement: {{item.input}}"
              }
            ],
            "passing_labels": [
              "positive"
            ],
            "labels": [
              "positive",
              "neutral",
              "negative"
            ],
            "name": "Example label grader"
          }
        ]
      }'
```

Response

```
{
  "object": "eval",
  "id": "eval_67b7fa9a81a88190ab4aa417e397ea21",
  "data_source_config": {
    "type": "stored_completions",
    "metadata": {
      "usecase": "chatbot"
    },
    "schema": {
      "type": "object",
      "properties": {
        "item": {
```

```
13          "type": "object"
14        },
15        "sample": {
16          "type": "object"
17        }
18      },
19      "required": [
20        "item",
21        "sample"
22      ]
23    },
24    "testing_criteria": [
25      {
26        "name": "Example label grader",
27        "type": "label_model",
28        "model": "o3-mini",
29        "input": [
30          {
31            "type": "message",
32            "role": "developer",
33            "content": {
34              "type": "input_text",
35              "text": "Classify the sentiment of the following statement as one of positive,
36            }
37          },
38          {
39            "type": "message",
40            "role": "user",
41            "content": {
42              "type": "input_text",
43              "text": "Statement: {{item.input}}"
44            }
45          }
46        ],
47        "passing_labels": [
48          "positive"
49        ],
50        "labels": [
51          "positive",
52          "neutral",
53          "negative"
54        ]
55      }
56    ],
57    "name": "Sentiment",
58    "created_at": 1740110490,
59    "metadata": {
60
```

```
    "description": "An eval for sentiment analysis"
  }
}
```

Get an eval

```
GET https://api.openai.com/v1/evals/{eval_id}
```

Get an evaluation by ID.

Path parameters

eval_id string Required
The ID of the evaluation to retrieve.

Returns

The Eval object matching the specified ID.

Example request curl

```
curl https://api.openai.com/v1/evals/eval_67abd54d9b0081909a86353f6fb9317a \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
  "object": "eval",
  "id": "eval_67abd54d9b0081909a86353f6fb9317a",
  "data_source_config": {
    "type": "custom",
    "schema": {
      "type": "object",
      "properties": {
        "item": {
          "type": "object",
          "properties": {
            "input": {
```

```
              "type": "string"
            },
            "ground_truth": {
              "type": "string"
            }
          },
          "required": [
            "input",
            "ground_truth"
          ]
        }
      },
      "required": [
        "item"
      ]
    }
  },
  "testing_criteria": [
    {
      "name": "String check",
      "id": "String check-2eaf2d8d-d649-4335-8148-9535a7ca73c2",
      "type": "string_check",
      "input": "{{item.input}}",
      "reference": "{{item.ground_truth}}",
      "operation": "eq"
    }
  ],
  "name": "External Data Eval",
  "created_at": 1739314509,
  "metadata": {},
}
```

Update an eval

```
POST https://api.openai.com/v1/evals/{eval_id}
```

Update certain properties of an evaluation.

Path parameters

eval_id string Required

The ID of the evaluation to update.

Request body

metadata map Optional

Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

name string Optional

Rename the evaluation.

Returns

The Eval object matching the updated version.

Example request curl

```
curl https://api.openai.com/v1/evals/eval_67abd54d9b0081909a86353f6fb9317a \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{"name": "Updated Eval", "metadata": {"description": "Updated description"}}'
```

Response

```
{
  "object": "eval",
  "id": "eval_67abd54d9b0081909a86353f6fb9317a",
  "data_source_config": {
    "type": "custom",
    "schema": {
      "type": "object",
      "properties": {
        "item": {
          "type": "object",
          "properties": {
            "input": {
              "type": "string"
            },
            "ground_truth": {
              "type": "string"
            }
```

```
          },
          "required": [
            "input",
            "ground_truth"
          ]
        }
      },
      "required": [
        "item"
      ]
    }
  },
  "testing_criteria": [
    {
      "name": "String check",
      "id": "String check-2eaf2d8d-d649-4335-8148-9535a7ca73c2",
      "type": "string_check",
      "input": "{{item.input}}",
      "reference": "{{item.ground_truth}}",
      "operation": "eq"
    }
  ],
  "name": "Updated Eval",
  "created_at": 1739314509,
  "metadata": {"description": "Updated description"},
}
```

Delete an eval

```
DELETE https://api.openai.com/v1/evals/{eval_id}
```

Delete an evaluation.

Path parameters

eval_id string Required
The ID of the evaluation to delete.

Returns

A deletion confirmation object.

Example request curl

```curl
curl https://api.openai.com/v1/evals/eval_abc123 \
  -X DELETE \
  -H "Authorization: Bearer $OPENAI_API_KEY"
```

Response

```response
{
  "object": "eval.deleted",
  "deleted": true,
  "eval_id": "eval_abc123"
}
```

List evals

`GET https://api.openai.com/v1/evals`

List evaluations for a project.

Query parameters

after string Optional
Identifier for the last eval from the previous pagination request.

limit integer Optional Defaults to 20
Number of evals to retrieve.

order string Optional Defaults to asc
Sort order for evals by timestamp. Use `asc` for ascending order or `desc` for descending order.

order_by string Optional Defaults to created_at
Evals can be ordered by creation time or last updated time. Use `created_at` for creation time or `updated_at` for last updated time.

Returns

A list of evals matching the specified filters.

Example request curl

```curl
curl https://api.openai.com/v1/evals?limit=1 \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json"
```

Response

```response
{
  "object": "list",
  "data": [
    {
      "id": "eval_67abd54d9b0081909a86353f6fb9317a",
      "object": "eval",
      "data_source_config": {
        "type": "stored_completions",
        "metadata": {
          "usecase": "push_notifications_summarizer"
        },
        "schema": {
          "type": "object",
          "properties": {
            "item": {
              "type": "object"
            },
            "sample": {
              "type": "object"
            }
          },
          "required": [
            "item",
            "sample"
          ]
        }
      },
      "testing_criteria": [
        {
          "name": "Push Notification Summary Grader",
          "id": "Push Notification Summary Grader-9b876f24-4762-4be9-aff4-db7a9b31c673",
          "type": "label_model",
```

```
          "model": "o3-mini",
          "input": [
            {
              "type": "message",
              "role": "developer",
              "content": {
                "type": "input_text",
                "text": "\nLabel the following push notification summary as either correct
              }
            },
            {
              "type": "message",
              "role": "user",
              "content": {
                "type": "input_text",
                "text": "\nPush notifications: {{item.input}}\nSummary: {{sample.output_te
              }
            }
          ],
          "passing_labels": [
            "correct"
          ],
          "labels": [
            "correct",
            "incorrect"
          ],
          "sampling_params": null
        }
      ],
      "name": "Push Notification Summary Grader",
      "created_at": 1739314509,
      "metadata": {
        "description": "A stored completions eval for push notification summaries"
      }
    }
  ],
  "first_id": "eval_67abd54d9b0081909a86353f6fb9317a",
  "last_id": "eval_67aa884cf6688190b58f657d4441c8b7",
  "has_more": true
}
```


Get eval runs

```
GET https://api.openai.com/v1/evals/{eval_id}/runs
```

Get a list of runs for an evaluation.

Path parameters

eval_id string Required
The ID of the evaluation to retrieve runs for.

Query parameters

after string Optional
Identifier for the last run from the previous pagination request.

limit integer Optional Defaults to 20
Number of runs to retrieve.

order string Optional Defaults to asc
Sort order for runs by timestamp. Use `asc` for ascending order or `desc` for descending order. Defaults to `asc`.

status string Optional
Filter runs by status. One of `queued` | `in_progress` | `failed` | `completed` | `canceled`.

Returns

A list of EvalRun objects matching the specified ID.

Example request curl

```curl
curl https://api.openai.com/v1/evals/egroup_67abd54d9b0081909a86353f6fb9317a/runs \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
  "object": "list",
```

```
  "data": [
    {
      "object": "eval.run",
      "id": "evalrun_67e0c7d31560819090d60c0780591042",
      "eval_id": "eval_67e0c726d560819083f19a957c4c640b",
      "report_url": "https://platform.openai.com/evaluations/eval_67e0c726d560819083f19a95
      "status": "completed",
      "model": "o3-mini",
      "name": "bulk_with_negative_examples_o3-mini",
      "created_at": 1742784467,
      "result_counts": {
        "total": 1,
        "errored": 0,
        "failed": 0,
        "passed": 1
      },
      "per_model_usage": [
        {
          "model_name": "o3-mini",
          "invocation_count": 1,
          "prompt_tokens": 563,
          "completion_tokens": 874,
          "total_tokens": 1437,
          "cached_tokens": 0
        }
      ],
      "per_testing_criteria_results": [
        {
          "testing_criteria": "Push Notification Summary Grader-1808cd0b-eeec-4e0b-a519-33
          "passed": 1,
          "failed": 0
        }
      ],
      "data_source": {
        "type": "completions",
        "source": {
          "type": "file_content",
          "content": [
            {
              "item": {
                "notifications": "\n- New message from Sarah: \"Can you call me later?\"\r
              }
            }
          ]
        },
        "input_messages": {
          "type": "template",
```

```
          "template": [
            {
              "type": "message",
              "role": "developer",
              "content": {
                "type": "input_text",
                "text": "\n\n\n\nYou are a helpful assistant that takes in an array of pus
              }
            },
            {
              "type": "message",
              "role": "user",
              "content": {
                "type": "input_text",
                "text": "<push_notifications>{{item.notifications}}</push_notifications>"
              }
            }
          ]
        },
        "model": "o3-mini",
        "sampling_params": null
      },
      "error": null,
      "metadata": {}
    }
  ],
  "first_id": "evalrun_67e0c7d31560819090d60c0780591042",
  "last_id": "evalrun_67e0c7d31560819090d60c0780591042",
  "has_more": true
}
```


Get an eval run

```
GET https://api.openai.com/v1/evals/{eval_id}/runs/{run_id}
```

Get an evaluation run by ID.

Path parameters

eval_id string Required

The ID of the evaluation to retrieve runs for.

run_id string Required

The ID of the run to retrieve.

Returns

The EvalRun object matching the specified ID.

Example request curl

```
curl https://api.openai.com/v1/evals/eval_67abd54d9b0081909a86353f6fb9317a/runs/evalrun_67a
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
  "object": "eval.run",
  "id": "evalrun_67abd54d60ec8190832b46859da808f7",
  "eval_id": "eval_67abd54d9b0081909a86353f6fb9317a",
  "report_url": "https://platform.openai.com/evaluations/eval_67abd54d9b0081909a86353f6fb
  "status": "queued",
  "model": "gpt-4o-mini",
  "name": "gpt-4o-mini",
  "created_at": 1743092069,
  "result_counts": {
    "total": 0,
    "errored": 0,
    "failed": 0,
    "passed": 0
  },
  "per_model_usage": null,
  "per_testing_criteria_results": null,
  "data_source": {
    "type": "completions",
    "source": {
      "type": "file_content",
      "content": [
        {
          "item": {
            "input": "Tech Company Launches Advanced Artificial Intelligence Platform",
            "ground_truth": "Technology"
          }

```

```
29          },
30          {
31            "item": {
32              "input": "Central Bank Increases Interest Rates Amid Inflation Concerns",
33              "ground_truth": "Markets"
34            }
35          },
36          {
37            "item": {
38              "input": "International Summit Addresses Climate Change Strategies",
39              "ground_truth": "World"
40            }
41          },
42          {
43            "item": {
44              "input": "Major Retailer Reports Record-Breaking Holiday Sales",
45              "ground_truth": "Business"
46            }
47          },
48          {
49            "item": {
50              "input": "National Team Qualifies for World Championship Finals",
51              "ground_truth": "Sports"
52            }
53          },
54          {
55            "item": {
56              "input": "Stock Markets Rally After Positive Economic Data Released",
57              "ground_truth": "Markets"
58            }
59          },
60          {
61            "item": {
62              "input": "Global Manufacturer Announces Merger with Competitor",
63              "ground_truth": "Business"
64            }
65          },
66          {
67            "item": {
68              "input": "Breakthrough in Renewable Energy Technology Unveiled",
69              "ground_truth": "Technology"
70            }
71          },
72          {
73            "item": {
74              "input": "World Leaders Sign Historic Climate Agreement",
75              "ground_truth": "World"
76
```

```
          }
        },
        {
          "item": {
            "input": "Professional Athlete Sets New Record in Championship Event",
            "ground_truth": "Sports"
          }
        },
        {
          "item": {
            "input": "Financial Institutions Adapt to New Regulatory Requirements",
            "ground_truth": "Business"
          }
        },
        {
          "item": {
            "input": "Tech Conference Showcases Advances in Artificial Intelligence",
            "ground_truth": "Technology"
          }
        },
        {
          "item": {
            "input": "Global Markets Respond to Oil Price Fluctuations",
            "ground_truth": "Markets"
          }
        },
        {
          "item": {
            "input": "International Cooperation Strengthened Through New Treaty",
            "ground_truth": "World"
          }
        },
        {
          "item": {
            "input": "Sports League Announces Revised Schedule for Upcoming Season",
            "ground_truth": "Sports"
          }
        }
      ]
    },
    "input_messages": {
      "type": "template",
      "template": [
        {
          "type": "message",
          "role": "developer",
          "content": {
```

```
125             "type": "input_text",
126             "text": "Categorize a given news headline into one of the following topics: T
127           }
128         },
129         {
130           "type": "message",
131           "role": "user",
132           "content": {
133             "type": "input_text",
134             "text": "{{item.input}}"
135           }
136         }
137       ]
138     },
139     "model": "gpt-4o-mini",
140     "sampling_params": {
141       "seed": 42,
142       "temperature": 1.0,
143       "top_p": 1.0,
144       "max_completions_tokens": 2048
145     }
146   },
  "error": null,
  "metadata": {}
}
```

Create eval run

```
POST https://api.openai.com/v1/evals/{eval_id}/runs
```

Kicks off a new run for a given evaluation, specifying the data source, and what model configuration to use to test. The datasource will be validated against the schema specified in the config of the evaluation.

Path parameters

eval_id string Required
The ID of the evaluation to create a run for.

Request body

data_source object Required
Details about the run's data source.

Hide possible types

JsonlRunDataSource object
A JsonlRunDataSource object with that specifies a JSONL file that matches the eval

Hide properties

source object Required
Determines what populates the `item` namespace in the data source.

Hide possible types

EvalJsonlFileContentSource object

Hide properties

content array Required
The content of the jsonl file.

Hide properties

item object Required

sample object Optional

type string Required
The type of jsonl source. Always `file_content`.

EvalJsonlFileIdSource object

Hide properties

id string Required
The identifier of the file.

type string Required
The type of jsonl source. Always `file_id`.

type string Required
The type of data source. Always `jsonl`.

CompletionsRunDataSource object
A CompletionsRunDataSource object describing a model sampling configuration.

Hide properties

source object Required

Determines what populates the `item` namespace in this run's data source.

Hide possible types

EvalJsonlFileContentSource object

Hide properties

content array Required

The content of the jsonl file.

Hide properties

item object Required

sample object Optional

type string Required

The type of jsonl source. Always `file_content`.

EvalJsonlFileIdSource object

Hide properties

id string Required

The identifier of the file.

type string Required

The type of jsonl source. Always `file_id`.

StoredCompletionsRunDataSource object

A StoredCompletionsRunDataSource configuration describing a set of filters

Hide properties

type string Required

The type of source. Always `stored_completions`.

created_after integer or null Optional

An optional Unix timestamp to filter items created after this time.

created_before integer or null Optional

An optional Unix timestamp to filter items created before this time.

limit integer or null Optional
An optional maximum number of items to return.

metadata map Optional
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

model string or null Optional
An optional model to filter by (e.g., 'gpt-4o').

type string Required
The type of run data source. Always `completions`.

input_messages object Optional
Used when sampling from a model. Dictates the structure of the messages passed into the model. Can either be a reference to a prebuilt trajectory (ie, `item.input_trajectory`), or a template with variable references to the `item` namespace.

Hide possible types

TemplateInputMessages object

Hide properties

template array Required
A list of chat messages forming the prompt or context. May include variable references to the `item` namespace, ie {{item.name}}.

Hide possible types

Input message object
A message input to the model with a role indicating instruction following hierarchy. Instructions given with the `developer` or `system` role take precedence over instructions given with the `user` role. Messages with the `assistant` role are presumed to have been generated by the model in previous interactions.

Hide properties

content string or array Required
Text, image, or audio input to the model, used to generate a response. Can also contain previous assistant responses.

Hide possible types

Text input string
A text input to the model.

Input item content list array
A list of one or many input items to the model, containing different content types.

Hide possible types

Input text object
A text input to the model.

Hide properties

text string Required
The text input to the model.

type string Required
The type of the input item. Always `input_text`.

Input image object
An image input to the model. Learn about image inputs.

Hide properties

detail string Required
The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

type string Required
The type of the input item. Always `input_image`.

file_id string Optional
The ID of the file to be sent to the model.

image_url string Optional
The URL of the image to be sent to the model. A fully qualified URL or base64 encoded image in a data URL.

Input file object
A file input to the model.

Hide properties

type string Required
The type of the input item. Always `input_file`.

file_data string Optional
The content of the file to be sent to the model.

file_id string Optional
The ID of the file to be sent to the model.

file_url string Optional
The URL of the file to be sent to the model.

filename string Optional
The name of the file to be sent to the model.

role string Required
The role of the message input. One of `user`, `assistant`, `system`, or `developer`.

type string Optional
The type of the message input. Always `message`.

Eval message object object
A message input to the model with a role indicating instruction following hierarchy. Instructions given with the `developer` or `system` role take precedence over instructions given with the `user` role. Messages with the `assistant` role are presumed to have been generated by the model in previous interactions.

Hide properties

content string / object / array Required
Inputs to the model - can contain template strings.

Hide possible types

Text input string
A text input to the model.

Input text object
A text input to the model.

Hide properties

text string Required
The text input to the model.

type string Required
The type of the input item. Always `input_text` .

Output text object
A text output from the model.

^ Hide properties

text string Required
The text output from the model.

type string Required
The type of the output text. Always `output_text` .

Input image object
An image input to the model.

^ Hide properties

image_url string Required
The URL of the image input.

type string Required
The type of the image input. Always `input_image` .

detail string Optional
The detail level of the image to be sent to the model. One of `high` , `low` , or `auto` . Defaults to `auto` .

An array of Input text and Input image array
A list of inputs, each of which may be either an input text or input image object.

role string Required
The role of the message input. One of `user` , `assistant` , `system` , or `developer` .

type string Optional
The type of the message input. Always `message` .

type string Required
The type of input messages. Always `template`.

ItemReferenceInputMessages object

Hide properties

item_reference string Required
A reference to a variable in the `item` namespace. Ie, "item.input_trajectory"

type string Required
The type of input messages. Always `item_reference`.

model string Optional
The name of the model to use for generating completions (e.g. "o3-mini").

sampling_params object Optional

Hide properties

max_completion_tokens integer Optional
The maximum number of tokens in the generated output.

response_format object Optional
An object specifying the format that the model must output.

Setting to `{ "type": "json_schema", "json_schema": {...} }` enables Structured Outputs which ensures the model will match your supplied JSON schema. Learn more in the Structured Outputs guide.

Setting to `{ "type": "json_object" }` enables the older JSON mode, which ensures the message the model generates is valid JSON. Using `json_schema` is preferred for models that support it.

Hide possible types

Text object
Default response format. Used to generate text responses.

Hide properties

type string Required
The type of response format being defined. Always `text`.

JSON schema object

JSON Schema response format. Used to generate structured JSON responses. Learn more about Structured Outputs.

Hide properties

`json_schema` object Required
Structured Outputs configuration options, including a JSON Schema.

Hide properties

`name` string Required
The name of the response format. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

`description` string Optional
A description of what the response format is for, used by the model to determine how to respond in the format.

`schema` object Optional
The schema for the response format, described as a JSON Schema object. Learn how to build JSON schemas here.

`strict` boolean or null Optional Defaults to false
Whether to enable strict schema adherence when generating the output. If set to true, the model will always follow the exact schema defined in the `schema` field. Only a subset of JSON Schema is supported when `strict` is `true`. To learn more, read the Structured Outputs guide.

`type` string Required
The type of response format being defined. Always `json_schema`.

JSON object object
JSON object response format. An older method of generating JSON responses. Using `json_schema` is recommended for models that support it. Note that the model will not generate JSON without a system or user message instructing it to do so.

Hide properties

`type` string Required
The type of response format being defined. Always `json_object`.

`seed` integer Optional Defaults to 42
A seed value to initialize the randomness, during sampling.

temperature number Optional Defaults to 1
A higher temperature increases randomness in the outputs.

tools array Optional
A list of tools the model may call. Currently, only functions are supported as a tool. Use this to provide a list of functions the model may generate JSON inputs for. A max of 128 functions are supported.

Hide properties

function object Required

Hide properties

name string Required
The name of the function to be called. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

description string Optional
A description of what the function does, used by the model to choose when and how to call the function.

parameters object Optional
The parameters the functions accepts, described as a JSON Schema object. See the guide for examples, and the JSON Schema reference for documentation about the format.

Omitting `parameters` defines a function with an empty parameter list.

strict boolean or null Optional Defaults to false
Whether to enable strict schema adherence when generating the function call. If set to true, the model will follow the exact schema defined in the `parameters` field. Only a subset of JSON Schema is supported when `strict` is `true`. Learn more about Structured Outputs in the function calling guide.

type string Required
The type of the tool. Currently, only `function` is supported.

top_p number Optional Defaults to 1
An alternative to temperature for nucleus sampling; 1.0 includes all tokens.

ResponsesRunDataSource object
A ResponsesRunDataSource object describing a model sampling configuration.

Hide properties

source object Required

Determines what populates the `item` namespace in this run's data source.

Hide possible types

EvalJsonlFileContentSource object

Hide properties

content array Required

The content of the jsonl file.

Hide properties

item object Required

sample object Optional

type string Required

The type of jsonl source. Always `file_content`.

EvalJsonlFileIdSource object

Hide properties

id string Required

The identifier of the file.

type string Required

The type of jsonl source. Always `file_id`.

EvalResponsesSource object

A EvalResponsesSource object describing a run data source configuration.

Hide properties

type string Required

The type of run data source. Always `responses`.

created_after integer or null Optional

Only include items created after this timestamp (inclusive). This is a query parameter used to select responses.

created_before integer or null Optional

Only include items created before this timestamp (inclusive). This is a query parameter used to select responses.

`instructions_search` string or null Optional
Optional string to search the 'instructions' field. This is a query parameter used to select responses.

`metadata` object or null Optional
Metadata filter for the responses. This is a query parameter used to select responses.

`model` string or null Optional
The name of the model to find responses for. This is a query parameter used to select responses.

`reasoning_effort` string or null Optional Defaults to medium
o-series models only

Constrains effort on reasoning for reasoning models. Currently supported values are `low`, `medium`, and `high`. Reducing reasoning effort can result in faster responses and fewer tokens used on reasoning in a response.

`temperature` number or null Optional
Sampling temperature. This is a query parameter used to select responses.

`tools` array or null Optional
List of tool names. This is a query parameter used to select responses.

`top_p` number or null Optional
Nucleus sampling parameter. This is a query parameter used to select responses.

`users` array or null Optional
List of user identifiers. This is a query parameter used to select responses.

`type` string Required
The type of run data source. Always `responses`.

`input_messages` object Optional
Used when sampling from a model. Dictates the structure of the messages passed into the model. Can either be a reference to a prebuilt trajectory (ie, `item.input_trajectory`), or a template with variable references to the `item` namespace.

^ Hide possible types

InputMessagesTemplate object

Hide properties

template array Required

A list of chat messages forming the prompt or context. May include variable references to the `item` namespace, ie {{item.name}}.

Hide possible types

ChatMessage object

Hide properties

content string Required

The content of the message.

role string Required

The role of the message (e.g. "system", "assistant", "user").

Eval message object object

A message input to the model with a role indicating instruction following hierarchy. Instructions given with the `developer` or `system` role take precedence over instructions given with the `user` role. Messages with the `assistant` role are presumed to have been generated by the model in previous interactions.

Hide properties

content string / object / array Required

Inputs to the model - can contain template strings.

Hide possible types

Text input string

A text input to the model.

Input text object

A text input to the model.

Hide properties

text string Required

The text input to the model.

type string Required

The type of the input item. Always `input_text` .

Output text object

A text output from the model.

Hide properties

text string Required

The text output from the model.

type string Required

The type of the output text. Always `output_text`.

Input image object

An image input to the model.

Hide properties

image_url string Required

The URL of the image input.

type string Required

The type of the image input. Always `input_image`.

detail string Optional

The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

An array of Input text and Input image array

A list of inputs, each of which may be either an input text or input image object.

role string Required

The role of the message input. One of `user`, `assistant`, `system`, or `developer`.

type string Optional

The type of the message input. Always `message`.

type string Required

The type of input messages. Always `template`.

InputMessagesItemReference object

Hide properties

item_reference string Required
A reference to a variable in the `item` namespace. Ie, "item.name"

type string Required
The type of input messages. Always `item_reference` .

model string Optional
The name of the model to use for generating completions (e.g. "o3-mini").

sampling_params object Optional

Hide properties

max_completion_tokens integer Optional
The maximum number of tokens in the generated output.

seed integer Optional Defaults to 42
A seed value to initialize the randomness, during sampling.

temperature number Optional Defaults to 1
A higher temperature increases randomness in the outputs.

text object Optional
Configuration options for a text response from the model. Can be plain text or structured JSON data. Learn more:

Text inputs and outputs

Structured Outputs

Hide properties

format object Optional
An object specifying the format that the model must output.

Configuring `{ "type": "json_schema" }` enables Structured Outputs, which ensures the model will match your supplied JSON schema. Learn more in the Structured Outputs guide.

The default format is `{ "type": "text" }` with no additional options.

Not recommended for gpt-4o and newer models:

Setting to `{ "type": "json_object" }` enables the older JSON mode, which ensures the message the model generates is valid JSON. Using `json_schema` is preferred for models that support it.

Hide possible types

Text object
Default response format. Used to generate text responses.

Hide properties

type string Required
The type of response format being defined. Always `text`.

JSON schema object
JSON Schema response format. Used to generate structured JSON responses. Learn more about Structured Outputs.

Hide properties

name string Required
The name of the response format. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

schema object Required
The schema for the response format, described as a JSON Schema object. Learn how to build JSON schemas here.

type string Required
The type of response format being defined. Always `json_schema`.

description string Optional
A description of what the response format is for, used by the model to determine how to respond in the format.

strict boolean or null Optional Defaults to false
Whether to enable strict schema adherence when generating the output. If set to true, the model will always follow the exact schema defined in the `schema` field. Only a subset of JSON Schema is supported when `strict` is `true`. To learn more, read the Structured Outputs guide.

JSON object object
JSON object response format. An older method of generating JSON responses. Using `json_schema` is recommended for models that support it. Note that the model will not generate JSON without a system or user message instructing it to do so.

Hide properties

type string Required

The type of response format being defined. Always `json_object`.

---

tools array Optional

An array of tools the model may call while generating a response. You can specify which tool to use by setting the `tool_choice` parameter.

The two categories of tools you can provide the model are:

Built-in tools: Tools that are provided by OpenAI that extend the model's capabilities, like web search or file search. Learn more about built-in tools.

Function calls (custom tools): Functions that are defined by you, enabling the model to call your own code. Learn more about function calling.

Hide possible types

Function object

Defines a function in your own code the model can choose to call. Learn more about function calling.

Hide properties

name string Required

The name of the function to call.

parameters object Required

A JSON schema object describing the parameters of the function.

strict boolean Required

Whether to enforce strict parameter validation. Default `true`.

type string Required

The type of the function tool. Always `function`.

description string Optional

A description of the function. Used by the model to determine whether or not to call the function.

File search object

A tool that searches for relevant content from uploaded files. Learn more about the file search tool.

Hide properties

type string Required

The type of the file search tool. Always `file_search`.

`vector_store_ids` array Required
The IDs of the vector stores to search.

`filters` object Optional
A filter to apply.

Hide possible types

Comparison Filter object
A filter used to compare a specified attribute key to a given value using a defined comparison operation.

Hide properties

`key` string Required
The key to compare against the value.

`type` string Required
Specifies the comparison operator: `eq`, `ne`, `gt`, `gte`, `lt`, `lte`.

- `eq`: equals
- `ne`: not equal
- `gt`: greater than
- `gte`: greater than or equal
- `lt`: less than
- `lte`: less than or equal

`value` string / number / boolean Required
The value to compare against the attribute key; supports string, number, or boolean types.

Compound Filter object
Combine multiple filters using `and` or `or`.

Hide properties

`filters` array Required
Array of filters to combine. Items can be `ComparisonFilter` or `CompoundFilter`.

Hide possible types

Comparison Filter object

A filter used to compare a specified attribute key to a given value using a defined comparison operation.

Hide properties

key string Required

The key to compare against the value.

type string Required

Specifies the comparison operator: `eq`, `ne`, `gt`, `gte`, `lt`, `lte`.

- `eq` : equals
- `ne` : not equal
- `gt` : greater than
- `gte` : greater than or equal
- `lt` : less than
- `lte` : less than or equal

value string / number / boolean Required

The value to compare against the attribute key; supports string, number, or boolean types.

type string Required

Type of operation: `and` or `or`.

max_num_results integer Optional

The maximum number of results to return. This number should be between 1 and 50 inclusive.

ranking_options object Optional

Ranking options for search.

Hide properties

ranker string Optional

The ranker to use for the file search.

score_threshold number Optional

The score threshold for the file search, a number between 0 and 1. Numbers closer to 1 will attempt to return only the most relevant results, but may return fewer results.

Web search preview object

This tool searches the web for relevant results to use in a response. Learn more about the web search tool.

Hide properties

type string Required

The type of the web search tool. One of `web_search_preview` or `web_search_preview_2025_03_11`.

search_context_size string Optional

High level guidance for the amount of context window space to use for the search. One of `low`, `medium`, or `high`. `medium` is the default.

user_location object Optional

The user's location.

Hide properties

type string Required

The type of location approximation. Always `approximate`.

city string Optional

Free text input for the city of the user, e.g. `San Francisco`.

country string Optional

The two-letter ISO country code of the user, e.g. `US`.

region string Optional

Free text input for the region of the user, e.g. `California`.

timezone string Optional

The IANA timezone of the user, e.g. `America/Los_Angeles`.

Computer use preview object

A tool that controls a virtual computer. Learn more about the computer tool.

Hide properties

display_height integer Required

The height of the computer display.

display_width integer Required
The width of the computer display.

environment string Required
The type of computer environment to control.

type string Required
The type of the computer use tool. Always `computer_use_preview`.

MCP tool object
Give the model access to additional tools via remote Model Context Protocol (MCP) servers. Learn more about MCP.

Hide properties

server_label string Required
A label for this MCP server, used to identify it in tool calls.

server_url string Required
The URL for the MCP server.

type string Required
The type of the MCP tool. Always `mcp`.

allowed_tools array or object Optional
List of allowed tool names or a filter object.

Hide possible types

MCP allowed tools array
A string array of allowed tool names

MCP allowed tools filter object
A filter object to specify which tools are allowed.

Hide properties

tool_names array Optional
List of allowed tool names.

headers object or null Optional
Optional HTTP headers to send to the MCP server. Use for authentication or other

purposes.

require_approval object or string Optional Defaults to always

Specify which of the MCP server's tools require approval.

Hide possible types

MCP tool approval filter object

Hide properties

always object Optional

A list of tools that always require approval.

Hide properties

tool_names array Optional

List of tools that require approval.

never object Optional

A list of tools that never require approval.

Hide properties

tool_names array Optional

List of tools that do not require approval.

MCP tool approval setting string

Specify a single approval policy for all tools. One of `always` or `never`. When set to `always`, all tools will require approval. When set to `never`, all tools will not require approval.

server_description string Optional

Optional description of the MCP server, used to provide more context.

Code interpreter object

A tool that runs Python code to help generate a response to a prompt.

Hide properties

container string or object Required

The code interpreter container. Can be a container ID or an object that specifies uploaded file IDs to make available to your code.

Hide possible types

string
The container ID.

CodeInterpreterContainerAuto object
Configuration for a code interpreter container. Optionally specify the IDs of the files to run the code on.

Hide properties

type string Required
Always `auto`.

file_ids array Optional
An optional list of uploaded files to make available to your code.

type string Required
The type of the code interpreter tool. Always `code_interpreter`.

Image generation tool object
A tool that generates images using a model like `gpt-image-1`.

Hide properties

type string Required
The type of the image generation tool. Always `image_generation`.

background string Optional Defaults to auto
Background type for the generated image. One of `transparent`, `opaque`, or `auto`. Default: `auto`.

input_image_mask object Optional
Optional mask for inpainting. Contains `image_url` (string, optional) and `file_id` (string, optional).

Hide properties

file_id string Optional
File ID for the mask image.

image_url string Optional
Base64-encoded mask image.

model string Optional Defaults to gpt-image-1

The image generation model to use. Default: `gpt-image-1`.

moderation string Optional Defaults to auto
Moderation level for the generated image. Default: `auto`.

output_compression integer Optional Defaults to 100
Compression level for the output image. Default: 100.

output_format string Optional Defaults to png
The output format of the generated image. One of `png`, `webp`, or `jpeg`. Default: `png`.

partial_images integer Optional Defaults to 0
Number of partial images to generate in streaming mode, from 0 (default value) to 3.

quality string Optional Defaults to auto
The quality of the generated image. One of `low`, `medium`, `high`, or `auto`. Default: `auto`.

size string Optional Defaults to auto
The size of the generated image. One of `1024x1024`, `1024x1536`, `1536x1024`, or `auto`. Default: `auto`.

Local shell tool object
A tool that allows the model to execute shell commands in a local environment.

^ Hide properties

type string Required
The type of the local shell tool. Always `local_shell`.

top_p number Optional Defaults to 1
An alternative to temperature for nucleus sampling; 1.0 includes all tokens.

metadata map Optional
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

name string Optional
The name of the run.

Returns

The EvalRun object matching the specified ID.

Example request curl

```
curl https://api.openai.com/v1/evals/eval_67e579652b548190aaa83ada4b125f47/runs \
  -X POST \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{"name":"gpt-4o-mini","data_source":{"type":"completions","input_messages":{"type":"t
```

Response

```
{
  "object": "eval.run",
  "id": "evalrun_67e57965b480819094274e3a32235e4c",
  "eval_id": "eval_67e579652b548190aaa83ada4b125f47",
  "report_url": "https://platform.openai.com/evaluations/eval_67e579652b548190aaa83ada4b12
  "status": "queued",
  "model": "gpt-4o-mini",
  "name": "gpt-4o-mini",
  "created_at": 1743092069,
  "result_counts": {
    "total": 0,
    "errored": 0,
    "failed": 0,
    "passed": 0
  },
  "per_model_usage": null,
  "per_testing_criteria_results": null,
  "data_source": {
    "type": "completions",
    "source": {
      "type": "file_content",
      "content": [
        {
          "item": {
            "input": "Tech Company Launches Advanced Artificial Intelligence Platform",

```

```
            "ground_truth": "Technology"
          }
        }
      ]
    },
    "input_messages": {
      "type": "template",
      "template": [
        {
          "type": "message",
          "role": "developer",
          "content": {
            "type": "input_text",
            "text": "Categorize a given news headline into one of the following topics: Te
          }
        },
        {
          "type": "message",
          "role": "user",
          "content": {
            "type": "input_text",
            "text": "{{item.input}}"
          }
        }
      ]
    },
    "model": "gpt-4o-mini",
    "sampling_params": {
      "seed": 42,
      "temperature": 1.0,
      "top_p": 1.0,
      "max_completions_tokens": 2048
    }
  },
  "error": null,
  "metadata": {}
}
```

Cancel eval run

```
POST https://api.openai.com/v1/evals/{eval_id}/runs/{run_id}
```

Cancel an ongoing evaluation run.

Path parameters

eval_id string Required

The ID of the evaluation whose run you want to cancel.

run_id string Required

The ID of the run to cancel.

Returns

The updated EvalRun object reflecting that the run is canceled.

Example request curl

```curl
curl https://api.openai.com/v1/evals/eval_67abd54d9b0081909a86353f6fb9317a/runs/evalrun_67a
  -X POST \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
  "object": "eval.run",
  "id": "evalrun_67abd54d60ec8190832b46859da808f7",
  "eval_id": "eval_67abd54d9b0081909a86353f6fb9317a",
  "report_url": "https://platform.openai.com/evaluations/eval_67abd54d9b0081909a86353f6fb
  "status": "canceled",
  "model": "gpt-4o-mini",
  "name": "gpt-4o-mini",
  "created_at": 1743092069,
  "result_counts": {
    "total": 0,
    "errored": 0,
    "failed": 0,
    "passed": 0
  },
  "per_model_usage": null,
  "per_testing_criteria_results": null,
  "data_source": {
    "type": "completions",
    "source": {
      "type": "file_content",
```

```
22        "content": [
23          {
24            "item": {
25              "input": "Tech Company Launches Advanced Artificial Intelligence Platform",
26              "ground_truth": "Technology"
27            }
28          },
29          {
30            "item": {
31              "input": "Central Bank Increases Interest Rates Amid Inflation Concerns",
32              "ground_truth": "Markets"
33            }
34          },
35          {
36            "item": {
37              "input": "International Summit Addresses Climate Change Strategies",
38              "ground_truth": "World"
39            }
40          },
41          {
42            "item": {
43              "input": "Major Retailer Reports Record-Breaking Holiday Sales",
44              "ground_truth": "Business"
45            }
46          },
47          {
48            "item": {
49              "input": "National Team Qualifies for World Championship Finals",
50              "ground_truth": "Sports"
51            }
52          },
53          {
54            "item": {
55              "input": "Stock Markets Rally After Positive Economic Data Released",
56              "ground_truth": "Markets"
57            }
58          },
59          {
60            "item": {
61              "input": "Global Manufacturer Announces Merger with Competitor",
62              "ground_truth": "Business"
63            }
64          },
65          {
66            "item": {
67              "input": "Breakthrough in Renewable Energy Technology Unveiled",
68              "ground_truth": "Technology"
69
```

```
          }
        },
        {
          "item": {
            "input": "World Leaders Sign Historic Climate Agreement",
            "ground_truth": "World"
          }
        },
        {
          "item": {
            "input": "Professional Athlete Sets New Record in Championship Event",
            "ground_truth": "Sports"
          }
        },
        {
          "item": {
            "input": "Financial Institutions Adapt to New Regulatory Requirements",
            "ground_truth": "Business"
          }
        },
        {
          "item": {
            "input": "Tech Conference Showcases Advances in Artificial Intelligence",
            "ground_truth": "Technology"
          }
        },
        {
          "item": {
            "input": "Global Markets Respond to Oil Price Fluctuations",
            "ground_truth": "Markets"
          }
        },
        {
          "item": {
            "input": "International Cooperation Strengthened Through New Treaty",
            "ground_truth": "World"
          }
        },
        {
          "item": {
            "input": "Sports League Announces Revised Schedule for Upcoming Season",
            "ground_truth": "Sports"
          }
        }
      ]
    },
    "input_messages": {
```

```
      "type": "template",
      "template": [
        {
          "type": "message",
          "role": "developer",
          "content": {
            "type": "input_text",
            "text": "Categorize a given news headline into one of the following topics: T
          }
        },
        {
          "type": "message",
          "role": "user",
          "content": {
            "type": "input_text",
            "text": "{{item.input}}"
          }
        }
      ]
    },
    "model": "gpt-4o-mini",
    "sampling_params": {
      "seed": 42,
      "temperature": 1.0,
      "top_p": 1.0,
      "max_completions_tokens": 2048
    }
  },
  "error": null,
"metadata": {}
}
```

Delete eval run

```
DELETE https://api.openai.com/v1/evals/{eval_id}/runs/{run_id}
```

Delete an eval run.

Path parameters

eval_id string Required

The ID of the evaluation to delete the run from.

run_id string Required
The ID of the run to delete.

Returns

An object containing the status of the delete operation.

Example request curl

```
curl https://api.openai.com/v1/evals/eval_123abc/runs/evalrun_abc456 \
  -X DELETE \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
  "object": "eval.run.deleted",
  "deleted": true,
  "run_id": "evalrun_abc456"
}
```

Get an output item of an eval run

```
GET https://api.openai.com/v1/evals/{eval_id}/runs/{run_id}/output_items/{output_item_id}
```

Get an evaluation run output item by ID.

Path parameters

eval_id string Required
The ID of the evaluation to retrieve runs for.

output_item_id string Required
The ID of the output item to retrieve.

run_id string Required

The ID of the run to retrieve.

Returns

The EvalRunOutputItem object matching the specified ID.

Example request curl

```
curl https://api.openai.com/v1/evals/eval_67abd54d9b0081909a86353f6fb9317a/runs/evalrun_67a
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
  "object": "eval.run.output_item",
  "id": "outputitem_67e5796c28e081909917bf79f6e6214d",
  "created_at": 1743092076,
  "run_id": "evalrun_67abd54d60ec8190832b46859da808f7",
  "eval_id": "eval_67abd54d9b0081909a86353f6fb9317a",
  "status": "pass",
  "datasource_item_id": 5,
  "datasource_item": {
    "input": "Stock Markets Rally After Positive Economic Data Released",
    "ground_truth": "Markets"
  },
  "results": [
    {
      "name": "String check-a2486074-d803-4445-b431-ad2262e85d47",
      "sample": null,
      "passed": true,
      "score": 1.0
    }
  ],
  "sample": {
    "input": [
      {
        "role": "developer",
        "content": "Categorize a given news headline into one of the following topics: Tec
        "tool_call_id": null,
        "tool_calls": null,
        "function_call": null
      },
```

```
      {
        "role": "user",
        "content": "Stock Markets Rally After Positive Economic Data Released",
        "tool_call_id": null,
        "tool_calls": null,
        "function_call": null
      }
    ],
    "output": [
      {
        "role": "assistant",
        "content": "Markets",
        "tool_call_id": null,
        "tool_calls": null,
        "function_call": null
      }
    ],
    "finish_reason": "stop",
    "model": "gpt-4o-mini-2024-07-18",
    "usage": {
      "total_tokens": 325,
      "completion_tokens": 2,
      "prompt_tokens": 323,
      "cached_tokens": 0
    },
    "error": null,
    "temperature": 1.0,
    "max_completion_tokens": 2048,
    "top_p": 1.0,
    "seed": 42
  }
}
```

Get eval run output items

```
GET https://api.openai.com/v1/evals/{eval_id}/runs/{run_id}/output_items
```

Get a list of output items for an evaluation run.

Path parameters

eval_id string Required

The ID of the evaluation to retrieve runs for.

run_id string Required
The ID of the run to retrieve output items for.

Query parameters

after string Optional
Identifier for the last output item from the previous pagination request.

limit integer Optional Defaults to 20
Number of output items to retrieve.

order string Optional Defaults to asc
Sort order for output items by timestamp. Use `asc` for ascending order or `desc` for descending order. Defaults to `asc`.

status string Optional
Filter output items by status. Use `failed` to filter by failed output items or `pass` to filter by passed output items.

Returns

A list of EvalRunOutputItem objects matching the specified ID.

Example request curl

```
curl https://api.openai.com/v1/evals/egroup_67abd54d9b0081909a86353f6fb9317a/runs/erun_67ab
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
  "object": "list",
  "data": [
    {
      "object": "eval.run.output_item",
      "id": "outputitem_67e5796c28e081909917bf79f6e6214d",
```

```
      "created_at": 1743092076,
      "run_id": "evalrun_67abd54d60ec8190832b46859da808f7",
      "eval_id": "eval_67abd54d9b0081909a86353f6fb9317a",
      "status": "pass",
      "datasource_item_id": 5,
      "datasource_item": {
        "input": "Stock Markets Rally After Positive Economic Data Released",
        "ground_truth": "Markets"
      },
      "results": [
        {
          "name": "String check-a2486074-d803-4445-b431-ad2262e85d47",
          "sample": null,
          "passed": true,
          "score": 1.0
        }
      ],
      "sample": {
        "input": [
          {
            "role": "developer",
            "content": "Categorize a given news headline into one of the following topics:
            "tool_call_id": null,
            "tool_calls": null,
            "function_call": null
          },
          {
            "role": "user",
            "content": "Stock Markets Rally After Positive Economic Data Released",
            "tool_call_id": null,
            "tool_calls": null,
            "function_call": null
          }
        ],
        "output": [
          {
            "role": "assistant",
            "content": "Markets",
            "tool_call_id": null,
            "tool_calls": null,
            "function_call": null
          }
        ],
        "finish_reason": "stop",
        "model": "gpt-4o-mini-2024-07-18",
        "usage": {
          "total_tokens": 325,
```

```
55          "completion_tokens": 2,
56          "prompt_tokens": 323,
57          "cached_tokens": 0
58        },
59        "error": null,
60        "temperature": 1.0,
61        "max_completion_tokens": 2048,
62        "top_p": 1.0,
63        "seed": 42
64      }
65    }
66  ],
67  "first_id": "outputitem_67e5796c28e081909917bf79f6e6214d",
68  "last_id": "outputitem_67e5796c28e081909917bf79f6e6214d",
69  "has_more": true
}
```

The eval object

An Eval object with a data source config and testing criteria. An Eval represents a task to be done for your LLM integration. Like:

- Improve the quality of my chatbot
- See how well my chatbot handles customer support
- Check if o4-mini is better at my usecase than gpt-4o

`created_at` integer
The Unix timestamp (in seconds) for when the eval was created.

`data_source_config` object
Configuration of data sources used in runs of the evaluation.

Hide possible types

CustomDataSourceConfig object
A CustomDataSourceConfig which specifies the schema of your `item` and optionally `sample` namespaces. The response schema defines the shape of the data that will be:

- Used to define your testing criteria and
- What data is required when creating a run

Hide properties

schema object
The json schema for the run data source items. Learn how to build JSON schemas here.

type string
The type of data source. Always `custom`.

LogsDataSourceConfig object
A LogsDataSourceConfig which specifies the metadata property of your logs query. This is usually metadata like `usecase=chatbot` or `prompt-version=v2`, etc. The schema returned by this data source config is used to defined what variables are available in your evals. `item` and `sample` are both defined when using this data source config.

Hide properties

schema object
The json schema for the run data source items. Learn how to build JSON schemas here.

type string
The type of data source. Always `logs`.

metadata map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

StoredCompletionsDataSourceConfig object Deprecated
Deprecated in favor of LogsDataSourceConfig.

Hide properties

schema object
The json schema for the run data source items. Learn how to build JSON schemas here.

type string
The type of data source. Always `stored_completions`.

metadata map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

---

id string
Unique identifier for the evaluation.

---

metadata map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

---

name string
The name of the evaluation.

---

object string
The object type.

---

testing_criteria array
A list of testing criteria.

Hide possible types

LabelModelGrader object
A LabelModelGrader object which uses a model to assign labels to each item in the evaluation.

Hide properties

input array

Hide properties

content string / object / array
Inputs to the model - can contain template strings.

Hide possible types

Text input string
A text input to the model.

Input text object
A text input to the model.

Hide properties

text string

The text input to the model.

type string
The type of the input item. Always `input_text` .

Output text object
A text output from the model.

^ Hide properties

text string
The text output from the model.

type string
The type of the output text. Always `output_text` .

Input image object
An image input to the model.

^ Hide properties

image_url string
The URL of the image input.

type string
The type of the image input. Always `input_image` .

detail string
The detail level of the image to be sent to the model. One of `high` , `low` , or `auto` . Defaults to `auto` .

An array of Input text and Input image array
A list of inputs, each of which may be either an input text or input image object.

role string
The role of the message input. One of `user` , `assistant` , `system` , or `developer` .

type string
The type of the message input. Always `message` .

labels array
The labels to assign to each item in the evaluation.

model string
The model to use for the evaluation. Must support structured outputs.

name string
The name of the grader.

passing_labels array
The labels that indicate a passing result. Must be a subset of labels.

type string
The object type, which is always `label_model`.

StringCheckGrader object
A StringCheckGrader object that performs a string comparison between input and reference using a specified operation.

Hide properties

input string
The input text. This may include template strings.

name string
The name of the grader.

operation string
The string check operation to perform. One of `eq`, `ne`, `like`, or `ilike`.

reference string
The reference text. This may include template strings.

type string
The object type, which is always `string_check`.

TextSimilarityGrader object
A TextSimilarityGrader object which grades text based on similarity metrics.

Hide properties

evaluation_metric string
The evaluation metric to use. One of `fuzzy_match`, `bleu`, `gleu`, `meteor`, `rouge_1`,

`rouge_2`, `rouge_3`, `rouge_4`, `rouge_5`, or `rouge_l`.

input string
The text being graded.

name string
The name of the grader.

pass_threshold number
The threshold for the score.

reference string
The text being graded against.

type string
The type of grader.

PythonGrader object
A PythonGrader object that runs a python script on the input.

Hide properties

name string
The name of the grader.

source string
The source code of the python script.

type string
The object type, which is always `python`.

image_tag string
The image tag to use for the python script.

pass_threshold number
The threshold for the score.

ScoreModelGrader object
A ScoreModelGrader object that uses a model to assign a score to the input.

Hide properties

input array
The input text. This may include template strings.

Hide properties

content string / object / array

Inputs to the model - can contain template strings.

Hide possible types

Text input string

A text input to the model.

Input text object

A text input to the model.

Hide properties

text string

The text input to the model.

type string

The type of the input item. Always `input_text` .

Output text object

A text output from the model.

Hide properties

text string

The text output from the model.

type string

The type of the output text. Always `output_text` .

Input image object

An image input to the model.

Hide properties

image_url string

The URL of the image input.

type string

The type of the image input. Always `input_image` .

detail string
The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

An array of Input text and Input image array
A list of inputs, each of which may be either an input text or input image object.

role string
The role of the message input. One of `user`, `assistant`, `system`, or `developer`.

type string
The type of the message input. Always `message`.

model string
The model to use for the evaluation.

name string
The name of the grader.

type string
The object type, which is always `score_model`.

pass_threshold number
The threshold for the score.

range array
The range of the score. Defaults to `[0, 1]`.

sampling_params object
The sampling parameters for the model.

OBJECT The eval object

```
{
  "object": "eval",
  "id": "eval_67abd54d9b0081909a86353f6fb9317a",
  "data_source_config": {
    "type": "custom",
    "item_schema": {
      "type": "object",
```

```
      "properties": {
        "label": {"type": "string"},
      },
      "required": ["label"]
    },
    "include_sample_schema": true
  },
  "testing_criteria": [
    {
      "name": "My string check grader",
      "type": "string_check",
      "input": "{{sample.output_text}}",
      "reference": "{{item.label}}",
      "operation": "eq",
    }
  ],
  "name": "External Data Eval",
  "created_at": 1739314509,
  "metadata": {
    "test": "synthetics",
  }
}
```

The eval run object

A schema representing an evaluation run.

`created_at` integer
Unix timestamp (in seconds) when the evaluation run was created.

`data_source` object
Information about the run's data source.

Hide possible types

JsonlRunDataSource object
A JsonlRunDataSource object with that specifies a JSONL file that matches the eval

Hide properties

`source` object
Determines what populates the `item` namespace in the data source.

Hide possible types

EvalJsonlFileContentSource object

Hide properties

content array

The content of the jsonl file.

Hide properties

item object

sample object

type string

The type of jsonl source. Always `file_content`.

EvalJsonlFileIdSource object

Hide properties

id string

The identifier of the file.

type string

The type of jsonl source. Always `file_id`.

type string

The type of data source. Always `jsonl`.

CompletionsRunDataSource object

A CompletionsRunDataSource object describing a model sampling configuration.

Hide properties

source object

Determines what populates the `item` namespace in this run's data source.

Hide possible types

EvalJsonlFileContentSource object

Hide properties

content array

The content of the jsonl file.

Hide properties

item object

sample object

type string
The type of jsonl source. Always `file_content`.

EvalJsonlFileIdSource object

Hide properties

id string
The identifier of the file.

type string
The type of jsonl source. Always `file_id`.

StoredCompletionsRunDataSource object
A StoredCompletionsRunDataSource configuration describing a set of filters

Hide properties

type string
The type of source. Always `stored_completions`.

created_after integer or null
An optional Unix timestamp to filter items created after this time.

created_before integer or null
An optional Unix timestamp to filter items created before this time.

limit integer or null
An optional maximum number of items to return.

metadata map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

model string or null

An optional model to filter by (e.g., 'gpt-4o').

type string
The type of run data source. Always `completions`.

input_messages object
Used when sampling from a model. Dictates the structure of the messages passed into the model. Can either be a reference to a prebuilt trajectory (ie, `item.input_trajectory`), or a template with variable references to the `item` namespace.

Hide possible types

TemplateInputMessages object

Hide properties

template array
A list of chat messages forming the prompt or context. May include variable references to the `item` namespace, ie {{item.name}}.

Hide possible types

Input message object
A message input to the model with a role indicating instruction following hierarchy. Instructions given with the `developer` or `system` role take precedence over instructions given with the `user` role. Messages with the `assistant` role are presumed to have been generated by the model in previous interactions.

Hide properties

content string or array
Text, image, or audio input to the model, used to generate a response. Can also contain previous assistant responses.

Hide possible types

Text input string
A text input to the model.

Input item content list array
A list of one or many input items to the model, containing different content types.

Hide possible types

Input text object
A text input to the model.

Hide properties

text string
The text input to the model.

type string
The type of the input item. Always `input_text`.

Input image object
An image input to the model. Learn about image inputs.

Hide properties

detail string
The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

type string
The type of the input item. Always `input_image`.

file_id string
The ID of the file to be sent to the model.

image_url string
The URL of the image to be sent to the model. A fully qualified URL or base64 encoded image in a data URL.

Input file object
A file input to the model.

Hide properties

type string
The type of the input item. Always `input_file`.

file_data string
The content of the file to be sent to the model.

file_id string
The ID of the file to be sent to the model.

file_url string
The URL of the file to be sent to the model.

filename string

The name of the file to be sent to the model.

role string
The role of the message input. One of `user`, `assistant`, `system`, or `developer`.

type string
The type of the message input. Always `message`.

Eval message object object
A message input to the model with a role indicating instruction following hierarchy. Instructions given with the `developer` or `system` role take precedence over instructions given with the `user` role. Messages with the `assistant` role are presumed to have been generated by the model in previous interactions.

Hide properties

content string / object / array
Inputs to the model - can contain template strings.

Hide possible types

Text input string
A text input to the model.

Input text object
A text input to the model.

Hide properties

text string
The text input to the model.

type string
The type of the input item. Always `input_text`.

Output text object
A text output from the model.

Hide properties

text string
The text output from the model.

type string
The type of the output text. Always `output_text` .

Input image object
An image input to the model.

Hide properties

image_url string
The URL of the image input.

type string
The type of the image input. Always `input_image` .

detail string
The detail level of the image to be sent to the model. One of `high` , `low` , or `auto` . Defaults to `auto` .

An array of Input text and Input image array
A list of inputs, each of which may be either an input text or input image object.

role string
The role of the message input. One of `user` , `assistant` , `system` , or `developer` .

type string
The type of the message input. Always `message` .

type string
The type of input messages. Always `template` .

ItemReferenceInputMessages object

Hide properties

item_reference string
A reference to a variable in the `item` namespace. Ie, "item.input_trajectory"

type string
The type of input messages. Always `item_reference` .

model string
The name of the model to use for generating completions (e.g. "o3-mini").

sampling_params object

Hide properties

max_completion_tokens integer
The maximum number of tokens in the generated output.

response_format object
An object specifying the format that the model must output.

Setting to `{ "type": "json_schema", "json_schema": {...} }` enables Structured Outputs which ensures the model will match your supplied JSON schema. Learn more in the Structured Outputs guide.

Setting to `{ "type": "json_object" }` enables the older JSON mode, which ensures the message the model generates is valid JSON. Using `json_schema` is preferred for models that support it.

Hide possible types

Text object
Default response format. Used to generate text responses.

Hide properties

type string
The type of response format being defined. Always `text`.

JSON schema object
JSON Schema response format. Used to generate structured JSON responses. Learn more about Structured Outputs.

Hide properties

json_schema object
Structured Outputs configuration options, including a JSON Schema.

Hide properties

name string
The name of the response format. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

description string
A description of what the response format is for, used by the model to determine how

to respond in the format.

schema object
The schema for the response format, described as a JSON Schema object. Learn how to build JSON schemas here.

strict boolean or null
Whether to enable strict schema adherence when generating the output. If set to true, the model will always follow the exact schema defined in the `schema` field. Only a subset of JSON Schema is supported when `strict` is `true`. To learn more, read the Structured Outputs guide.

type string
The type of response format being defined. Always `json_schema`.

JSON object object
JSON object response format. An older method of generating JSON responses. Using `json_schema` is recommended for models that support it. Note that the model will not generate JSON without a system or user message instructing it to do so.

Hide properties

type string
The type of response format being defined. Always `json_object`.

seed integer
A seed value to initialize the randomness, during sampling.

temperature number
A higher temperature increases randomness in the outputs.

tools array
A list of tools the model may call. Currently, only functions are supported as a tool. Use this to provide a list of functions the model may generate JSON inputs for. A max of 128 functions are supported.

Hide properties

function object

Hide properties

name string
The name of the function to be called. Must be a-z, A-Z, 0-9, or contain underscores and

dashes, with a maximum length of 64.

description string
A description of what the function does, used by the model to choose when and how to call the function.

parameters object
The parameters the functions accepts, described as a JSON Schema object. See the guide for examples, and the JSON Schema reference for documentation about the format.

Omitting `parameters` defines a function with an empty parameter list.

strict boolean or null
Whether to enable strict schema adherence when generating the function call. If set to true, the model will follow the exact schema defined in the `parameters` field. Only a subset of JSON Schema is supported when `strict` is `true`. Learn more about Structured Outputs in the function calling guide.

type string
The type of the tool. Currently, only `function` is supported.

top_p number
An alternative to temperature for nucleus sampling; 1.0 includes all tokens.

ResponsesRunDataSource object
A ResponsesRunDataSource object describing a model sampling configuration.

Hide properties

source object
Determines what populates the `item` namespace in this run's data source.

Hide possible types

EvalJsonlFileContentSource object

Hide properties

content array
The content of the jsonl file.

Hide properties

item object

sample object

type string
The type of jsonl source. Always `file_content`.

EvalJsonlFileIdSource object

Hide properties

id string
The identifier of the file.

type string
The type of jsonl source. Always `file_id`.

EvalResponsesSource object
A EvalResponsesSource object describing a run data source configuration.

Hide properties

type string
The type of run data source. Always `responses`.

created_after integer or null
Only include items created after this timestamp (inclusive). This is a query parameter used to select responses.

created_before integer or null
Only include items created before this timestamp (inclusive). This is a query parameter used to select responses.

instructions_search string or null
Optional string to search the 'instructions' field. This is a query parameter used to select responses.

metadata object or null
Metadata filter for the responses. This is a query parameter used to select responses.

model string or null
The name of the model to find responses for. This is a query parameter used to select responses.

reasoning_effort string or null

o-series models only

Constrains effort on reasoning for reasoning models. Currently supported values are `low`, `medium`, and `high`. Reducing reasoning effort can result in faster responses and fewer tokens used on reasoning in a response.

`temperature` number or null
Sampling temperature. This is a query parameter used to select responses.

`tools` array or null
List of tool names. This is a query parameter used to select responses.

`top_p` number or null
Nucleus sampling parameter. This is a query parameter used to select responses.

`users` array or null
List of user identifiers. This is a query parameter used to select responses.

`type` string
The type of run data source. Always `responses`.

`input_messages` object
Used when sampling from a model. Dictates the structure of the messages passed into the model. Can either be a reference to a prebuilt trajectory (ie, `item.input_trajectory`), or a template with variable references to the `item` namespace.

Hide possible types

InputMessagesTemplate object

Hide properties

`template` array
A list of chat messages forming the prompt or context. May include variable references to the `item` namespace, ie {{item.name}}.

Hide possible types

ChatMessage object

Hide properties

`content` string
The content of the message.

`role` string

The role of the message (e.g. "system", "assistant", "user").

Eval message object object

A message input to the model with a role indicating instruction following hierarchy. Instructions given with the `developer` or `system` role take precedence over instructions given with the `user` role. Messages with the `assistant` role are presumed to have been generated by the model in previous interactions.

Hide properties

content string / object / array

Inputs to the model - can contain template strings.

Hide possible types

Text input string

A text input to the model.

Input text object

A text input to the model.

Hide properties

text string

The text input to the model.

type string

The type of the input item. Always `input_text` .

Output text object

A text output from the model.

Hide properties

text string

The text output from the model.

type string

The type of the output text. Always `output_text` .

Input image object

An image input to the model.

Hide properties

image_url string

The URL of the image input.

type string
The type of the image input. Always `input_image`.

detail string
The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

An array of Input text and Input image array
A list of inputs, each of which may be either an input text or input image object.

role string
The role of the message input. One of `user`, `assistant`, `system`, or `developer`.

type string
The type of the message input. Always `message`.

type string
The type of input messages. Always `template`.

InputMessagesItemReference object

Hide properties

item_reference string
A reference to a variable in the `item` namespace. Ie, "item.name"

type string
The type of input messages. Always `item_reference`.

model string
The name of the model to use for generating completions (e.g. "o3-mini").

sampling_params object

Hide properties

max_completion_tokens integer
The maximum number of tokens in the generated output.

seed integer
A seed value to initialize the randomness, during sampling.

temperature number
A higher temperature increases randomness in the outputs.

text object
Configuration options for a text response from the model. Can be plain text or structured JSON data. Learn more:

Text inputs and outputs

Structured Outputs

Hide properties

format object
An object specifying the format that the model must output.

Configuring `{ "type": "json_schema" }` enables Structured Outputs, which ensures the model will match your supplied JSON schema. Learn more in the Structured Outputs guide.

The default format is `{ "type": "text" }` with no additional options.

Not recommended for gpt-4o and newer models:

Setting to `{ "type": "json_object" }` enables the older JSON mode, which ensures the message the model generates is valid JSON. Using `json_schema` is preferred for models that support it.

Hide possible types

Text object
Default response format. Used to generate text responses.

Hide properties

type string
The type of response format being defined. Always `text` .

JSON schema object
JSON Schema response format. Used to generate structured JSON responses. Learn more about Structured Outputs.

Hide properties

name string
The name of the response format. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

schema object
The schema for the response format, described as a JSON Schema object. Learn how to build JSON schemas here.

type string
The type of response format being defined. Always `json_schema`.

description string
A description of what the response format is for, used by the model to determine how to respond in the format.

strict boolean or null
Whether to enable strict schema adherence when generating the output. If set to true, the model will always follow the exact schema defined in the `schema` field. Only a subset of JSON Schema is supported when `strict` is `true`. To learn more, read the Structured Outputs guide.

JSON object object
JSON object response format. An older method of generating JSON responses. Using `json_schema` is recommended for models that support it. Note that the model will not generate JSON without a system or user message instructing it to do so.

Hide properties

type string
The type of response format being defined. Always `json_object`.

tools array
An array of tools the model may call while generating a response. You can specify which tool to use by setting the `tool_choice` parameter.

The two categories of tools you can provide the model are:

Built-in tools: Tools that are provided by OpenAI that extend the model's capabilities, like web search or file search. Learn more about built-in tools.

Function calls (custom tools): Functions that are defined by you, enabling the model to call your own code. Learn more about function calling.

Hide possible types

Function object

Defines a function in your own code the model can choose to call. Learn more about function calling.

Hide properties

name string

The name of the function to call.

parameters object

A JSON schema object describing the parameters of the function.

strict boolean

Whether to enforce strict parameter validation. Default `true`.

type string

The type of the function tool. Always `function`.

description string

A description of the function. Used by the model to determine whether or not to call the function.

File search object

A tool that searches for relevant content from uploaded files. Learn more about the file search tool.

Hide properties

type string

The type of the file search tool. Always `file_search`.

vector_store_ids array

The IDs of the vector stores to search.

filters object

A filter to apply.

Hide possible types

Comparison Filter object

A filter used to compare a specified attribute key to a given value using a defined comparison operation.

Hide properties

key string
The key to compare against the value.

type string
Specifies the comparison operator: `eq`, `ne`, `gt`, `gte`, `lt`, `lte`.

- `eq`: equals
- `ne`: not equal
- `gt`: greater than
- `gte`: greater than or equal
- `lt`: less than
- `lte`: less than or equal

value string / number / boolean
The value to compare against the attribute key; supports string, number, or boolean types.

Compound Filter object
Combine multiple filters using `and` or `or`.

Hide properties

filters array
Array of filters to combine. Items can be `ComparisonFilter` or `CompoundFilter`.

Hide possible types

Comparison Filter object
A filter used to compare a specified attribute key to a given value using a defined comparison operation.

Hide properties

key string
The key to compare against the value.

type string
Specifies the comparison operator: `eq`, `ne`, `gt`, `gte`, `lt`, `lte`.

- `eq`: equals
- `ne`: not equal
- `gt`: greater than

`gte` : greater than or equal

`lt` : less than

`lte` : less than or equal

`value` string / number / boolean
The value to compare against the attribute key; supports string, number, or boolean types.

`type` string
Type of operation: `and` or `or` .

`max_num_results` integer
The maximum number of results to return. This number should be between 1 and 50 inclusive.

`ranking_options` object
Ranking options for search.

Hide properties

`ranker` string
The ranker to use for the file search.

`score_threshold` number
The score threshold for the file search, a number between 0 and 1. Numbers closer to 1 will attempt to return only the most relevant results, but may return fewer results.

Web search preview object
This tool searches the web for relevant results to use in a response. Learn more about the web search tool.

Hide properties

`type` string
The type of the web search tool. One of `web_search_preview` or `web_search_preview_2025_03_11` .

`search_context_size` string
High level guidance for the amount of context window space to use for the search. One

of `low`, `medium`, or `high`. `medium` is the default.

`user_location` object
The user's location.

Hide properties

`type` string
The type of location approximation. Always `approximate`.

`city` string
Free text input for the city of the user, e.g. `San Francisco`.

`country` string
The two-letter ISO country code of the user, e.g. `US`.

`region` string
Free text input for the region of the user, e.g. `California`.

`timezone` string
The IANA timezone of the user, e.g. `America/Los_Angeles`.

Computer use preview object
A tool that controls a virtual computer. Learn more about the computer tool.

Hide properties

`display_height` integer
The height of the computer display.

`display_width` integer
The width of the computer display.

`environment` string
The type of computer environment to control.

`type` string
The type of the computer use tool. Always `computer_use_preview`.

MCP tool object
Give the model access to additional tools via remote Model Context Protocol (MCP) servers. Learn more about MCP.

Hide properties

server_label string
A label for this MCP server, used to identify it in tool calls.

server_url string
The URL for the MCP server.

type string
The type of the MCP tool. Always `mcp` .

allowed_tools array or object
List of allowed tool names or a filter object.

Hide possible types

MCP allowed tools array
A string array of allowed tool names

MCP allowed tools filter object
A filter object to specify which tools are allowed.

Hide properties

tool_names array
List of allowed tool names.

headers object or null
Optional HTTP headers to send to the MCP server. Use for authentication or other purposes.

require_approval object or string
Specify which of the MCP server's tools require approval.

Hide possible types

MCP tool approval filter object

Hide properties

always object
A list of tools that always require approval.

Hide properties

tool_names array
List of tools that require approval.

never object
A list of tools that never require approval.

Hide properties

tool_names array
List of tools that do not require approval.

MCP tool approval setting string
Specify a single approval policy for all tools. One of `always` or `never`. When set to `always`, all tools will require approval. When set to `never`, all tools will not require approval.

server_description string
Optional description of the MCP server, used to provide more context.

Code interpreter object
A tool that runs Python code to help generate a response to a prompt.

Hide properties

container string or object
The code interpreter container. Can be a container ID or an object that specifies uploaded file IDs to make available to your code.

Hide possible types

string
The container ID.

CodeInterpreterContainerAuto object
Configuration for a code interpreter container. Optionally specify the IDs of the files to run the code on.

Hide properties

type string
Always `auto`.

file_ids array
An optional list of uploaded files to make available to your code.

type string

The type of the code interpreter tool. Always `code_interpreter`.

Image generation tool object

A tool that generates images using a model like `gpt-image-1`.

Hide properties

type string

The type of the image generation tool. Always `image_generation`.

background string

Background type for the generated image. One of `transparent`, `opaque`, or `auto`. Default: `auto`.

input_image_mask object

Optional mask for inpainting. Contains `image_url` (string, optional) and `file_id` (string, optional).

Hide properties

file_id string

File ID for the mask image.

image_url string

Base64-encoded mask image.

model string

The image generation model to use. Default: `gpt-image-1`.

moderation string

Moderation level for the generated image. Default: `auto`.

output_compression integer

Compression level for the output image. Default: 100.

output_format string

The output format of the generated image. One of `png`, `webp`, or `jpeg`. Default: `png`.

partial_images integer

Number of partial images to generate in streaming mode, from 0 (default value) to 3.

quality string

The quality of the generated image. One of `low`, `medium`, `high`, or `auto`. Default:

`auto` .

size string
The size of the generated image. One of `1024x1024` , `1024x1536` , `1536x1024` , or `auto` . Default: `auto` .

Local shell tool object
A tool that allows the model to execute shell commands in a local environment.

Hide properties

type string
The type of the local shell tool. Always `local_shell` .

top_p number
An alternative to temperature for nucleus sampling; 1.0 includes all tokens.

error object
An object representing an error response from the Eval API.

Hide properties

code string
The error code.

message string
The error message.

eval_id string
The identifier of the associated evaluation.

id string
Unique identifier for the evaluation run.

metadata map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

model string
The model that is evaluated, if applicable.

name string
The name of the evaluation run.

object string
The type of the object. Always "eval.run".

per_model_usage array
Usage statistics for each model during the evaluation run.

Hide properties

cached_tokens integer
The number of tokens retrieved from cache.

completion_tokens integer
The number of completion tokens generated.

invocation_count integer
The number of invocations.

model_name string
The name of the model.

prompt_tokens integer
The number of prompt tokens used.

total_tokens integer
The total number of tokens used.

per_testing_criteria_results array
Results per testing criteria applied during the evaluation run.

Hide properties

failed integer
Number of tests failed for this criteria.

passed integer
Number of tests passed for this criteria.

testing_criteria string
A description of the testing criteria.

report_url string
The URL to the rendered evaluation run report on the UI dashboard.

result_counts object
Counters summarizing the outcomes of the evaluation run.

^ Hide properties

errored integer
Number of output items that resulted in an error.

failed integer
Number of output items that failed to pass the evaluation.

passed integer
Number of output items that passed the evaluation.

total integer
Total number of executed output items.

status string
The status of the evaluation run.

OBJECT The eval run object

```
{
  "object": "eval.run",
  "id": "evalrun_67e57965b480819094274e3a32235e4c",
  "eval_id": "eval_67e579652b548190aaa83ada4b125f47",
  "report_url": "https://platform.openai.com/evaluations/eval_67e579652b548190aaa83ada4b1
  "status": "queued",
  "model": "gpt-4o-mini",
  "name": "gpt-4o-mini",
  "created_at": 1743092069,
  "result_counts": {
    "total": 0,
    "errored": 0,
    "failed": 0,
    "passed": 0
  },
```

```
  "per_model_usage": null,
  "per_testing_criteria_results": null,
  "data_source": {
    "type": "completions",
    "source": {
      "type": "file_content",
      "content": [
        {
          "item": {
            "input": "Tech Company Launches Advanced Artificial Intelligence Platform",
            "ground_truth": "Technology"
          }
        },
        {
          "item": {
            "input": "Central Bank Increases Interest Rates Amid Inflation Concerns",
            "ground_truth": "Markets"
          }
        },
        {
          "item": {
            "input": "International Summit Addresses Climate Change Strategies",
            "ground_truth": "World"
          }
        },
        {
          "item": {
            "input": "Major Retailer Reports Record-Breaking Holiday Sales",
            "ground_truth": "Business"
          }
        },
        {
          "item": {
            "input": "National Team Qualifies for World Championship Finals",
            "ground_truth": "Sports"
          }
        },
        {
          "item": {
            "input": "Stock Markets Rally After Positive Economic Data Released",
            "ground_truth": "Markets"
          }
        },
        {
          "item": {
            "input": "Global Manufacturer Announces Merger with Competitor",
            "ground_truth": "Business"
```

```
            }
          },
          {
            "item": {
              "input": "Breakthrough in Renewable Energy Technology Unveiled",
              "ground_truth": "Technology"
            }
          },
          {
            "item": {
              "input": "World Leaders Sign Historic Climate Agreement",
              "ground_truth": "World"
            }
          },
          {
            "item": {
              "input": "Professional Athlete Sets New Record in Championship Event",
              "ground_truth": "Sports"
            }
          },
          {
            "item": {
              "input": "Financial Institutions Adapt to New Regulatory Requirements",
              "ground_truth": "Business"
            }
          },
          {
            "item": {
              "input": "Tech Conference Showcases Advances in Artificial Intelligence",
              "ground_truth": "Technology"
            }
          },
          {
            "item": {
              "input": "Global Markets Respond to Oil Price Fluctuations",
              "ground_truth": "Markets"
            }
          },
          {
            "item": {
              "input": "International Cooperation Strengthened Through New Treaty",
              "ground_truth": "World"
            }
          },
          {
            "item": {
              "input": "Sports League Announces Revised Schedule for Upcoming Season",
```

```
112             "ground_truth": "Sports"
113           }
114         }
115       ]
116     },
117     "input_messages": {
118       "type": "template",
119       "template": [
120         {
121           "type": "message",
122           "role": "developer",
123           "content": {
124             "type": "input_text",
125             "text": "Categorize a given news headline into one of the following topics: T
126           }
127         },
128         {
129           "type": "message",
130           "role": "user",
131           "content": {
132             "type": "input_text",
133             "text": "{{item.input}}"
134           }
135         }
136       ]
137     },
138     "model": "gpt-4o-mini",
139     "sampling_params": {
140       "seed": 42,
141       "temperature": 1.0,
142       "top_p": 1.0,
143       "max_completions_tokens": 2048
144     }
145   },
146   "error": null,
  "metadata": {}
}
```

The eval run output item object

A schema representing an evaluation run output item.

created_at integer

Unix timestamp (in seconds) when the evaluation run was created.

datasource_item object
Details of the input data source item.

datasource_item_id integer
The identifier for the data source item.

eval_id string
The identifier of the evaluation group.

id string
Unique identifier for the evaluation run output item.

object string
The type of the object. Always "eval.run.output_item".

results array
A list of results from the evaluation run.

Hide properties

run_id string
The identifier of the evaluation run associated with this output item.

sample object
A sample containing the input and output of the evaluation run.

Hide properties

error object
An object representing an error response from the Eval API.

Hide properties

code string
The error code.

message string
The error message.

finish_reason string
The reason why the sample generation was finished.

input array
An array of input messages.

Hide properties

content string
The content of the message.

role string
The role of the message sender (e.g., system, user, developer).

max_completion_tokens integer
The maximum number of tokens allowed for completion.

model string
The model used for generating the sample.

output array
An array of output messages.

Hide properties

content string
The content of the message.

role string
The role of the message (e.g. "system", "assistant", "user").

seed integer
The seed used for generating the sample.

temperature number
The sampling temperature used.

top_p number
The top_p value used for sampling.

usage object
Token usage details for the sample.

Hide properties

cached_tokens integer
The number of tokens retrieved from cache.

completion_tokens integer
The number of completion tokens generated.

prompt_tokens integer
The number of prompt tokens used.

total_tokens integer
The total number of tokens used.

status string
The status of the evaluation run.

OBJECT The eval run output item object

```
{
  "object": "eval.run.output_item",
  "id": "outputitem_67abd55eb6548190bb580745d5644a33",
  "run_id": "evalrun_67abd54d60ec8190832b46859da808f7",
  "eval_id": "eval_67abd54d9b0081909a86353f6fb9317a",
  "created_at": 1739314509,
  "status": "pass",
  "datasource_item_id": 137,
  "datasource_item": {
      "teacher": "To grade essays, I only check for style, content, and grammar.",
      "student": "I am a student who is trying to write the best essay."
  },
  "results": [
    {
      "name": "String Check Grader",
      "type": "string-check-grader",
      "score": 1.0,
      "passed": true,
    }
  ],
  "sample": {
    "input": [
      {
        "role": "system",
        "content": "You are an evaluator bot..."
      },
      {
        "role": "user",
        "content": "You are assessing..."

```

```
      }
    ],
    "output": [
      {
        "role": "assistant",
        "content": "The rubric is not clear nor concise."
      }
    ],
    "finish_reason": "stop",
    "model": "gpt-4o-2024-08-06",
    "usage": {
      "total_tokens": 521,
      "completion_tokens": 2,
      "prompt_tokens": 519,
      "cached_tokens": 0
    },
    "error": null,
    "temperature": 1.0,
    "max_completion_tokens": 2048,
    "top_p": 1.0,
    "seed": 42
  }
}
```

Fine-tuning

Manage fine-tuning jobs to tailor a model to your specific training data. Related guide: Fine-tune models

Create fine-tuning job

```
POST https://api.openai.com/v1/fine_tuning/jobs
```

Creates a fine-tuning job which begins the process of creating a new model from a given dataset.
Response includes details of the enqueued job including job status and the name of the fine-tuned models once complete.
Learn more about fine-tuning

Request body

`model` string Required
The name of the model to fine-tune. You can select one of the supported models.

`training_file` string Required
The ID of an uploaded file that contains training data.

See upload file for how to upload a file.

Your dataset must be formatted as a JSONL file. Additionally, you must upload your file with the purpose `fine-tune`.

The contents of the file should differ depending on if the model uses the chat, completions format, or if the fine-tuning method uses the preference format.

See the fine-tuning guide for more details.

`hyperparameters` Deprecated object Optional
The hyperparameters used for the fine-tuning job. This value is now deprecated in favor of `method`, and should be passed in under the `method` parameter.

Hide properties

`batch_size` "auto" or integer Optional Defaults to auto
Number of examples in each batch. A larger batch size means that model parameters are updated less frequently, but with lower variance.

`learning_rate_multiplier` "auto" or number Optional Defaults to auto
Scaling factor for the learning rate. A smaller learning rate may be useful to avoid overfitting.

`n_epochs` "auto" or integer Optional Defaults to auto
The number of epochs to train the model for. An epoch refers to one full cycle through the training dataset.

`integrations` array or null Optional
A list of integrations to enable for your fine-tuning job.

Hide properties

`type` "wandb" Required
The type of integration to enable. Currently, only "wandb" (Weights and Biases) is supported.

`wandb` object Required
The settings for your integration with Weights and Biases. This payload specifies the project that metrics will be sent to. Optionally, you can set an explicit display name for your run, add tags to your run, and set

a default entity (team, username, etc) to be associated with your run.

Hide properties

project string Required
The name of the project that the new run will be created under.

entity string or null Optional
The entity to use for the run. This allows you to set the team or username of the WandB user that you would like associated with the run. If not set, the default entity for the registered WandB API key is used.

name string or null Optional
A display name to set for the run. If not set, we will use the Job ID as the name.

tags array Optional
A list of tags to be attached to the newly created run. These tags are passed through directly to WandB. Some default tags are generated by OpenAI: "openai/finetune", "openai/{base-model}", "openai/{ftjob-abcdef}".

metadata map Optional
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

method object Optional
The method used for fine-tuning.

Hide properties

type string Required
The type of method. Is either `supervised`, `dpo`, or `reinforcement`.

dpo object Optional
Configuration for the DPO fine-tuning method.

Hide properties

hyperparameters object Optional
The hyperparameters used for the DPO fine-tuning job.

Hide properties

batch_size "auto" or integer Optional Defaults to auto

Number of examples in each batch. A larger batch size means that model parameters are updated less frequently, but with lower variance.

beta "auto" or number Optional Defaults to auto
The beta value for the DPO method. A higher beta value will increase the weight of the penalty between the policy and reference model.

learning_rate_multiplier "auto" or number Optional Defaults to auto
Scaling factor for the learning rate. A smaller learning rate may be useful to avoid overfitting.

n_epochs "auto" or integer Optional Defaults to auto
The number of epochs to train the model for. An epoch refers to one full cycle through the training dataset.

reinforcement object Optional
Configuration for the reinforcement fine-tuning method.

Hide properties

grader object Required
The grader used for the fine-tuning job.

Hide possible types

StringCheckGrader object

A StringCheckGrader object that performs a string comparison between input and reference using a specified operation.

Hide properties

input string Required

The input text. This may include template strings.

name string Required

The name of the grader.

operation string Required

The string check operation to perform. One of `eq`, `ne`, `like`, or `ilike`.

reference string Required

The reference text. This may include template strings.

type string Required

The object type, which is always `string_check`.

TextSimilarityGrader object

A TextSimilarityGrader object which grades text based on similarity metrics.

Hide properties

evaluation_metric string Required

The evaluation metric to use. One of `fuzzy_match`, `bleu`, `gleu`, `meteor`, `rouge_1`, `rouge_2`, `rouge_3`, `rouge_4`, `rouge_5`, or `rouge_l`.

input string Required

The text being graded.

name string Required

The name of the grader.

reference string Required

The text being graded against.

type string Required

The type of grader.

PythonGrader object

A PythonGrader object that runs a python script on the input.

Hide properties

name string Required
The name of the grader.

source string Required
The source code of the python script.

type string Required
The object type, which is always `python`.

image_tag string Optional
The image tag to use for the python script.

ScoreModelGrader object
A ScoreModelGrader object that uses a model to assign a score to the input.

Hide properties

input array Required
The input text. This may include template strings.

Hide properties

content string / object / array Required
Inputs to the model - can contain template strings.

Hide possible types

Text input string
A text input to the model.

Input text object
A text input to the model.

Hide properties

text string Required
The text input to the model.

type string Required
The type of the input item. Always `input_text`.

Output text object
A text output from the model.

Hide properties

text string Required
The text output from the model.

type string Required
The type of the output text. Always `output_text`.

Input image object
An image input to the model.

Hide properties

image_url string Required
The URL of the image input.

type string Required
The type of the image input. Always `input_image`.

detail string Optional
The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

An array of Input text and Input image array
A list of inputs, each of which may be either an input text or input image object.

role string Required
The role of the message input. One of `user`, `assistant`, `system`, or `developer`.

type string Optional
The type of the message input. Always `message`.

model string Required
The model to use for the evaluation.

name string Required
The name of the grader.

type string Required
The object type, which is always `score_model`.

range array Optional
The range of the score. Defaults to `[0, 1]`.

sampling_params object Optional
The sampling parameters for the model.

MultiGrader object
A MultiGrader object combines the output of multiple graders to produce a single score.

Hide properties

calculate_output string Required
A formula to calculate the output based on grader results.

graders object Required

Hide possible types

StringCheckGrader object
A StringCheckGrader object that performs a string comparison between input and reference using a specified operation.

Hide properties

input string Required
The input text. This may include template strings.

name string Required
The name of the grader.

operation string Required
The string check operation to perform. One of `eq`, `ne`, `like`, or `ilike`.

reference string Required
The reference text. This may include template strings.

type string Required
The object type, which is always `string_check`.

TextSimilarityGrader object
A TextSimilarityGrader object which grades text based on similarity metrics.

Hide properties

evaluation_metric string Required

The evaluation metric to use. One of `fuzzy_match`, `bleu`, `gleu`, `meteor`, `rouge_1`, `rouge_2`, `rouge_3`, `rouge_4`, `rouge_5`, or `rouge_l`.

`input` string Required
The text being graded.

`name` string Required
The name of the grader.

`reference` string Required
The text being graded against.

`type` string Required
The type of grader.

PythonGrader object
A PythonGrader object that runs a python script on the input.

Hide properties

`name` string Required
The name of the grader.

`source` string Required
The source code of the python script.

`type` string Required
The object type, which is always `python`.

`image_tag` string Optional
The image tag to use for the python script.

ScoreModelGrader object
A ScoreModelGrader object that uses a model to assign a score to the input.

Hide properties

`input` array Required
The input text. This may include template strings.

Hide properties

`content` string / object / array Required
Inputs to the model - can contain template strings.

Hide possible types

Text input string
A text input to the model.

Input text object
A text input to the model.

Hide properties

text string Required
The text input to the model.

type string Required
The type of the input item. Always `input_text`.

Output text object
A text output from the model.

Hide properties

text string Required
The text output from the model.

type string Required
The type of the output text. Always `output_text`.

Input image object
An image input to the model.

Hide properties

image_url string Required
The URL of the image input.

type string Required
The type of the image input. Always `input_image`.

detail string Optional
The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

An array of Input text and Input image array
A list of inputs, each of which may be either an input text or input image

object.

role string Required
The role of the message input. One of `user`, `assistant`, `system`, or `developer`.

type string Optional
The type of the message input. Always `message`.

model string Required
The model to use for the evaluation.

name string Required
The name of the grader.

type string Required
The object type, which is always `score_model`.

range array Optional
The range of the score. Defaults to `[0, 1]`.

sampling_params object Optional
The sampling parameters for the model.

LabelModelGrader object
A LabelModelGrader object which uses a model to assign labels to each item in the evaluation.

Hide properties

input array Required

Hide properties

content string / object / array Required
Inputs to the model - can contain template strings.

Hide possible types

Text input string
A text input to the model.

Input text object
A text input to the model.

Hide properties

text string Required
The text input to the model.

type string Required
The type of the input item. Always `input_text`.

Output text object
A text output from the model.

Hide properties

text string Required
The text output from the model.

type string Required
The type of the output text. Always `output_text`.

Input image object
An image input to the model.

Hide properties

image_url string Required
The URL of the image input.

type string Required
The type of the image input. Always `input_image`.

detail string Optional
The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

An array of Input text and Input image array
A list of inputs, each of which may be either an input text or input image object.

role string Required
The role of the message input. One of `user`, `assistant`, `system`, or `developer`.

type string Optional

The type of the message input. Always `message`.

labels array Required
The labels to assign to each item in the evaluation.

model string Required
The model to use for the evaluation. Must support structured outputs.

name string Required
The name of the grader.

passing_labels array Required
The labels that indicate a passing result. Must be a subset of labels.

type string Required
The object type, which is always `label_model`.

name string Required
The name of the grader.

type string Required
The object type, which is always `multi`.

hyperparameters object Optional
The hyperparameters used for the reinforcement fine-tuning job.

Hide properties

batch_size "auto" or integer Optional Defaults to auto
Number of examples in each batch. A larger batch size means that model parameters are updated less frequently, but with lower variance.

compute_multiplier "auto" or number Optional Defaults to auto
Multiplier on amount of compute used for exploring search space during training.

eval_interval "auto" or integer Optional Defaults to auto
The number of training steps between evaluation runs.

eval_samples "auto" or integer Optional Defaults to auto
Number of evaluation samples to generate per training step.

learning_rate_multiplier "auto" or number Optional Defaults to auto
Scaling factor for the learning rate. A smaller learning rate may be useful to avoid overfitting.

n_epochs "auto" or integer Optional Defaults to auto
The number of epochs to train the model for. An epoch refers to one full cycle through the training dataset.

reasoning_effort string Optional Defaults to default
Level of reasoning effort.

supervised object Optional
Configuration for the supervised fine-tuning method.

Hide properties

hyperparameters object Optional
The hyperparameters used for the fine-tuning job.

Hide properties

batch_size "auto" or integer Optional Defaults to auto
Number of examples in each batch. A larger batch size means that model parameters are updated less frequently, but with lower variance.

learning_rate_multiplier "auto" or number Optional Defaults to auto
Scaling factor for the learning rate. A smaller learning rate may be useful to avoid overfitting.

n_epochs "auto" or integer Optional Defaults to auto
The number of epochs to train the model for. An epoch refers to one full cycle through the training dataset.

seed integer or null Optional
The seed controls the reproducibility of the job. Passing in the same seed and job parameters should produce the same results, but may differ in rare cases. If a seed is not specified, one will be generated for you.

suffix string or null Optional Defaults to null
A string of up to 64 characters that will be added to your fine-tuned model name.

For example, a `suffix` of "custom-model-name" would produce a model name like `ft:gpt-4o-mini:openai:custom-model-name:7p4lURel`.

validation_file string or null Optional

The ID of an uploaded file that contains validation data.

If you provide this file, the data is used to generate validation metrics periodically during fine-tuning. These metrics can be viewed in the fine-tuning results file. The same data should not be present in both train and validation files.

Your dataset must be formatted as a JSONL file. You must upload your file with the purpose `fine-tune`.

See the fine-tuning guide for more details.

Returns

A fine-tuning.job object.

Default | Epochs | DPO | Reinforcement | Validation file | W&B Integration

Example request curl

```
curl https://api.openai.com/v1/fine_tuning/jobs \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -d '{
    "training_file": "file-BK7bzQj3FfZFXr7DbL6xJwfo",
    "model": "gpt-4o-mini"
  }'
```

Response

```
{
  "object": "fine_tuning.job",
  "id": "ftjob-abc123",
  "model": "gpt-4o-mini-2024-07-18",
  "created_at": 1721764800,
  "fine_tuned_model": null,
  "organization_id": "org-123",
  "result_files": [],
  "status": "queued",
  "validation_file": null,
  "training_file": "file-abc123",
  "method": {
    "type": "supervised",
    "supervised": {

```

```
      "hyperparameters": {
        "batch_size": "auto",
        "learning_rate_multiplier": "auto",
        "n_epochs": "auto",
      }
    }
  },
  "metadata": null
}
```

List fine-tuning jobs

`GET https://api.openai.com/v1/fine_tuning/jobs`

List your organization's fine-tuning jobs

Query parameters

after string Optional
Identifier for the last job from the previous pagination request.

limit integer Optional Defaults to 20
Number of fine-tuning jobs to retrieve.

metadata object or null Optional
Optional metadata filter. To filter, use the syntax `metadata[k]=v`. Alternatively, set `metadata=null` to indicate no metadata.

Returns

A list of paginated fine-tuning job objects.

Example request curl

```
curl https://api.openai.com/v1/fine_tuning/jobs?limit=2&metadata[key]=value \
  -H "Authorization: Bearer $OPENAI_API_KEY"
```

Response

```
{
  "object": "list",
  "data": [
    {
      "object": "fine_tuning.job",
      "id": "ftjob-abc123",
      "model": "gpt-4o-mini-2024-07-18",
      "created_at": 1721764800,
      "fine_tuned_model": null,
      "organization_id": "org-123",
      "result_files": [],
      "status": "queued",
      "validation_file": null,
      "training_file": "file-abc123",
      "metadata": {
        "key": "value"
      }
    },
    { ... },
    { ... }
  ], "has_more": true
}
```

List fine-tuning events

```
GET https://api.openai.com/v1/fine_tuning/jobs/{fine_tuning_job_id}/events
```

Get status updates for a fine-tuning job.

Path parameters

fine_tuning_job_id string Required
The ID of the fine-tuning job to get events for.

Query parameters

after string Optional

Identifier for the last event from the previous pagination request.

limit integer Optional Defaults to 20

Number of events to retrieve.

Returns

A list of fine-tuning event objects.

Example request curl

```curl
curl https://api.openai.com/v1/fine_tuning/jobs/ftjob-abc123/events \
  -H "Authorization: Bearer $OPENAI_API_KEY"
```

Response

```response
{
  "object": "list",
  "data": [
    {
      "object": "fine_tuning.job.event",
      "id": "ft-event-ddTJfwuMVpfLXseO0Am0Gqjm",
      "created_at": 1721764800,
      "level": "info",
      "message": "Fine tuning job successfully completed",
      "data": null,
      "type": "message"
    },
    {
      "object": "fine_tuning.job.event",
      "id": "ft-event-tyiGuB72evQncpH87xe505Sv",
      "created_at": 1721764800,
      "level": "info",
      "message": "New fine-tuned model created: ft:gpt-4o-mini:openai::7p4lURel",
      "data": null,
      "type": "message"
    }
  ],
  "has_more": true
}
```

List fine-tuning checkpoints

```
GET https://api.openai.com/v1/fine_tuning/jobs/{fine_tuning_job_id}/checkpoints
```

List checkpoints for a fine-tuning job.

Path parameters

fine_tuning_job_id string Required
The ID of the fine-tuning job to get checkpoints for.

Query parameters

after string Optional
Identifier for the last checkpoint ID from the previous pagination request.

limit integer Optional Defaults to 10
Number of checkpoints to retrieve.

Returns

A list of fine-tuning checkpoint objects for a fine-tuning job.

Example request curl

```curl
curl https://api.openai.com/v1/fine_tuning/jobs/ftjob-abc123/checkpoints \
  -H "Authorization: Bearer $OPENAI_API_KEY"
```

Response

```
{
  "object": "list",
  "data": [
    {
      "object": "fine_tuning.job.checkpoint",
      "id": "ftckpt_zc4Q7MP6XxulcVzj4MZdwsAB",
      "created_at": 1721764867,
      "fine_tuned_model_checkpoint": "ft:gpt-4o-mini-2024-07-18:my-org:custom-suffix:96oll
```

```
      "metrics": {
        "full_valid_loss": 0.134,
        "full_valid_mean_token_accuracy": 0.874
      },
      "fine_tuning_job_id": "ftjob-abc123",
      "step_number": 2000
    },
    {
      "object": "fine_tuning.job.checkpoint",
      "id": "ftckpt_enQCFmOTGj3syEpYVhBRLTSy",
      "created_at": 1721764800,
      "fine_tuned_model_checkpoint": "ft:gpt-4o-mini-2024-07-18:my-org:custom-suffix:7q8mp
      "metrics": {
        "full_valid_loss": 0.167,
        "full_valid_mean_token_accuracy": 0.781
      },
      "fine_tuning_job_id": "ftjob-abc123",
      "step_number": 1000
    }
  ],
  "first_id": "ftckpt_zc4Q7MP6XxulcVzj4MZdwsAB",
  "last_id": "ftckpt_enQCFmOTGj3syEpYVhBRLTSy",
  "has_more": true
}
```

List checkpoint permissions

```
GET https://api.openai.com/v1/fine_tuning/checkpoints/{fine_tuned_model_checkpoint}/permissio
    ns
```

NOTE: This endpoint requires an admin API key.

Organization owners can use this endpoint to view all permissions for a fine-tuned model checkpoint.

Path parameters

`fine_tuned_model_checkpoint` string Required

The ID of the fine-tuned model checkpoint to get permissions for.

Query parameters

after string Optional
Identifier for the last permission ID from the previous pagination request.

limit integer Optional Defaults to 10
Number of permissions to retrieve.

order string Optional Defaults to descending
The order in which to retrieve permissions.

project_id string Optional
The ID of the project to get permissions for.

Returns

A list of fine-tuned model checkpoint permission objects for a fine-tuned model checkpoint.

Example request curl

```curl
curl https://api.openai.com/v1/fine_tuning/checkpoints/ft:gpt-4o-mini-2024-07-18:org:weathe
  -H "Authorization: Bearer $OPENAI_API_KEY"
```

Response

```json
{
  "object": "list",
  "data": [
    {
      "object": "checkpoint.permission",
      "id": "cp_zc4Q7MP6XxulcVzj4MZdwsAB",
      "created_at": 1721764867,
      "project_id": "proj_abGMw1llN8IrBb6SvvY5A1iH"
    },
    {
      "object": "checkpoint.permission",
      "id": "cp_enQCFmOTGj3syEpYVhBRLTSy",
      "created_at": 1721764800,
      "project_id": "proj_iqGMw1llN8IrBb6SvvY5A1oF"
    },
  ],
  "first_id": "cp_zc4Q7MP6XxulcVzj4MZdwsAB",
  "last_id": "cp_enQCFmOTGj3syEpYVhBRLTSy",
```

```
  "has_more": false
}
```

Create checkpoint permissions

```
POST https://api.openai.com/v1/fine_tuning/checkpoints/{fine_tuned_model_checkpoint}/permissi
     ons
```

NOTE: Calling this endpoint requires an admin API key.
This enables organization owners to share fine-tuned models with other projects in their organization.

Path parameters

`fine_tuned_model_checkpoint` string Required
The ID of the fine-tuned model checkpoint to create a permission for.

Request body

`project_ids` array Required
The project identifiers to grant access to.

Returns

A list of fine-tuned model checkpoint permission objects for a fine-tuned model checkpoint.

Example request curl

```
curl https://api.openai.com/v1/fine_tuning/checkpoints/ft:gpt-4o-mini-2024-07-18:org:weathe
  -H "Authorization: Bearer $OPENAI_API_KEY"
  -d '{"project_ids": ["proj_abGMw1llN8IrBb6SvvY5A1iH"]}'
```

Response

```
{
  "object": "list",
```

```
  "data": [
    {
      "object": "checkpoint.permission",
      "id": "cp_zc4Q7MP6XxulcVzj4MZdwsAB",
      "created_at": 1721764867,
      "project_id": "proj_abGMw1llN8IrBb6SvvY5A1iH"
    }
  ],
  "first_id": "cp_zc4Q7MP6XxulcVzj4MZdwsAB",
  "last_id": "cp_zc4Q7MP6XxulcVzj4MZdwsAB",
  "has_more": false
}
```

Delete checkpoint permission

```
DELETE https://api.openai.com/v1/fine_tuning/checkpoints/{fine_tuned_model_checkpoint}/permis
       sions/{permission_id}
```

NOTE: This endpoint requires an admin API key.
Organization owners can use this endpoint to delete a permission for a fine-tuned model checkpoint.

Path parameters

`fine_tuned_model_checkpoint` string Required
The ID of the fine-tuned model checkpoint to delete a permission for.

`permission_id` string Required
The ID of the fine-tuned model checkpoint permission to delete.

Returns

The deletion status of the fine-tuned model checkpoint permission object.

Example request curl

```
curl https://api.openai.com/v1/fine_tuning/checkpoints/ft:gpt-4o-mini-2024-07-18:org:weathe
  -H "Authorization: Bearer $OPENAI_API_KEY"
```

Response

```
{
  "object": "checkpoint.permission",
  "id": "cp_zc4Q7MP6XxulcVzj4MZdwsAB",
  "deleted": true
}
```

Retrieve fine-tuning job

```
GET https://api.openai.com/v1/fine_tuning/jobs/{fine_tuning_job_id}
```

Get info about a fine-tuning job.
Learn more about fine-tuning

Path parameters

fine_tuning_job_id string Required
The ID of the fine-tuning job.

Returns

The fine-tuning object with the given ID.

Example request curl

```
curl https://api.openai.com/v1/fine_tuning/jobs/ft-AF1WoRqd3aJAHsqc9NY7iL8F \
  -H "Authorization: Bearer $OPENAI_API_KEY"
```

Response

```
{
  "object": "fine_tuning.job",
  "id": "ftjob-abc123",
  "model": "davinci-002",
  "created_at": 1692661014,
  "finished_at": 1692661190,
  "fine_tuned_model": "ft:davinci-002:my-org:custom_suffix:7q8mpxmy",
```

```
  "organization_id": "org-123",
  "result_files": [
      "file-abc123"
  ],
  "status": "succeeded",
  "validation_file": null,
  "training_file": "file-abc123",
  "hyperparameters": {
      "n_epochs": 4,
      "batch_size": 1,
      "learning_rate_multiplier": 1.0
  },
  "trained_tokens": 5768,
  "integrations": [],
  "seed": 0,
  "estimated_finish": 0,
  "method": {
    "type": "supervised",
    "supervised": {
      "hyperparameters": {
        "n_epochs": 4,
        "batch_size": 1,
        "learning_rate_multiplier": 1.0
      }
    }
  }
}
```

Cancel fine-tuning

```
POST https://api.openai.com/v1/fine_tuning/jobs/{fine_tuning_job_id}/cancel
```

Immediately cancel a fine-tune job.

Path parameters

`fine_tuning_job_id` string Required

The ID of the fine-tuning job to cancel.

Returns

The cancelled fine-tuning object.

Example request curl

```curl
curl -X POST https://api.openai.com/v1/fine_tuning/jobs/ftjob-abc123/cancel \
  -H "Authorization: Bearer $OPENAI_API_KEY"
```

Response

```response
{
  "object": "fine_tuning.job",
  "id": "ftjob-abc123",
  "model": "gpt-4o-mini-2024-07-18",
  "created_at": 1721764800,
  "fine_tuned_model": null,
  "organization_id": "org-123",
  "result_files": [],
  "status": "cancelled",
  "validation_file": "file-abc123",
  "training_file": "file-abc123"
}
```

Resume fine-tuning

`POST https://api.openai.com/v1/fine_tuning/jobs/{fine_tuning_job_id}/resume`

Resume a fine-tune job.

Path parameters

fine_tuning_job_id string Required
The ID of the fine-tuning job to resume.

Returns

The resumed fine-tuning object.

Example request curl

```
curl -X POST https://api.openai.com/v1/fine_tuning/jobs/ftjob-abc123/resume \
  -H "Authorization: Bearer $OPENAI_API_KEY"
```

Response

```
{
  "object": "fine_tuning.job",
  "id": "ftjob-abc123",
  "model": "gpt-4o-mini-2024-07-18",
  "created_at": 1721764800,
  "fine_tuned_model": null,
  "organization_id": "org-123",
  "result_files": [],
  "status": "queued",
  "validation_file": "file-abc123",
  "training_file": "file-abc123"
}
```

Pause fine-tuning

`POST https://api.openai.com/v1/fine_tuning/jobs/{fine_tuning_job_id}/pause`

Pause a fine-tune job.

Path parameters

fine_tuning_job_id string Required
The ID of the fine-tuning job to pause.

Returns

The paused fine-tuning object.

Example request curl

```
curl -X POST https://api.openai.com/v1/fine_tuning/jobs/ftjob-abc123/pause \
```

```
  -H "Authorization: Bearer $OPENAI_API_KEY"
```

Response

```
{
  "object": "fine_tuning.job",
  "id": "ftjob-abc123",
  "model": "gpt-4o-mini-2024-07-18",
  "created_at": 1721764800,
  "fine_tuned_model": null,
  "organization_id": "org-123",
  "result_files": [],
  "status": "paused",
  "validation_file": "file-abc123",
  "training_file": "file-abc123"
}
```

Training format for chat models using the supervised method

The per-line training example of a fine-tuning input file for chat models using the supervised method. Input messages may contain text or image content only. Audio and file input messages are not currently supported for fine-tuning.

functions Deprecated array

A list of functions the model may generate JSON inputs for.

Hide properties

name string

The name of the function to be called. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

description string

A description of what the function does, used by the model to choose when and how to call the function.

parameters object

The parameters the functions accepts, described as a JSON Schema object. See the guide for examples, and the JSON Schema reference for documentation about the format.

Omitting `parameters` defines a function with an empty parameter list.

messages array

Hide possible types

System message object

Developer-provided instructions that the model should follow, regardless of messages sent by the user. With o1 models and newer, use `developer` messages for this purpose instead.

Hide properties

content string or array

The contents of the system message.

Hide possible types

Text content string

The contents of the system message.

Array of content parts array

An array of content parts with a defined type. For system messages, only type `text` is supported.

Hide possible types

Text content part object

Learn about text inputs.

Hide properties

text string

The text content.

type string
The type of the content part.

role string
The role of the messages author, in this case `system`.

name string
An optional name for the participant. Provides the model information to differentiate between participants of the same role.

User message object
Messages sent by an end user, containing prompts or additional context information.

Hide properties

content string or array
The contents of the user message.

Hide possible types

Text content string
The text contents of the message.

Array of content parts array
An array of content parts with a defined type. Supported options differ based on the model being used to generate the response. Can contain text, image, or audio inputs.

Hide possible types

Text content part object
Learn about text inputs.

Hide properties

text string
The text content.

type string
The type of the content part.

Image content part object
Learn about image inputs.

Hide properties

image_url object

Hide properties

url string

Either a URL of the image or the base64 encoded image data.

detail string

Specifies the detail level of the image. Learn more in the Vision guide.

type string

The type of the content part.

Audio content part object

Learn about audio inputs.

Hide properties

input_audio object

Hide properties

data string

Base64 encoded audio data.

format string

The format of the encoded audio data. Currently supports "wav" and "mp3".

type string

The type of the content part. Always `input_audio` .

File content part object

Learn about file inputs for text generation.

Hide properties

file object

Hide properties

file_data string

The base64 encoded file data, used when passing the file to the model as a string.

file_id string

The ID of an uploaded file to use as input.

filename string
The name of the file, used when passing the file to the model as a string.

type string
The type of the content part. Always `file`.

role string
The role of the messages author, in this case `user`.

name string
An optional name for the participant. Provides the model information to differentiate between participants of the same role.

Assistant message object
Messages sent by the model in response to user messages.

Hide properties

role string
The role of the messages author, in this case `assistant`.

audio object or null
Data about a previous audio response from the model. Learn more.

Hide properties

id string
Unique identifier for a previous audio response from the model.

content string or array
The contents of the assistant message. Required unless `tool_calls` or `function_call` is specified.

Hide possible types

Text content string
The contents of the assistant message.

Array of content parts array
An array of content parts with a defined type. Can be one or more of type `text`, or exactly one

of type `refusal`.

Hide possible types

Text content part object

Learn about text inputs.

Hide properties

text string

The text content.

type string

The type of the content part.

Refusal content part object

Hide properties

refusal string

The refusal message generated by the model.

type string

The type of the content part.

function_call Deprecated object or null

Deprecated and replaced by `tool_calls`. The name and arguments of a function that should be called, as generated by the model.

Hide properties

arguments string

The arguments to call the function with, as generated by the model in JSON format. Note that the model does not always generate valid JSON, and may hallucinate parameters not defined by your function schema. Validate the arguments in your code before calling your function.

name string

The name of the function to call.

name string

An optional name for the participant. Provides the model information to differentiate between participants of the same role.

refusal string or null

The refusal message by the assistant.

tool_calls array

The tool calls generated by the model, such as function calls.

Hide properties

function object

The function that the model called.

Hide properties

arguments string

The arguments to call the function with, as generated by the model in JSON format. Note that the model does not always generate valid JSON, and may hallucinate parameters not defined by your function schema. Validate the arguments in your code before calling your function.

name string

The name of the function to call.

id string

The ID of the tool call.

type string

The type of the tool. Currently, only `function` is supported.

weight integer

Controls whether the assistant message is trained against (0 or 1)

Tool message object

Hide properties

content string or array

The contents of the tool message.

Hide possible types

Text content string

The contents of the tool message.

Array of content parts array

An array of content parts with a defined type. For tool messages, only type `text` is supported.

Hide possible types

Text content part object

Learn about text inputs.

Hide properties

text string

The text content.

type string

The type of the content part.

role string

The role of the messages author, in this case `tool`.

tool_call_id string

Tool call that this message is responding to.

Function message object Deprecated

Hide properties

content string or null

The contents of the function message.

name string

The name of the function to call.

role string

The role of the messages author, in this case `function`.

parallel_tool_calls boolean

Whether to enable parallel function calling during tool use.

tools array

A list of tools the model may generate JSON inputs for.

Hide properties

function object

Hide properties

name string

The name of the function to be called. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

description string
A description of what the function does, used by the model to choose when and how to call the function.

parameters object
The parameters the functions accepts, described as a JSON Schema object. See the guide for examples, and the JSON Schema reference for documentation about the format.

Omitting `parameters` defines a function with an empty parameter list.

strict boolean or null
Whether to enable strict schema adherence when generating the function call. If set to true, the model will follow the exact schema defined in the `parameters` field. Only a subset of JSON Schema is supported when `strict` is `true`. Learn more about Structured Outputs in the function calling guide.

type string
The type of the tool. Currently, only `function` is supported.

OBJECT Training format for chat models using the supervised method

```
{
  "messages": [
    { "role": "user", "content": "What is the weather in San Francisco?" },
    {
      "role": "assistant",
      "tool_calls": [
        {
          "id": "call_id",
          "type": "function",
          "function": {
            "name": "get_current_weather",
            "arguments": "{\"location\": \"San Francisco, USA\", \"format\": \"celsius\"}"
          }
        }
      ]
    }
  ],
  "parallel_tool_calls": false,
  "tools": [
    {
```

```
      "type": "function",
      "function": {
        "name": "get_current_weather",
        "description": "Get the current weather",
        "parameters": {
          "type": "object",
          "properties": {
            "location": {
                "type": "string",
                "description": "The city and country, eg. San Francisco, USA"
            },
            "format": { "type": "string", "enum": ["celsius", "fahrenheit"] }
          },
          "required": ["location", "format"]
        }
      }
    }
  ]
}
```

Training format for chat models using the preference method

The per-line training example of a fine-tuning input file for chat models using the dpo method. Input messages may contain text or image content only. Audio and file input messages are not currently supported for fine-tuning.

input object

Hide properties

messages array

Hide possible types

System message object

Developer-provided instructions that the model should follow, regardless of messages sent by the user. With o1 models and newer, use `developer` messages for this purpose instead.

Hide properties

content string or array

The contents of the system message.

Hide possible types

Text content string
The contents of the system message.

Array of content parts array
An array of content parts with a defined type. For system messages, only type `text` is supported.

Hide possible types

Text content part object
Learn about text inputs.

Hide properties

text string
The text content.

type string
The type of the content part.

role string
The role of the messages author, in this case `system`.

name string
An optional name for the participant. Provides the model information to differentiate between participants of the same role.

User message object
Messages sent by an end user, containing prompts or additional context information.

Hide properties

content string or array
The contents of the user message.

Hide possible types

Text content string
The text contents of the message.

Array of content parts array
An array of content parts with a defined type. Supported options differ based on the model being used to generate the response. Can contain text, image, or audio inputs.

Hide possible types

Text content part object

Learn about text inputs.

Hide properties

text string

The text content.

type string

The type of the content part.

Image content part object

Learn about image inputs.

Hide properties

image_url object

Hide properties

url string

Either a URL of the image or the base64 encoded image data.

detail string

Specifies the detail level of the image. Learn more in the Vision guide.

type string

The type of the content part.

Audio content part object

Learn about audio inputs.

Hide properties

input_audio object

Hide properties

data string

Base64 encoded audio data.

format string

The format of the encoded audio data. Currently supports "wav" and "mp3".

type string

The type of the content part. Always `input_audio`.

File content part object

Learn about file inputs for text generation.

Hide properties

file object

Hide properties

file_data string

The base64 encoded file data, used when passing the file to the model as a string.

file_id string

The ID of an uploaded file to use as input.

filename string

The name of the file, used when passing the file to the model as a string.

type string

The type of the content part. Always `file`.

role string

The role of the messages author, in this case `user`.

name string

An optional name for the participant. Provides the model information to differentiate between participants of the same role.

Assistant message object

Messages sent by the model in response to user messages.

Hide properties

role string

The role of the messages author, in this case `assistant`.

audio object or null

Data about a previous audio response from the model. Learn more.

Hide properties

id string
Unique identifier for a previous audio response from the model.

content string or array
The contents of the assistant message. Required unless `tool_calls` or `function_call` is specified.

Hide possible types

Text content string
The contents of the assistant message.

Array of content parts array
An array of content parts with a defined type. Can be one or more of type `text`, or exactly one of type `refusal`.

Hide possible types

Text content part object
Learn about text inputs.

Hide properties

text string
The text content.

type string
The type of the content part.

Refusal content part object

Hide properties

refusal string
The refusal message generated by the model.

type string
The type of the content part.

function_call Deprecated object or null
Deprecated and replaced by `tool_calls`. The name and arguments of a function that should be called, as generated by the model.

Hide properties

arguments string
The arguments to call the function with, as generated by the model in JSON format. Note that the model does not always generate valid JSON, and may hallucinate parameters not defined by your function schema. Validate the arguments in your code before calling your function.

name string
The name of the function to call.

name string
An optional name for the participant. Provides the model information to differentiate between participants of the same role.

refusal string or null
The refusal message by the assistant.

tool_calls array
The tool calls generated by the model, such as function calls.

Hide properties

function object
The function that the model called.

Hide properties

arguments string
The arguments to call the function with, as generated by the model in JSON format. Note that the model does not always generate valid JSON, and may hallucinate parameters not defined by your function schema. Validate the arguments in your code before calling your function.

name string
The name of the function to call.

id string
The ID of the tool call.

type string
The type of the tool. Currently, only `function` is supported.

weight integer
Controls whether the assistant message is trained against (0 or 1)

Tool message object

Hide properties

content string or array
The contents of the tool message.

Hide possible types

Text content string
The contents of the tool message.

Array of content parts array
An array of content parts with a defined type. For tool messages, only type `text` is supported.

Hide possible types

Text content part object
Learn about text inputs.

Hide properties

text string
The text content.

type string
The type of the content part.

role string
The role of the messages author, in this case `tool`.

tool_call_id string
Tool call that this message is responding to.

Function message object Deprecated

Hide properties

content string or null
The contents of the function message.

name string
The name of the function to call.

role string
The role of the messages author, in this case `function`.

parallel_tool_calls boolean
Whether to enable parallel function calling during tool use.

tools array
A list of tools the model may generate JSON inputs for.

Hide properties

function object

Hide properties

name string
The name of the function to be called. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

description string
A description of what the function does, used by the model to choose when and how to call the function.

parameters object
The parameters the functions accepts, described as a JSON Schema object. See the guide for examples, and the JSON Schema reference for documentation about the format.

Omitting `parameters` defines a function with an empty parameter list.

strict boolean or null
Whether to enable strict schema adherence when generating the function call. If set to true, the model will follow the exact schema defined in the `parameters` field. Only a subset of JSON Schema is supported when `strict` is `true`. Learn more about Structured Outputs in the function calling guide.

type string
The type of the tool. Currently, only `function` is supported.

non_preferred_output array

The non-preferred completion message for the output.

Hide possible types

Assistant message object

Messages sent by the model in response to user messages.

Hide properties

role string

The role of the messages author, in this case `assistant`.

audio object or null

Data about a previous audio response from the model. Learn more.

Hide properties

id string

Unique identifier for a previous audio response from the model.

content string or array

The contents of the assistant message. Required unless `tool_calls` or `function_call` is specified.

Hide possible types

Text content string

The contents of the assistant message.

Array of content parts array

An array of content parts with a defined type. Can be one or more of type `text`, or exactly one of type `refusal`.

Hide possible types

Text content part object

Learn about text inputs.

Hide properties

text string

The text content.

type string

The type of the content part.

Refusal content part object

Hide properties

refusal string
The refusal message generated by the model.

type string
The type of the content part.

function_call Deprecated object or null
Deprecated and replaced by `tool_calls`. The name and arguments of a function that should be called, as generated by the model.

Hide properties

arguments string
The arguments to call the function with, as generated by the model in JSON format. Note that the model does not always generate valid JSON, and may hallucinate parameters not defined by your function schema. Validate the arguments in your code before calling your function.

name string
The name of the function to call.

name string
An optional name for the participant. Provides the model information to differentiate between participants of the same role.

refusal string or null
The refusal message by the assistant.

tool_calls array
The tool calls generated by the model, such as function calls.

Hide properties

function object
The function that the model called.

Hide properties

arguments string
The arguments to call the function with, as generated by the model in JSON format. Note that the model does not always generate valid JSON, and may hallucinate parameters not defined by your function schema. Validate the arguments in your code before calling your function.

name string
The name of the function to call.

id string
The ID of the tool call.

type string
The type of the tool. Currently, only `function` is supported.

preferred_output array
The preferred completion message for the output.

Hide possible types

Assistant message object
Messages sent by the model in response to user messages.

Hide properties

role string
The role of the messages author, in this case `assistant`.

audio object or null
Data about a previous audio response from the model. Learn more.

Hide properties

id string
Unique identifier for a previous audio response from the model.

content string or array
The contents of the assistant message. Required unless `tool_calls` or `function_call` is specified.

Hide possible types

Text content string
The contents of the assistant message.

Array of content parts array
An array of content parts with a defined type. Can be one or more of type `text`, or exactly one of type `refusal`.

Hide possible types

Text content part object

Learn about text inputs.

Hide properties

text string

The text content.

type string

The type of the content part.

Refusal content part object

Hide properties

refusal string

The refusal message generated by the model.

type string

The type of the content part.

function_call Deprecated object or null

Deprecated and replaced by `tool_calls`. The name and arguments of a function that should be called, as generated by the model.

Hide properties

arguments string

The arguments to call the function with, as generated by the model in JSON format. Note that the model does not always generate valid JSON, and may hallucinate parameters not defined by your function schema. Validate the arguments in your code before calling your function.

name string

The name of the function to call.

name string

An optional name for the participant. Provides the model information to differentiate between participants of the same role.

refusal string or null

The refusal message by the assistant.

tool_calls array

The tool calls generated by the model, such as function calls.

Hide properties

function object
The function that the model called.

Hide properties

arguments string
The arguments to call the function with, as generated by the model in JSON format. Note that the model does not always generate valid JSON, and may hallucinate parameters not defined by your function schema. Validate the arguments in your code before calling your function.

name string
The name of the function to call.

id string
The ID of the tool call.

type string
The type of the tool. Currently, only `function` is supported.

OBJECT Training format for chat models using the preference method

```
{
  "input": {
    "messages": [
      { "role": "user", "content": "What is the weather in San Francisco?" }
    ]
  },
  "preferred_output": [
    {
      "role": "assistant",
      "content": "The weather in San Francisco is 70 degrees Fahrenheit."
    }
  ],
  "non_preferred_output": [
    {
      "role": "assistant",
      "content": "The weather in San Francisco is 21 degrees Celsius."
    }
  ]
```

```
}
```

Training format for reasoning models using the reinforcement method

Per-line training example for reinforcement fine-tuning. Note that `messages` and `tools` are the only reserved keywords. Any other arbitrary key-value data can be included on training datapoints and will be available to reference during grading under the `{{ item.XXX }}` template variable. Input messages may contain text or image content only. Audio and file input messages are not currently supported for fine-tuning.

messages array

Hide possible types

Developer message object

Developer-provided instructions that the model should follow, regardless of messages sent by the user. With o1 models and newer, `developer` messages replace the previous `system` messages.

Hide properties

content string or array

The contents of the developer message.

Hide possible types

Text content string

The contents of the developer message.

Array of content parts array

An array of content parts with a defined type. For developer messages, only type `text` is supported.

Hide properties

text string

The text content.

type string

The type of the content part.

role string

The role of the messages author, in this case `developer`.

name string
An optional name for the participant. Provides the model information to differentiate between participants of the same role.

User message object
Messages sent by an end user, containing prompts or additional context information.

Hide properties

content string or array
The contents of the user message.

Hide possible types

Text content string
The text contents of the message.

Array of content parts array
An array of content parts with a defined type. Supported options differ based on the model being used to generate the response. Can contain text, image, or audio inputs.

Hide possible types

Text content part object
Learn about text inputs.

Hide properties

text string
The text content.

type string
The type of the content part.

Image content part object
Learn about image inputs.

Hide properties

image_url object

Hide properties

url string
Either a URL of the image or the base64 encoded image data.

detail string
Specifies the detail level of the image. Learn more in the Vision guide.

type string
The type of the content part.

Audio content part object
Learn about audio inputs.

Hide properties

input_audio object

Hide properties

data string
Base64 encoded audio data.

format string
The format of the encoded audio data. Currently supports "wav" and "mp3".

type string
The type of the content part. Always `input_audio`.

File content part object
Learn about file inputs for text generation.

Hide properties

file object

Hide properties

file_data string
The base64 encoded file data, used when passing the file to the model as a string.

file_id string
The ID of an uploaded file to use as input.

filename string
The name of the file, used when passing the file to the model as a string.

type string
The type of the content part. Always `file`.

role string
The role of the messages author, in this case `user`.

name string
An optional name for the participant. Provides the model information to differentiate between participants of the same role.

Assistant message object
Messages sent by the model in response to user messages.

Hide properties

role string
The role of the messages author, in this case `assistant`.

audio object or null
Data about a previous audio response from the model. Learn more.

Hide properties

id string
Unique identifier for a previous audio response from the model.

content string or array
The contents of the assistant message. Required unless `tool_calls` or `function_call` is specified.

Hide possible types

Text content string
The contents of the assistant message.

Array of content parts array
An array of content parts with a defined type. Can be one or more of type `text`, or exactly one of type `refusal`.

Hide possible types

Text content part object
Learn about text inputs.

Hide properties

text string
The text content.

type string
The type of the content part.

Refusal content part object

Hide properties

refusal string
The refusal message generated by the model.

type string
The type of the content part.

function_call Deprecated object or null
Deprecated and replaced by `tool_calls`. The name and arguments of a function that should be called, as generated by the model.

Hide properties

arguments string
The arguments to call the function with, as generated by the model in JSON format. Note that the model does not always generate valid JSON, and may hallucinate parameters not defined by your function schema. Validate the arguments in your code before calling your function.

name string
The name of the function to call.

name string
An optional name for the participant. Provides the model information to differentiate between participants of the same role.

refusal string or null
The refusal message by the assistant.

tool_calls array
The tool calls generated by the model, such as function calls.

Hide properties

function object
The function that the model called.

Hide properties

arguments string
The arguments to call the function with, as generated by the model in JSON format. Note that the model does not always generate valid JSON, and may hallucinate parameters not defined by your function schema. Validate the arguments in your code before calling your function.

name string
The name of the function to call.

id string
The ID of the tool call.

type string
The type of the tool. Currently, only `function` is supported.

weight integer
Controls whether the assistant message is trained against (0 or 1)

Tool message object

Hide properties

content string or array
The contents of the tool message.

Hide possible types

Text content string
The contents of the tool message.

Array of content parts array
An array of content parts with a defined type. For tool messages, only type `text` is supported.

Hide possible types

Text content part object
Learn about text inputs.

Hide properties

text string

The text content.

type string
The type of the content part.

role string
The role of the messages author, in this case `tool`.

tool_call_id string
Tool call that this message is responding to.

tools array
A list of tools the model may generate JSON inputs for.

Hide properties

function object

Hide properties

name string
The name of the function to be called. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

description string
A description of what the function does, used by the model to choose when and how to call the function.

parameters object
The parameters the functions accepts, described as a JSON Schema object. See the guide for examples, and the JSON Schema reference for documentation about the format.

Omitting `parameters` defines a function with an empty parameter list.

strict boolean or null
Whether to enable strict schema adherence when generating the function call. If set to true, the model will follow the exact schema defined in the `parameters` field. Only a subset of JSON Schema is supported when `strict` is `true`. Learn more about Structured Outputs in the function calling guide.

type string
The type of the tool. Currently, only `function` is supported.

OBJECT Training format for reasoning models using the reinforcement method

```
{
  "messages": [
    {
      "role": "user",
      "content": "Your task is to take a chemical in SMILES format and predict the number
    },
  ],
  # Any other JSON data can be inserted into an example and referenced during RFT grading
  "reference_answer": {
    "donor_bond_counts": 5,
    "acceptor_bond_counts": 7
  }
}
```

The fine-tuning job object

The `fine_tuning.job` object represents a fine-tuning job that has been created through the API.

created_at integer
The Unix timestamp (in seconds) for when the fine-tuning job was created.

error object or null
For fine-tuning jobs that have `failed`, this will contain more information on the cause of the failure.

Hide properties

code string
A machine-readable error code.

message string
A human-readable error message.

param string or null
The parameter that was invalid, usually `training_file` or `validation_file`. This field will be null if the failure was not parameter-specific.

estimated_finish integer or null
The Unix timestamp (in seconds) for when the fine-tuning job is estimated to finish. The value will be null if the fine-tuning job is not running.

fine_tuned_model string or null
The name of the fine-tuned model that is being created. The value will be null if the fine-tuning job is still running.

finished_at integer or null
The Unix timestamp (in seconds) for when the fine-tuning job was finished. The value will be null if the fine-tuning job is still running.

hyperparameters object
The hyperparameters used for the fine-tuning job. This value will only be returned when running `supervised` jobs.

Hide properties

batch_size "auto" or integer
Number of examples in each batch. A larger batch size means that model parameters are updated less frequently, but with lower variance.

learning_rate_multiplier "auto" or number
Scaling factor for the learning rate. A smaller learning rate may be useful to avoid overfitting.

n_epochs "auto" or integer
The number of epochs to train the model for. An epoch refers to one full cycle through the training dataset.

id string
The object identifier, which can be referenced in the API endpoints.

integrations array or null
A list of integrations to enable for this fine-tuning job.

Hide possible types

Fine-Tuning Job Integration object

Hide properties

type string
The type of the integration being enabled for the fine-tuning job

wandb object
The settings for your integration with Weights and Biases. This payload specifies the project that metrics will be sent to. Optionally, you can set an explicit display name for your run, add tags to your run, and set a default entity (team, username, etc) to be associated with your run.

Hide properties

project string
The name of the project that the new run will be created under.

entity string or null
The entity to use for the run. This allows you to set the team or username of the WandB user that you would like associated with the run. If not set, the default entity for the registered WandB API key is used.

name string or null
A display name to set for the run. If not set, we will use the Job ID as the name.

tags array
A list of tags to be attached to the newly created run. These tags are passed through directly to WandB. Some default tags are generated by OpenAI: "openai/finetune", "openai/{base-model}", "openai/{ftjob-abcdef}".

metadata map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

method object
The method used for fine-tuning.

Hide properties

type string
The type of method. Is either `supervised`, `dpo`, or `reinforcement`.

dpo object

Configuration for the DPO fine-tuning method.

Hide properties

hyperparameters object
The hyperparameters used for the DPO fine-tuning job.

Hide properties

batch_size "auto" or integer
Number of examples in each batch. A larger batch size means that model parameters are updated less frequently, but with lower variance.

beta "auto" or number
The beta value for the DPO method. A higher beta value will increase the weight of the penalty between the policy and reference model.

learning_rate_multiplier "auto" or number
Scaling factor for the learning rate. A smaller learning rate may be useful to avoid overfitting.

n_epochs "auto" or integer
The number of epochs to train the model for. An epoch refers to one full cycle through the training dataset.

reinforcement object
Configuration for the reinforcement fine-tuning method.

Hide properties

grader object
The grader used for the fine-tuning job.

Hide possible types

StringCheckGrader object
A StringCheckGrader object that performs a string comparison between input and reference using a specified operation.

Hide properties

input string
The input text. This may include template strings.

name string
The name of the grader.

operation string

The string check operation to perform. One of `eq`, `ne`, `like`, or `ilike`.

`reference` string
The reference text. This may include template strings.

`type` string
The object type, which is always `string_check`.

TextSimilarityGrader object
A TextSimilarityGrader object which grades text based on similarity metrics.

Hide properties

`evaluation_metric` string
The evaluation metric to use. One of `fuzzy_match`, `bleu`, `gleu`, `meteor`, `rouge_1`, `rouge_2`, `rouge_3`, `rouge_4`, `rouge_5`, or `rouge_l`.

`input` string
The text being graded.

`name` string
The name of the grader.

`reference` string
The text being graded against.

`type` string
The type of grader.

PythonGrader object
A PythonGrader object that runs a python script on the input.

Hide properties

`name` string
The name of the grader.

`source` string
The source code of the python script.

`type` string
The object type, which is always `python`.

`image_tag` string

The image tag to use for the python script.

ScoreModelGrader object

A ScoreModelGrader object that uses a model to assign a score to the input.

Hide properties

input array

The input text. This may include template strings.

Hide properties

content string / object / array

Inputs to the model - can contain template strings.

Hide possible types

Text input string

A text input to the model.

Input text object

A text input to the model.

Hide properties

text string

The text input to the model.

type string

The type of the input item. Always `input_text`.

Output text object

A text output from the model.

Hide properties

text string

The text output from the model.

type string

The type of the output text. Always `output_text`.

Input image object

An image input to the model.

Hide properties

image_url string
The URL of the image input.

type string
The type of the image input. Always `input_image`.

detail string
The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

An array of Input text and Input image array
A list of inputs, each of which may be either an input text or input image object.

role string
The role of the message input. One of `user`, `assistant`, `system`, or `developer`.

type string
The type of the message input. Always `message`.

model string
The model to use for the evaluation.

name string
The name of the grader.

type string
The object type, which is always `score_model`.

range array
The range of the score. Defaults to `[0, 1]`.

sampling_params object
The sampling parameters for the model.

MultiGrader object
A MultiGrader object combines the output of multiple graders to produce a single score.

^ Hide properties

calculate_output string
A formula to calculate the output based on grader results.

graders object

Hide possible types

StringCheckGrader object
A StringCheckGrader object that performs a string comparison between input and reference using a specified operation.

Hide properties

input string
The input text. This may include template strings.

name string
The name of the grader.

operation string
The string check operation to perform. One of `eq`, `ne`, `like`, or `ilike`.

reference string
The reference text. This may include template strings.

type string
The object type, which is always `string_check`.

TextSimilarityGrader object
A TextSimilarityGrader object which grades text based on similarity metrics.

Hide properties

evaluation_metric string
The evaluation metric to use. One of `fuzzy_match`, `bleu`, `gleu`, `meteor`, `rouge_1`, `rouge_2`, `rouge_3`, `rouge_4`, `rouge_5`, or `rouge_l`.

input string
The text being graded.

name string
The name of the grader.

reference string
The text being graded against.

type string
The type of grader.

PythonGrader object
A PythonGrader object that runs a python script on the input.

Hide properties

name string
The name of the grader.

source string
The source code of the python script.

type string
The object type, which is always `python`.

image_tag string
The image tag to use for the python script.

ScoreModelGrader object
A ScoreModelGrader object that uses a model to assign a score to the input.

Hide properties

input array
The input text. This may include template strings.

Hide properties

content string / object / array
Inputs to the model - can contain template strings.

Hide possible types

Text input string
A text input to the model.

Input text object
A text input to the model.

Hide properties

text string
The text input to the model.

type string
The type of the input item. Always `input_text`.

Output text object
A text output from the model.

^ Hide properties

text string
The text output from the model.

type string
The type of the output text. Always `output_text`.

Input image object
An image input to the model.

^ Hide properties

image_url string
The URL of the image input.

type string
The type of the image input. Always `input_image`.

detail string
The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

An array of Input text and Input image array
A list of inputs, each of which may be either an input text or input image object.

role string
The role of the message input. One of `user`, `assistant`, `system`, or `developer`.

type string
The type of the message input. Always `message`.

model string
The model to use for the evaluation.

name string
The name of the grader.

type string
The object type, which is always `score_model`.

range array
The range of the score. Defaults to `[0, 1]`.

sampling_params object
The sampling parameters for the model.

LabelModelGrader object
A LabelModelGrader object which uses a model to assign labels to each item in the evaluation.

Hide properties

input array

Hide properties

content string / object / array
Inputs to the model - can contain template strings.

Hide possible types

Text input string
A text input to the model.

Input text object
A text input to the model.

Hide properties

text string
The text input to the model.

type string

The type of the input item. Always `input_text` .

Output text object
A text output from the model.

Hide properties

text string
The text output from the model.

type string
The type of the output text. Always `output_text` .

Input image object
An image input to the model.

Hide properties

image_url string
The URL of the image input.

type string
The type of the image input. Always `input_image` .

detail string
The detail level of the image to be sent to the model. One of `high` , `low` , or `auto` . Defaults to `auto` .

An array of Input text and Input image array
A list of inputs, each of which may be either an input text or input image object.

role string
The role of the message input. One of `user` , `assistant` , `system` , or `developer` .

type string
The type of the message input. Always `message` .

labels array
The labels to assign to each item in the evaluation.

model string
The model to use for the evaluation. Must support structured outputs.

name string
The name of the grader.

passing_labels array
The labels that indicate a passing result. Must be a subset of labels.

type string
The object type, which is always `label_model`.

name string
The name of the grader.

type string
The object type, which is always `multi`.

hyperparameters object
The hyperparameters used for the reinforcement fine-tuning job.

Hide properties

batch_size "auto" or integer
Number of examples in each batch. A larger batch size means that model parameters are updated less frequently, but with lower variance.

compute_multiplier "auto" or number
Multiplier on amount of compute used for exploring search space during training.

eval_interval "auto" or integer
The number of training steps between evaluation runs.

eval_samples "auto" or integer
Number of evaluation samples to generate per training step.

learning_rate_multiplier "auto" or number
Scaling factor for the learning rate. A smaller learning rate may be useful to avoid overfitting.

n_epochs "auto" or integer
The number of epochs to train the model for. An epoch refers to one full cycle through the

training dataset.

reasoning_effort string
Level of reasoning effort.

supervised object
Configuration for the supervised fine-tuning method.

Hide properties

hyperparameters object
The hyperparameters used for the fine-tuning job.

Hide properties

batch_size "auto" or integer
Number of examples in each batch. A larger batch size means that model parameters are updated less frequently, but with lower variance.

learning_rate_multiplier "auto" or number
Scaling factor for the learning rate. A smaller learning rate may be useful to avoid overfitting.

n_epochs "auto" or integer
The number of epochs to train the model for. An epoch refers to one full cycle through the training dataset.

model string
The base model that is being fine-tuned.

object string
The object type, which is always "fine_tuning.job".

organization_id string
The organization that owns the fine-tuning job.

result_files array
The compiled results file ID(s) for the fine-tuning job. You can retrieve the results with the Files API.

seed integer
The seed used for the fine-tuning job.

status string
The current status of the fine-tuning job, which can be either `validating_files`, `queued`, `running`, `succeeded`, `failed`, or `cancelled`.

trained_tokens integer or null
The total number of billable tokens processed by this fine-tuning job. The value will be null if the fine-tuning job is still running.

training_file string
The file ID used for training. You can retrieve the training data with the Files API.

validation_file string or null
The file ID used for validation. You can retrieve the validation results with the Files API.

OBJECT The fine-tuning job object

```
{
  "object": "fine_tuning.job",
  "id": "ftjob-abc123",
  "model": "davinci-002",
  "created_at": 1692661014,
  "finished_at": 1692661190,
  "fine_tuned_model": "ft:davinci-002:my-org:custom_suffix:7q8mpxmy",
  "organization_id": "org-123",
  "result_files": [
      "file-abc123"
  ],
  "status": "succeeded",
  "validation_file": null,
  "training_file": "file-abc123",
  "hyperparameters": {
      "n_epochs": 4,
      "batch_size": 1,
      "learning_rate_multiplier": 1.0
  },
  "trained_tokens": 5768,
  "integrations": [],
  "seed": 0,
  "estimated_finish": 0,
  "method": {
    "type": "supervised",
    "supervised": {
      "hyperparameters": {
        "n_epochs": 4,
```

```
        "batch_size": 1,
        "learning_rate_multiplier": 1.0
      }
    }
  },
  "metadata": {
    "key": "value"
  }
}
```

The fine-tuning job event object

Fine-tuning job event object

created_at integer
The Unix timestamp (in seconds) for when the fine-tuning job was created.

data object
The data associated with the event.

id string
The object identifier.

level string
The log level of the event.

message string
The message of the event.

object string
The object type, which is always "fine_tuning.job.event".

type string
The type of event.

OBJECT The fine-tuning job event object

```
{
  "object": "fine_tuning.job.event",
  "id": "ftevent-abc123"
```

```
  "created_at": 1677610602,
  "level": "info",
  "message": "Created fine-tuning job",
  "data": {},
  "type": "message"
}
```

The fine-tuning job checkpoint object

The `fine_tuning.job.checkpoint` object represents a model checkpoint for a fine-tuning job that is ready to use.

`created_at` integer
The Unix timestamp (in seconds) for when the checkpoint was created.

`fine_tuned_model_checkpoint` string
The name of the fine-tuned checkpoint model that is created.

`fine_tuning_job_id` string
The name of the fine-tuning job that this checkpoint was created from.

`id` string
The checkpoint identifier, which can be referenced in the API endpoints.

`metrics` object
Metrics at the step number during the fine-tuning job.

Hide properties

`full_valid_loss` number

`full_valid_mean_token_accuracy` number

`step` number

`train_loss` number

`train_mean_token_accuracy` number

`valid_loss` number

valid_mean_token_accuracy number

object string
The object type, which is always "fine_tuning.job.checkpoint".

step_number integer
The step number that the checkpoint was created at.

OBJECT The fine-tuning job checkpoint object

```
{
  "object": "fine_tuning.job.checkpoint",
  "id": "ftckpt_qtZ5Gyk4BLq1SfLFWp3RtO3P",
  "created_at": 1712211699,
  "fine_tuned_model_checkpoint": "ft:gpt-4o-mini-2024-07-18:my-org:custom_suffix:9ABel2dg:
  "fine_tuning_job_id": "ftjob-fpbNQ3H1GrMehXRf8cO97xTN",
  "metrics": {
    "step": 88,
    "train_loss": 0.478,
    "train_mean_token_accuracy": 0.924,
    "valid_loss": 10.112,
    "valid_mean_token_accuracy": 0.145,
    "full_valid_loss": 0.567,
    "full_valid_mean_token_accuracy": 0.944
  },
  "step_number": 88
}
```

The fine-tuned model checkpoint permission object

The `checkpoint.permission` object represents a permission for a fine-tuned model checkpoint.

created_at integer
The Unix timestamp (in seconds) for when the permission was created.

id string
The permission identifier, which can be referenced in the API endpoints.

object string
The object type, which is always "checkpoint.permission".

project_id string
The project identifier that the permission is for.

OBJECT The fine-tuned model checkpoint permission object

```
{
  "object": "checkpoint.permission",
  "id": "cp_zc4Q7MP6XxulcVzj4MZdwsAB",
  "created_at": 1712211699,
  "project_id": "proj_abGMw1llN8IrBb6SvvY5A1iH"
}
```

Graders

Manage and run graders in the OpenAI platform. Related guide: Graders

String Check Grader

A StringCheckGrader object that performs a string comparison between input and reference using a specified operation.

input string
The input text. This may include template strings.

name string
The name of the grader.

operation string
The string check operation to perform. One of `eq`, `ne`, `like`, or `ilike`.

reference string
The reference text. This may include template strings.

type string

The object type, which is always `string_check`.

OBJECT String Check Grader

```
{
  "type": "string_check",
  "name": "Example string check grader",
  "input": "{{sample.output_text}}",
  "reference": "{{item.label}}",
  "operation": "eq"
}
```

Text Similarity Grader

A TextSimilarityGrader object which grades text based on similarity metrics.

`evaluation_metric` string
The evaluation metric to use. One of `fuzzy_match`, `bleu`, `gleu`, `meteor`, `rouge_1`, `rouge_2`, `rouge_3`, `rouge_4`, `rouge_5`, or `rouge_l`.

`input` string
The text being graded.

`name` string
The name of the grader.

`reference` string
The text being graded against.

`type` string
The type of grader.

OBJECT Text Similarity Grader

```
{
  "type": "text_similarity",
  "name": "Example text similarity grader",
  "input": "{{sample.output_text}}",
  "reference": "{{item.label}}",
  "evaluation_metric": "fuzzy_match"
```

```
}
```

Score Model Grader

A ScoreModelGrader object that uses a model to assign a score to the input.

input array
The input text. This may include template strings.

Hide properties

content string / object / array
Inputs to the model - can contain template strings.

Hide possible types

Text input string
A text input to the model.

Input text object
A text input to the model.

Hide properties

text string
The text input to the model.

type string
The type of the input item. Always `input_text`.

Output text object
A text output from the model.

Hide properties

text string
The text output from the model.

type string
The type of the output text. Always `output_text`.

Input image object
An image input to the model.

Hide properties

image_url string
The URL of the image input.

type string
The type of the image input. Always `input_image`.

detail string
The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

An array of Input text and Input image array
A list of inputs, each of which may be either an input text or input image object.

role string
The role of the message input. One of `user`, `assistant`, `system`, or `developer`.

type string
The type of the message input. Always `message`.

model string
The model to use for the evaluation.

name string
The name of the grader.

range array
The range of the score. Defaults to `[0, 1]`.

sampling_params object
The sampling parameters for the model.

type string
The object type, which is always `score_model`.

OBJECT Score Model Grader

```
{
    "type": "score_model",
```

```
    "name": "Example score model grader",
    "input": [
        {
            "role": "user",
            "content": (
                "Score how close the reference answer is to the model answer. Score 1.0 if
                " Return just a floating point score\n\n"
                " Reference answer: {{item.label}}\n\n"
                " Model answer: {{sample.output_text}}"
            ),
        }
    ],
    "model": "gpt-4o-2024-08-06",
    "sampling_params": {
        "temperature": 1,
        "top_p": 1,
        "seed": 42,
    },
}
```

Label Model Grader

A LabelModelGrader object which uses a model to assign labels to each item in the evaluation.

input array

Hide properties

content string / object / array

Inputs to the model - can contain template strings.

Hide possible types

Text input string

A text input to the model.

Input text object

A text input to the model.

Hide properties

text string

The text input to the model.

type string

The type of the input item. Always `input_text`.

Output text object
A text output from the model.

Hide properties

text string
The text output from the model.

type string
The type of the output text. Always `output_text`.

Input image object
An image input to the model.

Hide properties

image_url string
The URL of the image input.

type string
The type of the image input. Always `input_image`.

detail string
The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

An array of Input text and Input image array
A list of inputs, each of which may be either an input text or input image object.

role string
The role of the message input. One of `user`, `assistant`, `system`, or `developer`.

type string
The type of the message input. Always `message`.

labels array
The labels to assign to each item in the evaluation.

model string
The model to use for the evaluation. Must support structured outputs.

name string
The name of the grader.

passing_labels array
The labels that indicate a passing result. Must be a subset of labels.

type string
The object type, which is always `label_model`.

OBJECT Label Model Grader

```
{
  "name": "First label grader",
  "type": "label_model",
  "model": "gpt-4o-2024-08-06",
  "input": [
    {
      "type": "message",
      "role": "system",
      "content": {
        "type": "input_text",
        "text": "Classify the sentiment of the following statement as one of positive, neu
      }
    },
    {
      "type": "message",
      "role": "user",
      "content": {
        "type": "input_text",
        "text": "Statement: {{item.response}}"
      }
    }
  ],
  "passing_labels": [
    "positive"
  ],
  "labels": [
    "positive",
    "neutral",
    "negative"
  ]
```

```
}
```

Python Grader

A PythonGrader object that runs a python script on the input.

`image_tag` string
The image tag to use for the python script.

`name` string
The name of the grader.

`source` string
The source code of the python script.

`type` string
The object type, which is always `python`.

OBJECT Python Grader

```
{
  "type": "python",
  "name": "Example python grader",
  "image_tag": "2025-05-08",
  "source": """
def grade(sample: dict, item: dict) -> float:
    \"""
    Returns 1.0 if `output_text` equals `label`, otherwise 0.0.
    \"""
    output = sample.get("output_text")
    label = item.get("label")
    return 1.0 if output == label else 0.0
""",
}
```

Multi Grader

A MultiGrader object combines the output of multiple graders to produce a single score.

calculate_output string
A formula to calculate the output based on grader results.

graders object

Hide possible types

StringCheckGrader object
A StringCheckGrader object that performs a string comparison between input and reference using a specified operation.

Hide properties

input string
The input text. This may include template strings.

name string
The name of the grader.

operation string
The string check operation to perform. One of `eq`, `ne`, `like`, or `ilike`.

reference string
The reference text. This may include template strings.

type string
The object type, which is always `string_check`.

TextSimilarityGrader object
A TextSimilarityGrader object which grades text based on similarity metrics.

Hide properties

evaluation_metric string
The evaluation metric to use. One of `fuzzy_match`, `bleu`, `gleu`, `meteor`, `rouge_1`, `rouge_2`, `rouge_3`, `rouge_4`, `rouge_5`, or `rouge_l`.

input string
The text being graded.

name string
The name of the grader.

reference string

The text being graded against.

type string
The type of grader.

PythonGrader object
A PythonGrader object that runs a python script on the input.

Hide properties

name string
The name of the grader.

source string
The source code of the python script.

type string
The object type, which is always `python`.

image_tag string
The image tag to use for the python script.

ScoreModelGrader object
A ScoreModelGrader object that uses a model to assign a score to the input.

Hide properties

input array
The input text. This may include template strings.

Hide properties

content string / object / array
Inputs to the model - can contain template strings.

Hide possible types

Text input string
A text input to the model.

Input text object
A text input to the model.

Hide properties

text string
The text input to the model.

type string
The type of the input item. Always `input_text` .

Output text object
A text output from the model.

^ Hide properties

text string
The text output from the model.

type string
The type of the output text. Always `output_text` .

Input image object
An image input to the model.

^ Hide properties

image_url string
The URL of the image input.

type string
The type of the image input. Always `input_image` .

detail string
The detail level of the image to be sent to the model. One of `high` , `low` , or `auto` . Defaults to `auto` .

An array of Input text and Input image array
A list of inputs, each of which may be either an input text or input image object.

role string
The role of the message input. One of `user` , `assistant` , `system` , or `developer` .

type string
The type of the message input. Always `message` .

model string
The model to use for the evaluation.

name string
The name of the grader.

type string
The object type, which is always `score_model`.

range array
The range of the score. Defaults to `[0, 1]`.

sampling_params object
The sampling parameters for the model.

LabelModelGrader object
A LabelModelGrader object which uses a model to assign labels to each item in the evaluation.

Hide properties

input array

Hide properties

content string / object / array
Inputs to the model - can contain template strings.

Hide possible types

Text input string
A text input to the model.

Input text object
A text input to the model.

Hide properties

text string
The text input to the model.

type string
The type of the input item. Always `input_text`.

Output text object
A text output from the model.

Hide properties

text string
The text output from the model.

type string
The type of the output text. Always `output_text`.

Input image object
An image input to the model.

Hide properties

image_url string
The URL of the image input.

type string
The type of the image input. Always `input_image`.

detail string
The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

An array of Input text and Input image array
A list of inputs, each of which may be either an input text or input image object.

role string
The role of the message input. One of `user`, `assistant`, `system`, or `developer`.

type string
The type of the message input. Always `message`.

labels array
The labels to assign to each item in the evaluation.

model string
The model to use for the evaluation. Must support structured outputs.

name string
The name of the grader.

passing_labels array

The labels that indicate a passing result. Must be a subset of labels.

type string
The object type, which is always `label_model`.

name string
The name of the grader.

type string
The object type, which is always `multi`.

OBJECT Multi Grader

```
{
  "type": "multi",
  "name": "example multi grader",
  "graders": [
    {
      "type": "text_similarity",
      "name": "example text similarity grader",
      "input": "The graded text",
      "reference": "The reference text",
      "evaluation_metric": "fuzzy_match"
    },
    {
      "type": "string_check",
      "name": "Example string check grader",
      "input": "{{sample.output_text}}",
      "reference": "{{item.label}}",
      "operation": "eq"
    }
  ],
  "calculate_output": "0.5 * text_similarity_score +  0.5 * string_check_score)"
}
```

Run grader Beta

`POST https://api.openai.com/v1/fine_tuning/alpha/graders/run`

Run a grader.

Request body

grader object Required

The grader used for the fine-tuning job.

Hide possible types

StringCheckGrader object

A StringCheckGrader object that performs a string comparison between input and reference using a specified operation.

Hide properties

input string Required

The input text. This may include template strings.

name string Required

The name of the grader.

operation string Required

The string check operation to perform. One of `eq`, `ne`, `like`, or `ilike`.

reference string Required

The reference text. This may include template strings.

type string Required

The object type, which is always `string_check`.

TextSimilarityGrader object

A TextSimilarityGrader object which grades text based on similarity metrics.

Hide properties

evaluation_metric string Required

The evaluation metric to use. One of `fuzzy_match`, `bleu`, `gleu`, `meteor`, `rouge_1`, `rouge_2`, `rouge_3`, `rouge_4`, `rouge_5`, or `rouge_l`.

input string Required

The text being graded.

name string Required

The name of the grader.

reference string Required
The text being graded against.

type string Required
The type of grader.

PythonGrader object
A PythonGrader object that runs a python script on the input.

Hide properties

name string Required
The name of the grader.

source string Required
The source code of the python script.

type string Required
The object type, which is always `python`.

image_tag string Optional
The image tag to use for the python script.

ScoreModelGrader object
A ScoreModelGrader object that uses a model to assign a score to the input.

Hide properties

input array Required
The input text. This may include template strings.

Hide properties

content string / object / array Required
Inputs to the model - can contain template strings.

Hide possible types

Text input string
A text input to the model.

Input text object
A text input to the model.

Hide properties

text string Required
The text input to the model.

type string Required
The type of the input item. Always `input_text`.

Output text object
A text output from the model.

^ Hide properties

text string Required
The text output from the model.

type string Required
The type of the output text. Always `output_text`.

Input image object
An image input to the model.

^ Hide properties

image_url string Required
The URL of the image input.

type string Required
The type of the image input. Always `input_image`.

detail string Optional
The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

An array of Input text and Input image array
A list of inputs, each of which may be either an input text or input image object.

role string Required
The role of the message input. One of `user`, `assistant`, `system`, or `developer`.

type string Optional
The type of the message input. Always `message`.

model string Required
The model to use for the evaluation.

name string Required
The name of the grader.

type string Required
The object type, which is always `score_model`.

range array Optional
The range of the score. Defaults to `[0, 1]`.

sampling_params object Optional
The sampling parameters for the model.

MultiGrader object
A MultiGrader object combines the output of multiple graders to produce a single score.

Hide properties

calculate_output string Required
A formula to calculate the output based on grader results.

graders object Required

Hide possible types

StringCheckGrader object
A StringCheckGrader object that performs a string comparison between input and reference using a specified operation.

Hide properties

input string Required
The input text. This may include template strings.

name string Required
The name of the grader.

operation string Required
The string check operation to perform. One of `eq`, `ne`, `like`, or `ilike`.

reference string Required
The reference text. This may include template strings.

type string Required
The object type, which is always `string_check`.

TextSimilarityGrader object
A TextSimilarityGrader object which grades text based on similarity metrics.

Hide properties

evaluation_metric string Required
The evaluation metric to use. One of `fuzzy_match`, `bleu`, `gleu`, `meteor`, `rouge_1`, `rouge_2`, `rouge_3`, `rouge_4`, `rouge_5`, or `rouge_l`.

input string Required
The text being graded.

name string Required
The name of the grader.

reference string Required
The text being graded against.

type string Required
The type of grader.

PythonGrader object
A PythonGrader object that runs a python script on the input.

Hide properties

name string Required
The name of the grader.

source string Required
The source code of the python script.

type string Required
The object type, which is always `python`.

image_tag string Optional
The image tag to use for the python script.

ScoreModelGrader object

A ScoreModelGrader object that uses a model to assign a score to the input.

Hide properties

input array Required

The input text. This may include template strings.

Hide properties

content string / object / array Required

Inputs to the model - can contain template strings.

Hide possible types

Text input string

A text input to the model.

Input text object

A text input to the model.

Hide properties

text string Required

The text input to the model.

type string Required

The type of the input item. Always `input_text`.

Output text object

A text output from the model.

Hide properties

text string Required

The text output from the model.

type string Required

The type of the output text. Always `output_text`.

Input image object

An image input to the model.

Hide properties

image_url string Required

The URL of the image input.

type string Required

The type of the image input. Always `input_image`.

`detail` string Optional
The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

An array of Input text and Input image array
A list of inputs, each of which may be either an input text or input image object.

`role` string Required
The role of the message input. One of `user`, `assistant`, `system`, or `developer`.

`type` string Optional
The type of the message input. Always `message`.

`model` string Required
The model to use for the evaluation.

`name` string Required
The name of the grader.

`type` string Required
The object type, which is always `score_model`.

`range` array Optional
The range of the score. Defaults to `[0, 1]`.

`sampling_params` object Optional
The sampling parameters for the model.

LabelModelGrader object
A LabelModelGrader object which uses a model to assign labels to each item in the evaluation.

Hide properties

`input` array Required

Hide properties

`content` string / object / array Required
Inputs to the model - can contain template strings.

Hide possible types

Text input string

A text input to the model.

Input text object

A text input to the model.

Hide properties

text string Required

The text input to the model.

type string Required

The type of the input item. Always `input_text`.

Output text object

A text output from the model.

Hide properties

text string Required

The text output from the model.

type string Required

The type of the output text. Always `output_text`.

Input image object

An image input to the model.

Hide properties

image_url string Required

The URL of the image input.

type string Required

The type of the image input. Always `input_image`.

detail string Optional

The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

An array of Input text and Input image array

A list of inputs, each of which may be either an input text or input image object.

role string Required
The role of the message input. One of `user`, `assistant`, `system`, or `developer`.

type string Optional
The type of the message input. Always `message`.

labels array Required
The labels to assign to each item in the evaluation.

model string Required
The model to use for the evaluation. Must support structured outputs.

name string Required
The name of the grader.

passing_labels array Required
The labels that indicate a passing result. Must be a subset of labels.

type string Required
The object type, which is always `label_model`.

name string Required
The name of the grader.

type string Required
The object type, which is always `multi`.

model_sample string Required
The model sample to be evaluated. This value will be used to populate the `sample` namespace. See the guide for more details. The `output_json` variable will be populated if the model sample is a valid JSON string.

item object Optional
The dataset item provided to the grader. This will be used to populate the `item` namespace. See the guide for more details.

Returns

The results from the grader run.

Example request curl

```curl
curl -X POST https://api.openai.com/v1/fine_tuning/alpha/graders/run \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -d '{
    "grader": {
      "type": "score_model",
      "name": "Example score model grader",
      "input": [
        {
          "role": "user",
          "content": "Score how close the reference answer is to the model answer. Score 1
        }
      ],
      "model": "gpt-4o-2024-08-06",
      "sampling_params": {
        "temperature": 1,
        "top_p": 1,
        "seed": 42
      }
    },
    "item": {
      "reference_answer": "fuzzy wuzzy was a bear"
    },
    "model_sample": "fuzzy wuzzy was a bear"
  }'
```

Response

```response
{
  "reward": 1.0,
  "metadata": {
    "name": "Example score model grader",
    "type": "score_model",
    "errors": {
      "formula_parse_error": false,
      "sample_parse_error": false,
      "truncated_observation_error": false,
      "unresponsive_reward_error": false,
      "invalid_variable_error": false,
      "other_error": false,
```

```
      "python_grader_server_error": false,
      "python_grader_server_error_type": null,
      "python_grader_runtime_error": false,
      "python_grader_runtime_error_details": null,
      "model_grader_server_error": false,
      "model_grader_refusal_error": false,
      "model_grader_parse_error": false,
      "model_grader_server_error_details": null
    },
    "execution_time": 4.365238428115845,
    "scores": {},
    "token_usage": {
      "prompt_tokens": 190,
      "total_tokens": 324,
      "completion_tokens": 134,
      "cached_tokens": 0
    },
    "sampled_model_name": "gpt-4o-2024-08-06"
  },
  "sub_rewards": {},
  "model_grader_token_usage_per_model": {
    "gpt-4o-2024-08-06": {
      "prompt_tokens": 190,
      "total_tokens": 324,
      "completion_tokens": 134,
      "cached_tokens": 0
    }
  }
}
```

Validate grader Beta

```
POST https://api.openai.com/v1/fine_tuning/alpha/graders/validate
```

Validate a grader.

Request body

grader object Required

The grader used for the fine-tuning job.

^ Hide possible types

StringCheckGrader object

A StringCheckGrader object that performs a string comparison between input and reference using a specified operation.

Hide properties

input string Required

The input text. This may include template strings.

name string Required

The name of the grader.

operation string Required

The string check operation to perform. One of `eq`, `ne`, `like`, or `ilike`.

reference string Required

The reference text. This may include template strings.

type string Required

The object type, which is always `string_check`.

TextSimilarityGrader object

A TextSimilarityGrader object which grades text based on similarity metrics.

Hide properties

evaluation_metric string Required

The evaluation metric to use. One of `fuzzy_match`, `bleu`, `gleu`, `meteor`, `rouge_1`, `rouge_2`, `rouge_3`, `rouge_4`, `rouge_5`, or `rouge_l`.

input string Required

The text being graded.

name string Required

The name of the grader.

reference string Required

The text being graded against.

type string Required

The type of grader.

PythonGrader object

A PythonGrader object that runs a python script on the input.

Hide properties

name string Required
The name of the grader.

source string Required
The source code of the python script.

type string Required
The object type, which is always `python`.

image_tag string Optional
The image tag to use for the python script.

ScoreModelGrader object
A ScoreModelGrader object that uses a model to assign a score to the input.

Hide properties

input array Required
The input text. This may include template strings.

Hide properties

content string / object / array Required
Inputs to the model - can contain template strings.

Hide possible types

Text input string
A text input to the model.

Input text object
A text input to the model.

Hide properties

text string Required
The text input to the model.

type string Required
The type of the input item. Always `input_text`.

Output text object
A text output from the model.

Hide properties

text string Required
The text output from the model.

type string Required
The type of the output text. Always `output_text` .

Input image object
An image input to the model.

Hide properties

image_url string Required
The URL of the image input.

type string Required
The type of the image input. Always `input_image` .

detail string Optional
The detail level of the image to be sent to the model. One of `high` , `low` , or `auto` . Defaults to `auto` .

An array of Input text and Input image array
A list of inputs, each of which may be either an input text or input image object.

role string Required
The role of the message input. One of `user` , `assistant` , `system` , or `developer` .

type string Optional
The type of the message input. Always `message` .

model string Required
The model to use for the evaluation.

name string Required
The name of the grader.

type string Required
The object type, which is always `score_model` .

range array Optional
The range of the score. Defaults to `[0, 1]`.

sampling_params object Optional
The sampling parameters for the model.

MultiGrader object
A MultiGrader object combines the output of multiple graders to produce a single score.

Hide properties

calculate_output string Required
A formula to calculate the output based on grader results.

graders object Required

Hide possible types

StringCheckGrader object
A StringCheckGrader object that performs a string comparison between input and reference using a specified operation.

Hide properties

input string Required
The input text. This may include template strings.

name string Required
The name of the grader.

operation string Required
The string check operation to perform. One of `eq`, `ne`, `like`, or `ilike`.

reference string Required
The reference text. This may include template strings.

type string Required
The object type, which is always `string_check`.

TextSimilarityGrader object
A TextSimilarityGrader object which grades text based on similarity metrics.

Hide properties

evaluation_metric string Required

The evaluation metric to use. One of `fuzzy_match`, `bleu`, `gleu`, `meteor`, `rouge_1`, `rouge_2`, `rouge_3`, `rouge_4`, `rouge_5`, or `rouge_l`.

`input` string Required
The text being graded.

`name` string Required
The name of the grader.

`reference` string Required
The text being graded against.

`type` string Required
The type of grader.

PythonGrader object
A PythonGrader object that runs a python script on the input.

Hide properties

`name` string Required
The name of the grader.

`source` string Required
The source code of the python script.

`type` string Required
The object type, which is always `python`.

`image_tag` string Optional
The image tag to use for the python script.

ScoreModelGrader object
A ScoreModelGrader object that uses a model to assign a score to the input.

Hide properties

`input` array Required
The input text. This may include template strings.

Hide properties

`content` string / object / array Required
Inputs to the model - can contain template strings.

Hide possible types

Text input string
A text input to the model.

Input text object
A text input to the model.

Hide properties

text string Required
The text input to the model.

type string Required
The type of the input item. Always `input_text`.

Output text object
A text output from the model.

Hide properties

text string Required
The text output from the model.

type string Required
The type of the output text. Always `output_text`.

Input image object
An image input to the model.

Hide properties

image_url string Required
The URL of the image input.

type string Required
The type of the image input. Always `input_image`.

detail string Optional
The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

An array of Input text and Input image array
A list of inputs, each of which may be either an input text or input image object.

role string Required
The role of the message input. One of `user`, `assistant`, `system`, or `developer`.

type string Optional
The type of the message input. Always `message`.

model string Required
The model to use for the evaluation.

name string Required
The name of the grader.

type string Required
The object type, which is always `score_model`.

range array Optional
The range of the score. Defaults to `[0, 1]`.

sampling_params object Optional
The sampling parameters for the model.

LabelModelGrader object
A LabelModelGrader object which uses a model to assign labels to each item in the evaluation.

Hide properties

input array Required

Hide properties

content string / object / array Required
Inputs to the model - can contain template strings.

Hide possible types

Text input string
A text input to the model.

Input text object
A text input to the model.

Hide properties

text string Required
The text input to the model.

type string Required
The type of the input item. Always `input_text`.

Output text object
A text output from the model.

Hide properties

text string Required
The text output from the model.

type string Required
The type of the output text. Always `output_text`.

Input image object
An image input to the model.

Hide properties

image_url string Required
The URL of the image input.

type string Required
The type of the image input. Always `input_image`.

detail string Optional
The detail level of the image to be sent to the model. One of `high`, `low`, or `auto`. Defaults to `auto`.

An array of Input text and Input image array
A list of inputs, each of which may be either an input text or input image object.

role string Required
The role of the message input. One of `user`, `assistant`, `system`, or `developer`.

type string Optional
The type of the message input. Always `message`.

labels array Required
The labels to assign to each item in the evaluation.

model string Required
The model to use for the evaluation. Must support structured outputs.

name string Required
The name of the grader.

passing_labels array Required
The labels that indicate a passing result. Must be a subset of labels.

type string Required
The object type, which is always `label_model`.

name string Required
The name of the grader.

type string Required
The object type, which is always `multi`.

Returns

The validated grader object.

Example request curl

```
curl https://api.openai.com/v1/fine_tuning/alpha/graders/validate \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "grader": {
      "type": "string_check",
      "name": "Example string check grader",
      "input": "{{sample.output_text}}",
      "reference": "{{item.label}}",
      "operation": "eq"
    }
  }'
```

Response

```
{
  "grader": {
    "type": "string_check",
    "name": "Example string check grader",
    "input": "{{sample.output_text}}",
    "reference": "{{item.label}}",
    "operation": "eq"
  }
}
```

Batch

Create large batches of API requests for asynchronous processing. The Batch API returns completions within 24 hours for a 50% discount. Related guide: Batch

Create batch

`POST https://api.openai.com/v1/batches`

Creates and executes a batch from an uploaded file of requests

Request body

completion_window string Required
The time frame within which the batch should be processed. Currently only `24h` is supported.

endpoint string Required
The endpoint to be used for all requests in the batch. Currently `/v1/responses`, `/v1/chat/completions`, `/v1/embeddings`, and `/v1/completions` are supported. Note that `/v1/embeddings` batches are also restricted to a maximum of 50,000 embedding inputs across all requests in the batch.

input_file_id string Required
The ID of an uploaded file that contains requests for the new batch.

See upload file for how to upload a file.

Your input file must be formatted as a JSONL file, and must be uploaded with the purpose `batch`. The file can contain up to 50,000 requests, and can be up to 200 MB in size.

metadata map Optional
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

Returns

The created Batch object.

Example request curl

```curl
curl https://api.openai.com/v1/batches \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "input_file_id": "file-abc123",
    "endpoint": "/v1/chat/completions",
    "completion_window": "24h"
  }'
```

Response

```
{
  "id": "batch_abc123",
```

```
  "object": "batch",
  "endpoint": "/v1/chat/completions",
  "errors": null,
  "input_file_id": "file-abc123",
  "completion_window": "24h",
  "status": "validating",
  "output_file_id": null,
  "error_file_id": null,
  "created_at": 1711471533,
  "in_progress_at": null,
  "expires_at": null,
  "finalizing_at": null,
  "completed_at": null,
  "failed_at": null,
  "expired_at": null,
  "cancelling_at": null,
  "cancelled_at": null,
  "request_counts": {
    "total": 0,
    "completed": 0,
    "failed": 0
  },
  "metadata": {
    "customer_id": "user_123456789",
    "batch_description": "Nightly eval job",
  }
}
```

Retrieve batch

```
GET https://api.openai.com/v1/batches/{batch_id}
```

Retrieves a batch.

Path parameters

batch_id string Required

The ID of the batch to retrieve.

Returns

The Batch object matching the specified ID.

Example request curl

```
curl https://api.openai.com/v1/batches/batch_abc123 \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
```

Response

```
{
  "id": "batch_abc123",
  "object": "batch",
  "endpoint": "/v1/completions",
  "errors": null,
  "input_file_id": "file-abc123",
  "completion_window": "24h",
  "status": "completed",
  "output_file_id": "file-cvaTdG",
  "error_file_id": "file-HOWS94",
  "created_at": 1711471533,
  "in_progress_at": 1711471538,
  "expires_at": 1711557933,
  "finalizing_at": 1711493133,
  "completed_at": 1711493163,
  "failed_at": null,
  "expired_at": null,
  "cancelling_at": null,
  "cancelled_at": null,
  "request_counts": {
    "total": 100,
    "completed": 95,
    "failed": 5
  },
  "metadata": {
    "customer_id": "user_123456789",
    "batch_description": "Nightly eval job",
  }
}
```

Cancel batch

```
POST https://api.openai.com/v1/batches/{batch_id}/cancel
```

Cancels an in-progress batch. The batch will be in status `cancelling` for up to 10 minutes, before changing to `cancelled`, where it will have partial results (if any) available in the output file.

Path parameters

batch_id string Required
The ID of the batch to cancel.

Returns

The Batch object matching the specified ID.

Example request curl

```curl
curl https://api.openai.com/v1/batches/batch_abc123/cancel \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -X POST
```

Response

```
{
  "id": "batch_abc123",
  "object": "batch",
  "endpoint": "/v1/chat/completions",
  "errors": null,
  "input_file_id": "file-abc123",
  "completion_window": "24h",
  "status": "cancelling",
  "output_file_id": null,
  "error_file_id": null,
  "created_at": 1711471533,
  "in_progress_at": 1711471538,
  "expires_at": 1711557933,
  "finalizing_at": null,
  "completed_at": null,
  "failed_at": null,
  "expired_at": null,
```

```
  "cancelling_at": 1711475133,
  "cancelled_at": null,
  "request_counts": {
    "total": 100,
    "completed": 23,
    "failed": 1
  },
  "metadata": {
    "customer_id": "user_123456789",
    "batch_description": "Nightly eval job",
  }
}
```

List batch

`GET https://api.openai.com/v1/batches`

List your organization's batches.

Query parameters

after string Optional

A cursor for use in pagination. `after` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, ending with obj_foo, your subsequent call can include after=obj_foo in order to fetch the next page of the list.

limit integer Optional Defaults to 20

A limit on the number of objects to be returned. Limit can range between 1 and 100, and the default is 20.

Returns

A list of paginated Batch objects.

Example request curl

```
curl https://api.openai.com/v1/batches?limit=2 \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
  "object": "list",
  "data": [
    {
      "id": "batch_abc123",
      "object": "batch",
      "endpoint": "/v1/chat/completions",
      "errors": null,
      "input_file_id": "file-abc123",
      "completion_window": "24h",
      "status": "completed",
      "output_file_id": "file-cvaTdG",
      "error_file_id": "file-HOWS94",
      "created_at": 1711471533,
      "in_progress_at": 1711471538,
      "expires_at": 1711557933,
      "finalizing_at": 1711493133,
      "completed_at": 1711493163,
      "failed_at": null,
      "expired_at": null,
      "cancelling_at": null,
      "cancelled_at": null,
      "request_counts": {
        "total": 100,
        "completed": 95,
        "failed": 5
      },
      "metadata": {
        "customer_id": "user_123456789",
        "batch_description": "Nightly job",
      }
    },
    { ... },
  ],
  "first_id": "batch_abc123",
  "last_id": "batch_abc456",
  "has_more": true
}
```

The batch object

cancelled_at integer
The Unix timestamp (in seconds) for when the batch was cancelled.

cancelling_at integer
The Unix timestamp (in seconds) for when the batch started cancelling.

completed_at integer
The Unix timestamp (in seconds) for when the batch was completed.

completion_window string
The time frame within which the batch should be processed.

created_at integer
The Unix timestamp (in seconds) for when the batch was created.

endpoint string
The OpenAI API endpoint used by the batch.

error_file_id string
The ID of the file containing the outputs of requests with errors.

errors object
Hide properties

data array
Hide properties

code string
An error code identifying the error type.

line integer or null
The line number of the input file where the error occurred, if applicable.

message string
A human-readable message providing more details about the error.

param string or null
The name of the parameter that caused the error, if applicable.

object string

The object type, which is always `list`.

`expired_at` integer
The Unix timestamp (in seconds) for when the batch expired.

`expires_at` integer
The Unix timestamp (in seconds) for when the batch will expire.

`failed_at` integer
The Unix timestamp (in seconds) for when the batch failed.

`finalizing_at` integer
The Unix timestamp (in seconds) for when the batch started finalizing.

`id` string

`in_progress_at` integer
The Unix timestamp (in seconds) for when the batch started processing.

`input_file_id` string
The ID of the input file for the batch.

`metadata` map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

`object` string
The object type, which is always `batch`.

`output_file_id` string
The ID of the file containing the outputs of successfully executed requests.

`request_counts` object
The request counts for different statuses within the batch.

^ Hide properties

completed integer
Number of requests that have been completed successfully.

failed integer
Number of requests that have failed.

total integer
Total number of requests in the batch.

status string
The current status of the batch.

OBJECT The batch object

```
{
  "id": "batch_abc123",
  "object": "batch",
  "endpoint": "/v1/completions",
  "errors": null,
  "input_file_id": "file-abc123",
  "completion_window": "24h",
  "status": "completed",
  "output_file_id": "file-cvaTdG",
  "error_file_id": "file-HOWS94",
  "created_at": 1711471533,
  "in_progress_at": 1711471538,
  "expires_at": 1711557933,
  "finalizing_at": 1711493133,
  "completed_at": 1711493163,
  "failed_at": null,
  "expired_at": null,
  "cancelling_at": null,
  "cancelled_at": null,
  "request_counts": {
    "total": 100,
    "completed": 95,
    "failed": 5
  },
  "metadata": {
    "customer_id": "user_123456789",
    "batch_description": "Nightly eval job",
  }
}
```

The request input object

The per-line object of the batch input file

custom_id string
A developer-provided per-request id that will be used to match outputs to inputs. Must be unique for each request in a batch.

method string
The HTTP method to be used for the request. Currently only `POST` is supported.

url string
The OpenAI API relative URL to be used for the request. Currently `/v1/chat/completions`, `/v1/embeddings`, and `/v1/completions` are supported.

```
OBJECT The request input object
```

```
{"custom_id": "request-1", "method": "POST", "url": "/v1/chat/completions", "body": {"model":
```

The request output object

The per-line object of the batch output and error files

custom_id string
A developer-provided per-request id that will be used to match outputs to inputs.

error object or null
For requests that failed with a non-HTTP error, this will contain more information on the cause of the failure.

Hide properties

code string
A machine-readable error code.

message string
A human-readable error message.

id string

response object or null

Hide properties

body map
The JSON body of the response

request_id string
An unique identifier for the OpenAI API request. Please include this request ID when contacting support.

status_code integer
The HTTP status code of the response

```
OBJECT The request output object
```


```
{"id": "batch_req_wnaDys", "custom_id": "request-2", "response": {"status_code": 200, "reques
```


Files

Files are used to upload documents that can be used with features like Assistants, Fine-tuning, and Batch API.

Upload file

```
POST https://api.openai.com/v1/files
```

Upload a file that can be used across various endpoints. Individual files can be up to 512 MB, and the size of all files uploaded by one organization can be up to 100 GB.
The Assistants API supports files up to 2 million tokens and of specific file types. See the Assistants Tools guide for details.
The Fine-tuning API only supports `.jsonl` files. The input also has certain required formats for fine-tuning chat or completions models.
The Batch API only supports `.jsonl` files up to 200 MB in size. The input also has a specific required format.
Please contact us if you need to increase these storage limits.

Request body

file file Required
The File object (not file name) to be uploaded.

purpose string Required
The intended purpose of the uploaded file. One of: - `assistants` : Used in the Assistants API - `batch` : Used in the Batch API - `fine-tune` : Used for fine-tuning - `vision` : Images used for vision fine-tuning - `user_data` : Flexible file type for any purpose - `evals` : Used for eval data sets

Returns

The uploaded File object.

Example request curl

```
curl https://api.openai.com/v1/files \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -F purpose="fine-tune" \
  -F file="@mydata.jsonl"
```

Response

```
{
  "id": "file-abc123",
  "object": "file",
  "bytes": 120000,
  "created_at": 1677610602,
  "filename": "mydata.jsonl",
  "purpose": "fine-tune",
}
```

List files

`GET https://api.openai.com/v1/files`

Returns a list of files.

Query parameters

after string Optional
A cursor for use in pagination. `after` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, ending with obj_foo, your subsequent call can include after=obj_foo in order to fetch the next page of the list.

limit integer Optional Defaults to 10000
A limit on the number of objects to be returned. Limit can range between 1 and 10,000, and the default is 10,000.

order string Optional Defaults to desc
Sort order by the `created_at` timestamp of the objects. `asc` for ascending order and `desc` for descending order.

purpose string Optional
Only return files with the given purpose.

Returns

A list of File objects.

Example request curl

```curl
curl https://api.openai.com/v1/files \
  -H "Authorization: Bearer $OPENAI_API_KEY"
```

Response

```json
{
  "data": [
    {
      "id": "file-abc123",
      "object": "file",
      "bytes": 175,
      "created_at": 1613677385,
      "filename": "salesOverview.pdf",
      "purpose": "assistants",
    },
    {
      "id": "file-abc123",
      "object": "file",

```

```
      "bytes": 140,
      "created_at": 1613779121,
      "filename": "puppy.jsonl",
      "purpose": "fine-tune",
    }
  ],
  "object": "list"
}
```

Retrieve file

```
GET https://api.openai.com/v1/files/{file_id}
```

Returns information about a specific file.

Path parameters

file_id string Required
The ID of the file to use for this request.

Returns

The File object matching the specified ID.

Example request curl

```
curl https://api.openai.com/v1/files/file-abc123 \
  -H "Authorization: Bearer $OPENAI_API_KEY"
```

Response

```
{
  "id": "file-abc123",
  "object": "file",
  "bytes": 120000,
  "created_at": 1677610602,
  "filename": "mydata.jsonl",
  "purpose": "fine-tune",
}
```

Delete file

```
DELETE https://api.openai.com/v1/files/{file_id}
```

Delete a file.

Path parameters

file_id string Required
The ID of the file to use for this request.

Returns

Deletion status.

Example request curl

```
curl https://api.openai.com/v1/files/file-abc123 \
  -X DELETE \
  -H "Authorization: Bearer $OPENAI_API_KEY"
```

Response

```
{
  "id": "file-abc123",
  "object": "file",
  "deleted": true
}
```

Retrieve file content

```
GET https://api.openai.com/v1/files/{file_id}/content
```

Returns the contents of the specified file.

Path parameters

file_id string Required
The ID of the file to use for this request.

Returns

The file content.

Example request curl

```
curl https://api.openai.com/v1/files/file-abc123/content \
  -H "Authorization: Bearer $OPENAI_API_KEY" > file.jsonl
```

The file object

The `File` object represents a document that has been uploaded to OpenAI.

bytes integer
The size of the file, in bytes.

created_at integer
The Unix timestamp (in seconds) for when the file was created.

expires_at integer
The Unix timestamp (in seconds) for when the file will expire.

filename string
The name of the file.

id string
The file identifier, which can be referenced in the API endpoints.

object string
The object type, which is always `file`.

purpose string
The intended purpose of the file. Supported values are `assistants`, `assistants_output`, `batch`,

`batch_output`, `fine-tune`, `fine-tune-results`, `vision`, and `user_data`.

status Deprecated string
Deprecated. The current status of the file, which can be either `uploaded`, `processed`, or `error`.

status_details Deprecated string
Deprecated. For details on why a fine-tuning training file failed validation, see the `error` field on `fine_tuning.job`.

OBJECT The file object

```
{
  "id": "file-abc123",
  "object": "file",
  "bytes": 120000,
  "created_at": 1677610602,
  "expires_at": 1680202602,
  "filename": "salesOverview.pdf",
  "purpose": "assistants",
}
```

Uploads

Allows you to upload large files in multiple parts.

Create upload

`POST https://api.openai.com/v1/uploads`

Creates an intermediate Upload object that you can add Parts to. Currently, an Upload can accept at most 8 GB in total and expires after an hour after you create it.
Once you complete the Upload, we will create a File object that contains all the parts you uploaded. This File is usable in the rest of our platform as a regular File object.
For certain `purpose` values, the correct `mime_type` must be specified. Please refer to documentation for the supported MIME types for your use case.
For guidance on the proper filename extensions for each purpose, please follow the documentation on creating a File.

Request body

bytes integer Required
The number of bytes in the file you are uploading.

filename string Required
The name of the file to upload.

mime_type string Required
The MIME type of the file.

This must fall within the supported MIME types for your file purpose. See the supported MIME types for assistants and vision.

purpose string Required
The intended purpose of the uploaded file.

See the documentation on File purposes.

Returns

The Upload object with status `pending`.

Example request curl

```curl
curl https://api.openai.com/v1/uploads \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -d '{
    "purpose": "fine-tune",
    "filename": "training_examples.jsonl",
    "bytes": 2147483648,
    "mime_type": "text/jsonl"
  }'
```

Response

```
{
  "id": "upload_abc123",
  "object": "upload",
  "bytes": 2147483648,
  "created_at": 1719184911,
  "filename": "training_examples.jsonl",
```

```
  "purpose": "fine-tune",
  "status": "pending",
  "expires_at": 1719127296
}
```

Add upload part

`POST https://api.openai.com/v1/uploads/{upload_id}/parts`

Adds a Part to an Upload object. A Part represents a chunk of bytes from the file you are trying to upload.
Each Part can be at most 64 MB, and you can add Parts until you hit the Upload maximum of 8 GB.
It is possible to add multiple Parts in parallel. You can decide the intended order of the Parts when you complete the Upload.

Path parameters

upload_id string Required
The ID of the Upload.

Request body

data file Required
The chunk of bytes for this Part.

Returns

The upload Part object.

Example request curl

```
curl https://api.openai.com/v1/uploads/upload_abc123/parts
  -F data="aHR0cHM6Ly9hcGkub3BlbmFpLmNvbS92MS91cGxvYWRz..."
```

Response

```
{
  "id": "part_def456",
  "object": "upload.part",
  "created_at": 1719185911,
  "upload_id": "upload_abc123"
}
```

Complete upload

`POST https://api.openai.com/v1/uploads/{upload_id}/complete`

Completes the Upload.
Within the returned Upload object, there is a nested File object that is ready to use in the rest of the platform.
You can specify the order of the Parts by passing in an ordered list of the Part IDs.
The number of bytes uploaded upon completion must match the number of bytes initially specified when creating the Upload object. No Parts may be added after an Upload is completed.

Path parameters

upload_id string Required
The ID of the Upload.

Request body

part_ids array Required
The ordered list of Part IDs.

md5 string Optional
The optional md5 checksum for the file contents to verify if the bytes uploaded matches what you expect.

Returns

The Upload object with status `completed` with an additional `file` property containing the created usable File object.

Example request curl

```
curl https://api.openai.com/v1/uploads/upload_abc123/complete
  -d '{
    "part_ids": ["part_def456", "part_ghi789"]
  }'
```

Response

```
{
  "id": "upload_abc123",
  "object": "upload",
  "bytes": 2147483648,
  "created_at": 1719184911,
  "filename": "training_examples.jsonl",
  "purpose": "fine-tune",
  "status": "completed",
  "expires_at": 1719127296,
  "file": {
    "id": "file-xyz321",
    "object": "file",
    "bytes": 2147483648,
    "created_at": 1719186911,
    "filename": "training_examples.jsonl",
    "purpose": "fine-tune",
  }
}
```

Cancel upload

`POST https://api.openai.com/v1/uploads/{upload_id}/cancel`

Cancels the Upload. No Parts may be added after an Upload is cancelled.

Path parameters

upload_id string Required

The ID of the Upload.

Returns

The Upload object with status `cancelled`.

Example request curl

```
curl https://api.openai.com/v1/uploads/upload_abc123/cancel
```

Response

```
{
  "id": "upload_abc123",
  "object": "upload",
  "bytes": 2147483648,
  "created_at": 1719184911,
  "filename": "training_examples.jsonl",
  "purpose": "fine-tune",
  "status": "cancelled",
  "expires_at": 1719127296
}
```

The upload object

The Upload object can accept byte chunks in the form of Parts.

`bytes` integer
The intended number of bytes to be uploaded.

`created_at` integer
The Unix timestamp (in seconds) for when the Upload was created.

`expires_at` integer
The Unix timestamp (in seconds) for when the Upload will expire.

file undefined or null
The ready File object after the Upload is completed.

filename string
The name of the file to be uploaded.

id string
The Upload unique identifier, which can be referenced in API endpoints.

object string
The object type, which is always "upload".

purpose string
The intended purpose of the file. Please refer here for acceptable values.

status string
The status of the Upload.

OBJECT The upload object

```
{
  "id": "upload_abc123",
  "object": "upload",
  "bytes": 2147483648,
  "created_at": 1719184911,
  "filename": "training_examples.jsonl",
  "purpose": "fine-tune",
  "status": "completed",
  "expires_at": 1719127296,
  "file": {
    "id": "file-xyz321",
    "object": "file",
    "bytes": 2147483648,
    "created_at": 1719186911,
    "filename": "training_examples.jsonl",
    "purpose": "fine-tune",
  }
}
```

The upload part object

The upload Part represents a chunk of bytes we can add to an Upload object.

created_at integer
The Unix timestamp (in seconds) for when the Part was created.

id string
The upload Part unique identifier, which can be referenced in API endpoints.

object string
The object type, which is always `upload.part` .

upload_id string
The ID of the Upload object that this Part was added to.

OBJECT The upload part object

```
{
    "id": "part_def456",
    "object": "upload.part",
    "created_at": 1719186911,
    "upload_id": "upload_abc123"
}
```

Models

List and describe the various models available in the API. You can refer to the Models documentation to understand what models are available and the differences between them.

List models

`GET https://api.openai.com/v1/models`

Lists the currently available models, and provides basic information about each one such as the owner and availability.

Returns

A list of model objects.

Example request curl

```
curl https://api.openai.com/v1/models \
  -H "Authorization: Bearer $OPENAI_API_KEY"
```

Response

```
{
  "object": "list",
  "data": [
    {
      "id": "model-id-0",
      "object": "model",
      "created": 1686935002,
      "owned_by": "organization-owner"
    },
    {
      "id": "model-id-1",
      "object": "model",
      "created": 1686935002,
      "owned_by": "organization-owner",
    },
    {
      "id": "model-id-2",
      "object": "model",
      "created": 1686935002,
      "owned_by": "openai"
    },
  ],
  "object": "list"
}
```

Retrieve model

`GET https://api.openai.com/v1/models/{model}`

Retrieves a model instance, providing basic information about the model such as the owner and permissioning.

Path parameters

model string Required
The ID of the model to use for this request

Returns

The model object matching the specified ID.

Example request gpt-4.1 curl

```
curl https://api.openai.com/v1/models/gpt-4.1 \
  -H "Authorization: Bearer $OPENAI_API_KEY"
```

Response

```
{
  "id": "gpt-4.1",
  "object": "model",
  "created": 1686935002,
  "owned_by": "openai"
}
```

Delete a fine-tuned model

`DELETE https://api.openai.com/v1/models/{model}`

Delete a fine-tuned model. You must have the Owner role in your organization to delete a model.

Path parameters

model string Required
The model to delete

Returns

Deletion status.

Example request curl

```curl
curl https://api.openai.com/v1/models/ft:gpt-4o-mini:acemeco:suffix:abc123 \
  -X DELETE \
  -H "Authorization: Bearer $OPENAI_API_KEY"
```

Response

```
{
  "id": "ft:gpt-4o-mini:acemeco:suffix:abc123",
  "object": "model",
  "deleted": true
}
```

The model object

Describes an OpenAI model offering that can be used with the API.

created integer
The Unix timestamp (in seconds) when the model was created.

id string
The model identifier, which can be referenced in the API endpoints.

object string
The object type, which is always "model".

owned_by string
The organization that owns the model.

OBJECT The model object

```
{
  "id": "gpt-4.1",
  "object": "model",
  "created": 1686935002,
  "owned_by": "openai"
```

```
}
```

Moderations

Given text and/or image inputs, classifies if those inputs are potentially harmful across several categories. Related guide: Moderations

Create moderation

`POST https://api.openai.com/v1/moderations`

Classifies if text and/or image inputs are potentially harmful. Learn more in the moderation guide.

Request body

`input` string or array Required
Input (or inputs) to classify. Can be a single string, an array of strings, or an array of multi-modal input objects similar to other models.

Hide possible types

string
A string of text to classify for moderation.

array
An array of strings to classify for moderation.

array
An array of multi-modal inputs to the moderation model.

Hide possible types

object
An object describing an image to classify.

Hide properties

`image_url` object Required
Contains either an image URL or a data URL for a base64 encoded image.

Hide properties

url string Required
Either a URL of the image or the base64 encoded image data.

type string Required
Always `image_url`.

object
An object describing text to classify.

Hide properties

text string Required
A string of text to classify.

type string Required
Always `text`.

model string Optional Defaults to omni-moderation-latest
The content moderation model you would like to use. Learn more in the moderation guide, and learn about available models here.

Returns

A moderation object.

Single string | Image and text

Example request curl

```
curl https://api.openai.com/v1/moderations \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -d '{
    "input": "I want to kill them."
  }'
```

Response

```
{
  "id": "modr-AB8CjOTu2jiq12hp1AQPfeqFWaORR",
  "model": "text-moderation-007",
  "results": [
    {
      "flagged": true,
      "categories": {
        "sexual": false,
        "hate": false,
        "harassment": true,
        "self-harm": false,
        "sexual/minors": false,
        "hate/threatening": false,
        "violence/graphic": false,
        "self-harm/intent": false,
        "self-harm/instructions": false,
        "harassment/threatening": true,
        "violence": true
      },
      "category_scores": {
        "sexual": 0.000011726012417057063,
        "hate": 0.22706663608551025,
        "harassment": 0.5215635299682617,
        "self-harm": 2.227119921371923e-6,
        "sexual/minors": 7.107352217872176e-8,
        "hate/threatening": 0.023547329008579254,
        "violence/graphic": 0.00003391829886822961,
        "self-harm/intent": 1.646940972932498e-6,
        "self-harm/instructions": 1.1198755256458526e-9,
        "harassment/threatening": 0.5694745779037476,
        "violence": 0.9971134662628174
      }
    }
  ]
}
```

The moderation object

Represents if a given text input is potentially harmful.

id string

The unique identifier for the moderation request.

model string
The model used to generate the moderation results.

results array
A list of moderation objects.

Hide properties

categories object
A list of the categories, and whether they are flagged or not.

Hide properties

harassment boolean
Content that expresses, incites, or promotes harassing language towards any target.

harassment/threatening boolean
Harassment content that also includes violence or serious harm towards any target.

hate boolean
Content that expresses, incites, or promotes hate based on race, gender, ethnicity, religion, nationality, sexual orientation, disability status, or caste. Hateful content aimed at non-protected groups (e.g., chess players) is harassment.

hate/threatening boolean
Hateful content that also includes violence or serious harm towards the targeted group based on race, gender, ethnicity, religion, nationality, sexual orientation, disability status, or caste.

illicit boolean or null
Content that includes instructions or advice that facilitate the planning or execution of wrongdoing, or that gives advice or instruction on how to commit illicit acts. For example, "how to shoplift" would fit this category.

illicit/violent boolean or null
Content that includes instructions or advice that facilitate the planning or execution of wrongdoing that also includes violence, or that gives advice or instruction on the procurement of any weapon.

self-harm boolean
Content that promotes, encourages, or depicts acts of self-harm, such as suicide, cutting, and eating disorders.

self-harm/instructions boolean
Content that encourages performing acts of self-harm, such as suicide, cutting, and eating disorders, or that gives instructions or advice on how to commit such acts.

self-harm/intent boolean
Content where the speaker expresses that they are engaging or intend to engage in acts of self-harm, such as suicide, cutting, and eating disorders.

sexual boolean
Content meant to arouse sexual excitement, such as the description of sexual activity, or that promotes sexual services (excluding sex education and wellness).

sexual/minors boolean
Sexual content that includes an individual who is under 18 years old.

violence boolean
Content that depicts death, violence, or physical injury.

violence/graphic boolean
Content that depicts death, violence, or physical injury in graphic detail.

category_applied_input_types object
A list of the categories along with the input type(s) that the score applies to.

Hide properties

harassment array
The applied input type(s) for the category 'harassment'.

harassment/threatening array
The applied input type(s) for the category 'harassment/threatening'.

hate array
The applied input type(s) for the category 'hate'.

hate/threatening array
The applied input type(s) for the category 'hate/threatening'.

illicit array
The applied input type(s) for the category 'illicit'.

illicit/violent array
The applied input type(s) for the category 'illicit/violent'.

self-harm array
The applied input type(s) for the category 'self-harm'.

self-harm/instructions array
The applied input type(s) for the category 'self-harm/instructions'.

self-harm/intent array
The applied input type(s) for the category 'self-harm/intent'.

sexual array
The applied input type(s) for the category 'sexual'.

sexual/minors array
The applied input type(s) for the category 'sexual/minors'.

violence array
The applied input type(s) for the category 'violence'.

violence/graphic array
The applied input type(s) for the category 'violence/graphic'.

category_scores object
A list of the categories along with their scores as predicted by model.

Hide properties

harassment number
The score for the category 'harassment'.

harassment/threatening number
The score for the category 'harassment/threatening'.

hate number
The score for the category 'hate'.

hate/threatening number
The score for the category 'hate/threatening'.

illicit number
The score for the category 'illicit'.

illicit/violent number
The score for the category 'illicit/violent'.

self-harm number
The score for the category 'self-harm'.

self-harm/instructions number
The score for the category 'self-harm/instructions'.

self-harm/intent number
The score for the category 'self-harm/intent'.

sexual number
The score for the category 'sexual'.

sexual/minors number
The score for the category 'sexual/minors'.

violence number
The score for the category 'violence'.

violence/graphic number
The score for the category 'violence/graphic'.

flagged boolean
Whether any of the below categories are flagged.

OBJECT The moderation object

```
{
  "id": "modr-0d9740456c391e43c445bf0f010940c7",
  "model": "omni-moderation-latest",
  "results": [
    {
      "flagged": true,
      "categories": {
        "harassment": true,
        "harassment/threatening": true,
        "sexual": false,
```

```
        "hate": false,
        "hate/threatening": false,
        "illicit": false,
        "illicit/violent": false,
        "self-harm/intent": false,
        "self-harm/instructions": false,
        "self-harm": false,
        "sexual/minors": false,
        "violence": true,
        "violence/graphic": true
      },
      "category_scores": {
        "harassment": 0.8189693396524255,
        "harassment/threatening": 0.804985420696006,
        "sexual": 1.573112165348997e-6,
        "hate": 0.007562942636942845,
        "hate/threatening": 0.004208854591835476,
        "illicit": 0.030535955153511665,
        "illicit/violent": 0.008925306722380033,
        "self-harm/intent": 0.00023023930975076432,
        "self-harm/instructions": 0.0002293869201073356,
        "self-harm": 0.012598046106750154,
        "sexual/minors": 2.212566909570261e-8,
        "violence": 0.9999992735124786,
        "violence/graphic": 0.843064871157054
      },
      "category_applied_input_types": {
        "harassment": [
          "text"
        ],
        "harassment/threatening": [
          "text"
        ],
        "sexual": [
          "text",
          "image"
        ],
        "hate": [
          "text"
        ],
        "hate/threatening": [
          "text"
        ],
        "illicit": [
          "text"
        ],
        "illicit/violent": [
```

```
          "text"
        ],
        "self-harm/intent": [
          "text",
          "image"
        ],
        "self-harm/instructions": [
          "text",
          "image"
        ],
        "self-harm": [
          "text",
          "image"
        ],
        "sexual/minors": [
          "text"
        ],
        "violence": [
          "text",
          "image"
        ],
        "violence/graphic": [
          "text",
          "image"
        ]
      }
    }
  ]
}
```

Vector stores

Vector stores power semantic search for the Retrieval API and the `file_search` tool in the Responses and Assistants APIs.

Related guide: File Search

Create vector store

```
POST https://api.openai.com/v1/vector_stores
```

Create a vector store.

Request body

chunking_strategy object Optional
The chunking strategy used to chunk the file(s). If not set, will use the `auto` strategy. Only applicable if `file_ids` is non-empty.

Hide possible types

Auto Chunking Strategy object
The default strategy. This strategy currently uses a `max_chunk_size_tokens` of `800` and `chunk_overlap_tokens` of `400`.

Hide properties

type string Required
Always `auto`.

Static Chunking Strategy object
Customize your own chunking strategy by setting chunk size and chunk overlap.

Hide properties

static object Required

Hide properties

chunk_overlap_tokens integer Required
The number of tokens that overlap between chunks. The default value is `400`.

Note that the overlap must not exceed half of `max_chunk_size_tokens`.

max_chunk_size_tokens integer Required
The maximum number of tokens in each chunk. The default value is `800`. The minimum value is `100` and the maximum value is `4096`.

type string Required
Always `static`.

expires_after object Optional
The expiration policy for a vector store.

Hide properties

anchor string Required
Anchor timestamp after which the expiration policy applies. Supported anchors: `last_active_at`.

days integer Required
The number of days after the anchor time that the vector store will expire.

file_ids array Optional
A list of File IDs that the vector store should use. Useful for tools like `file_search` that can access files.

metadata map Optional
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

name string Optional
The name of the vector store.

Returns

A vector store object.

Example request curl

```
curl https://api.openai.com/v1/vector_stores \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -H "OpenAI-Beta: assistants=v2" \
  -d '{
    "name": "Support FAQ"
  }'
```

Response

```
{
  "id": "vs_abc123",
  "object": "vector_store",
  "created_at": 1699061776,
  "name": "Support FAQ",
```

```
  "bytes": 139920,
  "file_counts": {
    "in_progress": 0,
    "completed": 3,
    "failed": 0,
    "cancelled": 0,
    "total": 3
  }
}
```

List vector stores

`GET https://api.openai.com/v1/vector_stores`

Returns a list of vector stores.

Query parameters

after string Optional
A cursor for use in pagination. `after` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, ending with obj_foo, your subsequent call can include after=obj_foo in order to fetch the next page of the list.

before string Optional
A cursor for use in pagination. `before` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, starting with obj_foo, your subsequent call can include before=obj_foo in order to fetch the previous page of the list.

limit integer Optional Defaults to 20
A limit on the number of objects to be returned. Limit can range between 1 and 100, and the default is 20.

order string Optional Defaults to desc
Sort order by the `created_at` timestamp of the objects. `asc` for ascending order and `desc` for descending order.

Returns

A list of vector store objects.

Example request curl

```curl
curl https://api.openai.com/v1/vector_stores \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -H "OpenAI-Beta: assistants=v2"
```

Response

```json
{
  "object": "list",
  "data": [
    {
      "id": "vs_abc123",
      "object": "vector_store",
      "created_at": 1699061776,
      "name": "Support FAQ",
      "bytes": 139920,
      "file_counts": {
        "in_progress": 0,
        "completed": 3,
        "failed": 0,
        "cancelled": 0,
        "total": 3
      }
    },
    {
      "id": "vs_abc456",
      "object": "vector_store",
      "created_at": 1699061776,
      "name": "Support FAQ v2",
      "bytes": 139920,
      "file_counts": {
        "in_progress": 0,
        "completed": 3,
        "failed": 0,
        "cancelled": 0,
        "total": 3
      }
    }
  ],
  "first_id": "vs_abc123",

```

```
  "last_id": "vs_abc456",
  "has_more": false
}
```

Retrieve vector store

```
GET https://api.openai.com/v1/vector_stores/{vector_store_id}
```

Retrieves a vector store.

Path parameters

vector_store_id string Required
The ID of the vector store to retrieve.

Returns

The vector store object matching the specified ID.

Example request curl

```
curl https://api.openai.com/v1/vector_stores/vs_abc123 \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -H "OpenAI-Beta: assistants=v2"
```

Response

```
{
  "id": "vs_abc123",
  "object": "vector_store",
  "created_at": 1699061776
}
```

Modify vector store

```
POST https://api.openai.com/v1/vector_stores/{vector_store_id}
```

Modifies a vector store.

Path parameters

vector_store_id string Required
The ID of the vector store to modify.

Request body

expires_after object or null Optional
The expiration policy for a vector store.

Hide properties

anchor string Required
Anchor timestamp after which the expiration policy applies. Supported anchors: `last_active_at`.

days integer Required
The number of days after the anchor time that the vector store will expire.

metadata map Optional
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

name string or null Optional
The name of the vector store.

Returns

The modified vector store object.

Example request curl

```
curl https://api.openai.com/v1/vector_stores/vs_abc123 \
```

```
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -H "OpenAI-Beta: assistants=v2"
  -d '{
    "name": "Support FAQ"
  }'
```

Response

```
{
  "id": "vs_abc123",
  "object": "vector_store",
  "created_at": 1699061776,
  "name": "Support FAQ",
  "bytes": 139920,
  "file_counts": {
    "in_progress": 0,
    "completed": 3,
    "failed": 0,
    "cancelled": 0,
    "total": 3
  }
}
```

Delete vector store

```
DELETE https://api.openai.com/v1/vector_stores/{vector_store_id}
```

Delete a vector store.

Path parameters

vector_store_id string Required
The ID of the vector store to delete.

Returns

Deletion status

Example request curl

```
curl https://api.openai.com/v1/vector_stores/vs_abc123 \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -H "OpenAI-Beta: assistants=v2" \
  -X DELETE
```

Response

```
{
  id: "vs_abc123",
  object: "vector_store.deleted",
  deleted: true
}
```

Search vector store

`POST https://api.openai.com/v1/vector_stores/{vector_store_id}/search`

Search a vector store for relevant chunks based on a query and file attributes filter.

Path parameters

vector_store_id string Required
The ID of the vector store to search.

Request body

query string or array Required
A query string for a search

filters object Optional
A filter to apply based on file attributes.

Hide possible types

Comparison Filter object
A filter used to compare a specified attribute key to a given value using a defined comparison operation.

Hide properties

key string Required
The key to compare against the value.

type string Required
Specifies the comparison operator: `eq`, `ne`, `gt`, `gte`, `lt`, `lte`.

- `eq`: equals
- `ne`: not equal
- `gt`: greater than
- `gte`: greater than or equal
- `lt`: less than
- `lte`: less than or equal

value string / number / boolean Required
The value to compare against the attribute key; supports string, number, or boolean types.

Compound Filter object
Combine multiple filters using `and` or `or`.

Hide properties

filters array Required
Array of filters to combine. Items can be `ComparisonFilter` or `CompoundFilter`.

Hide possible types

Comparison Filter object
A filter used to compare a specified attribute key to a given value using a defined comparison operation.

Hide properties

key string Required
The key to compare against the value.

type string Required
Specifies the comparison operator: `eq`, `ne`, `gt`, `gte`, `lt`, `lte`.

- `eq`: equals
- `ne`: not equal
- `gt`: greater than

`gte` : greater than or equal

`lt` : less than

`lte` : less than or equal

value string / number / boolean Required

The value to compare against the attribute key; supports string, number, or boolean types.

type string Required

Type of operation: `and` or `or` .

max_num_results integer Optional Defaults to 10

The maximum number of results to return. This number should be between 1 and 50 inclusive.

ranking_options object Optional

Ranking options for search.

Hide properties

ranker string Optional Defaults to auto

score_threshold number Optional Defaults to 0

rewrite_query boolean Optional Defaults to false

Whether to rewrite the natural language query for vector search.

Returns

A page of search results from the vector store.

Example request curl

```
curl -X POST \
https://api.openai.com/v1/vector_stores/vs_abc123/search \
-H "Authorization: Bearer $OPENAI_API_KEY" \
-H "Content-Type: application/json" \
-d '{"query": "What is the return policy?", "filters": {...}}'
```

Response

```
{
  "object": "vector_store.search_results.page",
  "search_query": "What is the return policy?",
  "data": [
    {
      "file_id": "file_123",
      "filename": "document.pdf",
      "score": 0.95,
      "attributes": {
        "author": "John Doe",
        "date": "2023-01-01"
      },
      "content": [
        {
          "type": "text",
          "text": "Relevant chunk"
        }
      ]
    },
    {
      "file_id": "file_456",
      "filename": "notes.txt",
      "score": 0.89,
      "attributes": {
        "author": "Jane Smith",
        "date": "2023-01-02"
      },
      "content": [
        {
          "type": "text",
          "text": "Sample text content from the vector store."
        }
      ]
    }
  ],
  "has_more": false,
  "next_page": null
}
```

The vector store object

A vector store is a collection of processed files can be used by the `file_search` tool.

created_at integer
The Unix timestamp (in seconds) for when the vector store was created.

expires_after object
The expiration policy for a vector store.

Hide properties

anchor string
Anchor timestamp after which the expiration policy applies. Supported anchors: `last_active_at` .

days integer
The number of days after the anchor time that the vector store will expire.

expires_at integer or null
The Unix timestamp (in seconds) for when the vector store will expire.

file_counts object

Hide properties

cancelled integer
The number of files that were cancelled.

completed integer
The number of files that have been successfully processed.

failed integer
The number of files that have failed to process.

in_progress integer
The number of files that are currently being processed.

total integer
The total number of files.

id string
The identifier, which can be referenced in API endpoints.

last_active_at integer or null

The Unix timestamp (in seconds) for when the vector store was last active.

metadata map

Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

name string

The name of the vector store.

object string

The object type, which is always `vector_store`.

status string

The status of the vector store, which can be either `expired`, `in_progress`, or `completed`. A status of `completed` indicates that the vector store is ready for use.

usage_bytes integer

The total number of bytes used by the files in the vector store.

OBJECT The vector store object

```
{
  "id": "vs_123",
  "object": "vector_store",
  "created_at": 1698107661,
  "usage_bytes": 123456,
  "last_active_at": 1698107661,
  "name": "my_vector_store",
  "status": "completed",
  "file_counts": {
    "in_progress": 0,
    "completed": 100,
    "cancelled": 0,
    "failed": 0,
    "total": 100
  },
  "last_used_at": 1698107661
}
```

Vector store files

Vector store files represent files inside a vector store.

Related guide: File Search

Create vector store file

```
POST https://api.openai.com/v1/vector_stores/{vector_store_id}/files
```

Create a vector store file by attaching a File to a vector store.

Path parameters

`vector_store_id` string Required
The ID of the vector store for which to create a File.

Request body

`file_id` string Required
A File ID that the vector store should use. Useful for tools like `file_search` that can access files.

`attributes` map Optional
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard. Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters, booleans, or numbers.

`chunking_strategy` object Optional
The chunking strategy used to chunk the file(s). If not set, will use the `auto` strategy.

^ Hide possible types

Auto Chunking Strategy object
The default strategy. This strategy currently uses a `max_chunk_size_tokens` of `800` and `chunk_overlap_tokens` of `400`.

^ Hide properties

`type` string Required

Always `auto`.

Static Chunking Strategy object

Customize your own chunking strategy by setting chunk size and chunk overlap.

Hide properties

static object Required

Hide properties

chunk_overlap_tokens integer Required

The number of tokens that overlap between chunks. The default value is `400`.

Note that the overlap must not exceed half of `max_chunk_size_tokens`.

max_chunk_size_tokens integer Required

The maximum number of tokens in each chunk. The default value is `800`. The minimum value is `100` and the maximum value is `4096`.

type string Required

Always `static`.

Returns

A vector store file object.

Example request curl

```
curl https://api.openai.com/v1/vector_stores/vs_abc123/files \
    -H "Authorization: Bearer $OPENAI_API_KEY" \
    -H "Content-Type: application/json" \
    -H "OpenAI-Beta: assistants=v2" \
    -d '{
      "file_id": "file-abc123"
    }'
```

Response

```
{
  "id": "file-abc123",
  "object": "vector_store.file",
```

```
  "created_at": 1699061776,
  "usage_bytes": 1234,
  "vector_store_id": "vs_abcd",
  "status": "completed",
  "last_error": null
}
```

List vector store files

```
GET https://api.openai.com/v1/vector_stores/{vector_store_id}/files
```

Returns a list of vector store files.

Path parameters

vector_store_id string Required
The ID of the vector store that the files belong to.

Query parameters

after string Optional
A cursor for use in pagination. `after` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, ending with obj_foo, your subsequent call can include after=obj_foo in order to fetch the next page of the list.

before string Optional
A cursor for use in pagination. `before` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, starting with obj_foo, your subsequent call can include before=obj_foo in order to fetch the previous page of the list.

filter string Optional
Filter by file status. One of `in_progress`, `completed`, `failed`, `cancelled`.

limit integer Optional Defaults to 20
A limit on the number of objects to be returned. Limit can range between 1 and 100, and the default is 20.

order string Optional Defaults to desc
Sort order by the `created_at` timestamp of the objects. `asc` for ascending order and `desc` for

descending order.

Returns

A list of vector store file objects.

Example request curl

```
curl https://api.openai.com/v1/vector_stores/vs_abc123/files \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -H "OpenAI-Beta: assistants=v2"
```

Response

```
{
  "object": "list",
  "data": [
    {
      "id": "file-abc123",
      "object": "vector_store.file",
      "created_at": 1699061776,
      "vector_store_id": "vs_abc123"
    },
    {
      "id": "file-abc456",
      "object": "vector_store.file",
      "created_at": 1699061776,
      "vector_store_id": "vs_abc123"
    }
  ],
  "first_id": "file-abc123",
  "last_id": "file-abc456",
  "has_more": false
}
```

Retrieve vector store file

```
GET https://api.openai.com/v1/vector_stores/{vector_store_id}/files/{file_id}
```

Retrieves a vector store file.

Path parameters

`file_id` string Required
The ID of the file being retrieved.

`vector_store_id` string Required
The ID of the vector store that the file belongs to.

Returns

The vector store file object.

Example request curl

```
curl https://api.openai.com/v1/vector_stores/vs_abc123/files/file-abc123 \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -H "OpenAI-Beta: assistants=v2"
```

Response

```
{
  "id": "file-abc123",
  "object": "vector_store.file",
  "created_at": 1699061776,
  "vector_store_id": "vs_abcd",
  "status": "completed",
  "last_error": null
}
```

Retrieve vector store file content

```
GET https://api.openai.com/v1/vector_stores/{vector_store_id}/files/{file_id}/content
```

Retrieve the parsed contents of a vector store file.

Path parameters

file_id string Required
The ID of the file within the vector store.

vector_store_id string Required
The ID of the vector store.

Returns

The parsed contents of the specified vector store file.

Example request curl

```curl
curl \
https://api.openai.com/v1/vector_stores/vs_abc123/files/file-abc123/content \
-H "Authorization: Bearer $OPENAI_API_KEY"
```

Response

```response
{
  "file_id": "file-abc123",
  "filename": "example.txt",
  "attributes": {"key": "value"},
  "content": [
    {"type": "text", "text": "..."},
    ...
  ]
}
```

Update vector store file attributes

```
POST https://api.openai.com/v1/vector_stores/{vector_store_id}/files/{file_id}
```

Update attributes on a vector store file.

Path parameters

file_id string Required
The ID of the file to update attributes.

vector_store_id string Required
The ID of the vector store the file belongs to.

Request body

attributes map Required
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard. Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters, booleans, or numbers.

Returns

The updated vector store file object.

Example request curl

```curl
curl https://api.openai.com/v1/vector_stores/{vector_store_id}/files/{file_id} \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{"attributes": {"key1": "value1", "key2": 2}}'
```

Response

```
{
  "id": "file-abc123",
  "object": "vector_store.file",
  "usage_bytes": 1234,
  "created_at": 1699061776,
  "vector_store_id": "vs_abcd",
  "status": "completed",
  "last_error": null,
  "chunking_strategy": {...},
  "attributes": {"key1": "value1", "key2": 2}
}
```

Delete vector store file

```
DELETE https://api.openai.com/v1/vector_stores/{vector_store_id}/files/{file_id}
```

Delete a vector store file. This will remove the file from the vector store but the file itself will not be deleted. To delete the file, use the delete file endpoint.

Path parameters

file_id string Required
The ID of the file to delete.

vector_store_id string Required
The ID of the vector store that the file belongs to.

Returns

Deletion status

Example request curl

```curl
curl https://api.openai.com/v1/vector_stores/vs_abc123/files/file-abc123 \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -H "OpenAI-Beta: assistants=v2" \
  -X DELETE
```

Response

```
{
  id: "file-abc123",
  object: "vector_store.file.deleted",
  deleted: true
}
```

The vector store file object Beta

A list of files attached to a vector store.

attributes map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard. Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters, booleans, or numbers.

chunking_strategy object
The strategy used to chunk the file.

Hide possible types

Static Chunking Strategy object

Hide properties

static object

Hide properties

chunk_overlap_tokens integer
The number of tokens that overlap between chunks. The default value is `400`.

Note that the overlap must not exceed half of `max_chunk_size_tokens`.

max_chunk_size_tokens integer
The maximum number of tokens in each chunk. The default value is `800`. The minimum value is `100` and the maximum value is `4096`.

type string
Always `static`.

Other Chunking Strategy object
This is returned when the chunking strategy is unknown. Typically, this is because the file was indexed before the `chunking_strategy` concept was introduced in the API.

Hide properties

type string
Always `other`.

created_at integer
The Unix timestamp (in seconds) for when the vector store file was created.

id string
The identifier, which can be referenced in API endpoints.

last_error object or null
The last error associated with this vector store file. Will be `null` if there are no errors.

Hide properties

code string
One of `server_error` or `rate_limit_exceeded`.

message string
A human-readable description of the error.

object string
The object type, which is always `vector_store.file`.

status string
The status of the vector store file, which can be either `in_progress`, `completed`, `cancelled`, or `failed`. The status `completed` indicates that the vector store file is ready for use.

usage_bytes integer
The total vector store usage in bytes. Note that this may be different from the original file size.

vector_store_id string
The ID of the vector store that the File is attached to.

OBJECT The vector store file object

```
{
  "id": "file-abc123",
  "object": "vector_store.file",
  "usage_bytes": 1234,
  "created_at": 1698107661,
  "vector_store_id": "vs_abc123",
  "status": "completed",
  "last_error": null,
  "chunking_strategy": {
    "type": "static",
    "static": {
      "max_chunk_size_tokens": 800,
      "chunk_overlap_tokens": 400
    }
```

```
  }
}
```

Vector store file batches

Vector store file batches represent operations to add multiple files to a vector store. Related guide: File Search

Create vector store file batch

`POST https://api.openai.com/v1/vector_stores/{vector_store_id}/file_batches`

Create a vector store file batch.

Path parameters

`vector_store_id` string Required
The ID of the vector store for which to create a File Batch.

Request body

`file_ids` array Required
A list of File IDs that the vector store should use. Useful for tools like `file_search` that can access files.

`attributes` map Optional
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard. Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters, booleans, or numbers.

`chunking_strategy` object Optional
The chunking strategy used to chunk the file(s). If not set, will use the `auto` strategy.

Hide possible types

Auto Chunking Strategy object
The default strategy. This strategy currently uses a `max_chunk_size_tokens` of `800` and

`chunk_overlap_tokens` of `400`.

Hide properties

type string Required
Always `auto`.

Static Chunking Strategy object
Customize your own chunking strategy by setting chunk size and chunk overlap.

Hide properties

static object Required

Hide properties

chunk_overlap_tokens integer Required
The number of tokens that overlap between chunks. The default value is `400`.

Note that the overlap must not exceed half of `max_chunk_size_tokens`.

max_chunk_size_tokens integer Required
The maximum number of tokens in each chunk. The default value is `800`. The minimum value is `100` and the maximum value is `4096`.

type string Required
Always `static`.

Returns

A vector store file batch object.

Example request curl

```
curl https://api.openai.com/v1/vector_stores/vs_abc123/file_batches \
    -H "Authorization: Bearer $OPENAI_API_KEY" \
    -H "Content-Type: application/json \
    -H "OpenAI-Beta: assistants=v2" \
    -d '{
      "file_ids": ["file-abc123", "file-abc456"]
    }'
```

Response

```
{
  "id": "vsfb_abc123",
  "object": "vector_store.file_batch",
  "created_at": 1699061776,
  "vector_store_id": "vs_abc123",
  "status": "in_progress",
  "file_counts": {
    "in_progress": 1,
    "completed": 1,
    "failed": 0,
    "cancelled": 0,
    "total": 0,
  }
}
```

Retrieve vector store file batch

`GET https://api.openai.com/v1/vector_stores/{vector_store_id}/file_batches/{batch_id}`

Retrieves a vector store file batch.

Path parameters

batch_id string Required
The ID of the file batch being retrieved.

vector_store_id string Required
The ID of the vector store that the file batch belongs to.

Returns

The vector store file batch object.

Example request curl

```
curl https://api.openai.com/v1/vector_stores/vs_abc123/files_batches/vsfb_abc123 \
```

```
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -H "OpenAI-Beta: assistants=v2"
```

Response

```
{
  "id": "vsfb_abc123",
  "object": "vector_store.file_batch",
  "created_at": 1699061776,
  "vector_store_id": "vs_abc123",
  "status": "in_progress",
  "file_counts": {
    "in_progress": 1,
    "completed": 1,
    "failed": 0,
    "cancelled": 0,
    "total": 0,
  }
}
```

Cancel vector store file batch

```
POST https://api.openai.com/v1/vector_stores/{vector_store_id}/file_batches/{batch_id}/cancel
```

Cancel a vector store file batch. This attempts to cancel the processing of files in this batch as soon as possible.

Path parameters

batch_id string Required
The ID of the file batch to cancel.

vector_store_id string Required
The ID of the vector store that the file batch belongs to.

Returns

The modified vector store file batch object.

Example request curl

```curl
curl https://api.openai.com/v1/vector_stores/vs_abc123/files_batches/vsfb_abc123/cancel \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -H "OpenAI-Beta: assistants=v2" \
  -X POST
```

Response

```{
  "id": "vsfb_abc123",
  "object": "vector_store.file_batch",
  "created_at": 1699061776,
  "vector_store_id": "vs_abc123",
  "status": "in_progress",
  "file_counts": {
    "in_progress": 12,
    "completed": 3,
    "failed": 0,
    "cancelled": 0,
    "total": 15,
  }
}
```

List vector store files in a batch

```
GET https://api.openai.com/v1/vector_stores/{vector_store_id}/file_batches/{batch_id}/files
```

Returns a list of vector store files in a batch.

Path parameters

batch_id string Required
The ID of the file batch that the files belong to.

vector_store_id string Required

The ID of the vector store that the files belong to.

Query parameters

after string Optional
A cursor for use in pagination. `after` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, ending with obj_foo, your subsequent call can include after=obj_foo in order to fetch the next page of the list.

before string Optional
A cursor for use in pagination. `before` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, starting with obj_foo, your subsequent call can include before=obj_foo in order to fetch the previous page of the list.

filter string Optional
Filter by file status. One of `in_progress`, `completed`, `failed`, `cancelled`.

limit integer Optional Defaults to 20
A limit on the number of objects to be returned. Limit can range between 1 and 100, and the default is 20.

order string Optional Defaults to desc
Sort order by the `created_at` timestamp of the objects. `asc` for ascending order and `desc` for descending order.

Returns

A list of vector store file objects.

Example request curl

```curl
curl https://api.openai.com/v1/vector_stores/vs_abc123/files_batches/vsfb_abc123/files \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -H "OpenAI-Beta: assistants=v2"
```

Response

```
{
```

```
  "object": "list",
  "data": [
    {
      "id": "file-abc123",
      "object": "vector_store.file",
      "created_at": 1699061776,
      "vector_store_id": "vs_abc123"
    },
    {
      "id": "file-abc456",
      "object": "vector_store.file",
      "created_at": 1699061776,
      "vector_store_id": "vs_abc123"
    }
  ],
  "first_id": "file-abc123",
  "last_id": "file-abc456",
  "has_more": false
}
```

The vector store files batch object Beta

A batch of files attached to a vector store.

created_at integer
The Unix timestamp (in seconds) for when the vector store files batch was created.

file_counts object

Hide properties

cancelled integer
The number of files that where cancelled.

completed integer
The number of files that have been processed.

failed integer
The number of files that have failed to process.

in_progress integer
The number of files that are currently being processed.

total integer
The total number of files.

id string
The identifier, which can be referenced in API endpoints.

object string
The object type, which is always `vector_store.file_batch`.

status string
The status of the vector store files batch, which can be either `in_progress`, `completed`, `cancelled` or `failed`.

vector_store_id string
The ID of the vector store that the File is attached to.

OBJECT The vector store files batch object

```
{
  "id": "vsfb_123",
  "object": "vector_store.files_batch",
  "created_at": 1698107661,
  "vector_store_id": "vs_abc123",
  "status": "completed",
  "file_counts": {
    "in_progress": 0,
    "completed": 100,
    "failed": 0,
    "cancelled": 0,
    "total": 100
  }
}
```

Containers

Create and manage containers for use with the Code Interpreter tool.

Create container

```
POST https://api.openai.com/v1/containers
```

Create Container

Request body

name string Required
Name of the container to create.

expires_after object Optional
Container expiration time in seconds relative to the 'anchor' time.

Hide properties

anchor string Required
Time anchor for the expiration time. Currently only 'last_active_at' is supported.

minutes integer Required

file_ids array Optional
IDs of files to copy to the container.

Returns

The created container object.

Example request curl

```
curl https://api.openai.com/v1/containers \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
        "name": "My Container"
      }'
```

Response

```
{
    "id": "cntr_682e30645a488191b6363a0cbefc0f0a025ec61b66250591",
    "object": "container",
```

```
    "created_at": 1747857508,
    "status": "running",
    "expires_after": {
        "anchor": "last_active_at",
        "minutes": 20
    },
    "last_active_at": 1747857508,
    "name": "My Container"
}
```

List containers

`GET https://api.openai.com/v1/containers`

List Containers

Query parameters

`after` string Optional
A cursor for use in pagination. `after` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, ending with obj_foo, your subsequent call can include after=obj_foo in order to fetch the next page of the list.

`limit` integer Optional Defaults to 20
A limit on the number of objects to be returned. Limit can range between 1 and 100, and the default is 20.

`order` string Optional Defaults to desc
Sort order by the `created_at` timestamp of the objects. `asc` for ascending order and `desc` for descending order.

Returns

a list of container objects.

Example request curl

```
curl https://api.openai.com/v1/containers \
  -H "Authorization: Bearer $OPENAI_API_KEY"
```

Response

```
{
  "object": "list",
  "data": [
    {
        "id": "cntr_682dfebaacac8198bbfe9c2474fb6f4a085685cbe3cb5863",
        "object": "container",
        "created_at": 1747844794,
        "status": "running",
        "expires_after": {
            "anchor": "last_active_at",
            "minutes": 20
        },
        "last_active_at": 1747844794,
        "name": "My Container"
    }
  ],
  "first_id": "container_123",
  "last_id": "container_123",
  "has_more": false
}
```

Retrieve container

```
GET https://api.openai.com/v1/containers/{container_id}
```

Retrieve Container

Path parameters

container_id string Required

Returns

The container object.

Example request curl

```curl
curl https://api.openai.com/v1/containers/cntr_682dfebaacac8198bbfe9c2474fb6f4a085685cbe3cb
  -H "Authorization: Bearer $OPENAI_API_KEY"
```

Response

```
{
    "id": "cntr_682dfebaacac8198bbfe9c2474fb6f4a085685cbe3cb5863",
    "object": "container",
    "created_at": 1747844794,
    "status": "running",
    "expires_after": {
        "anchor": "last_active_at",
        "minutes": 20
    },
    "last_active_at": 1747844794,
    "name": "My Container"
}
```

Delete a container

`DELETE https://api.openai.com/v1/containers/{container_id}`

Delete Container

Path parameters

container_id string Required
The ID of the container to delete.

Returns

Deletion Status

Example request curl

```
curl -X DELETE https://api.openai.com/v1/containers/cntr_682dfebaacac8198bbfe9c2474fb6f4a08
  -H "Authorization: Bearer $OPENAI_API_KEY"
```

Response

```
{
    "id": "cntr_682dfebaacac8198bbfe9c2474fb6f4a085685cbe3cb5863",
    "object": "container.deleted",
    "deleted": true
}
```

The container object

created_at integer
Unix timestamp (in seconds) when the container was created.

expires_after object
The container will expire after this time period. The anchor is the reference point for the expiration. The minutes is the number of minutes after the anchor before the container expires.

Hide properties

anchor string
The reference point for the expiration.

minutes integer
The number of minutes after the anchor before the container expires.

id string
Unique identifier for the container.

name string
Name of the container.

object string
The type of this object.

status string
Status of the container (e.g., active, deleted).

OBJECT The container object

```
{
   "id": "cntr_682dfebaacac8198bbfe9c2474fb6f4a085685cbe3cb5863",
   "object": "container",
   "created_at": 1747844794,
   "status": "running",
   "expires_after": {
     "anchor": "last_active_at",
     "minutes": 20
   },
   "last_active_at": 1747844794,
   "name": "My Container"
}
```

Container Files

Create and manage container files for use with the Code Interpreter tool.

Create container file

`POST https://api.openai.com/v1/containers/{container_id}/files`

Create a Container File
You can send either a multipart/form-data request with the raw file content, or a JSON request with a file ID.

Path parameters

container_id string Required

Request body

file file Optional
The File object (not file name) to be uploaded.

file_id string Optional
Name of the file to create.

Returns

The created container file object.

Example request curl

```
curl https://api.openai.com/v1/containers/cntr_682e0e7318108198aa783fd921ff305e08e78805b9fc
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -F file="@example.txt"
```

Response

```
{
  "id": "cfile_682e0e8a43c88191a7978f477a09bdf5",
  "object": "container.file",
  "created_at": 1747848842,
  "bytes": 880,
  "container_id": "cntr_682e0e7318108198aa783fd921ff305e08e78805b9fdbb04",
  "path": "/mnt/data/88e12fa445d32636f190a0b33daed6cb-tsconfig.json",
  "source": "user"
}
```

List container files

`GET https://api.openai.com/v1/containers/{container_id}/files`

List Container files

Path parameters

container_id string Required

Query parameters

after string Optional

A cursor for use in pagination. `after` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, ending with obj_foo, your subsequent call can include after=obj_foo in order to fetch the next page of the list.

limit integer Optional Defaults to 20

A limit on the number of objects to be returned. Limit can range between 1 and 100, and the default is 20.

order string Optional Defaults to desc

Sort order by the `created_at` timestamp of the objects. `asc` for ascending order and `desc` for descending order.

Returns

a list of container file objects.

Example request curl

```
curl https://api.openai.com/v1/containers/cntr_682e0e7318108198aa783fd921ff305e08e78805b9fc
  -H "Authorization: Bearer $OPENAI_API_KEY"
```

Response

```
{
    "object": "list",
    "data": [
        {
            "id": "cfile_682e0e8a43c88191a7978f477a09bdf5",
            "object": "container.file",
            "created_at": 1747848842,
            "bytes": 880,
            "container_id": "cntr_682e0e7318108198aa783fd921ff305e08e78805b9fdbb04",
            "path": "/mnt/data/88e12fa445d32636f190a0b33daed6cb-tsconfig.json",
            "source": "user"
        }
    ],
    "first_id": "cfile_682e0e8a43c88191a7978f477a09bdf5",
    "has_more": false,
    "last_id": "cfile_682e0e8a43c88191a7978f477a09bdf5"
```

```
}
```

Retrieve container file

`GET https://api.openai.com/v1/containers/{container_id}/files/{file_id}`

Retrieve Container File

Path parameters

container_id string Required

file_id string Required

Returns

The container file object.

Example request curl

```
curl https://api.openai.com/v1/containers/container_123/files/file_456 \
  -H "Authorization: Bearer $OPENAI_API_KEY"
```

Response

```
{
    "id": "cfile_682e0e8a43c88191a7978f477a09bdf5",
    "object": "container.file",
    "created_at": 1747848842,
    "bytes": 880,
    "container_id": "cntr_682e0e7318108198aa783fd921ff305e08e78805b9fdbb04",
    "path": "/mnt/data/88e12fa445d32636f190a0b33daed6cb-tsconfig.json",
    "source": "user"
}
```

Retrieve container file content

```
GET https://api.openai.com/v1/containers/{container_id}/files/{file_id}/content
```

Retrieve Container File Content

Path parameters

container_id string Required

file_id string Required

Returns

The contents of the container file.

Example request curl

```curl
curl https://api.openai.com/v1/containers/container_123/files/cfile_456/content \
  -H "Authorization: Bearer $OPENAI_API_KEY"
```

Response

```
<binary content of the file>
```

Delete a container file

```
DELETE https://api.openai.com/v1/containers/{container_id}/files/{file_id}
```

Delete Container File

Path parameters

container_id string Required

file_id string Required

Returns

Deletion Status

Example request curl

```curl
curl -X DELETE https://api.openai.com/v1/containers/cntr_682dfebaacac8198bbfe9c2474fb6f4a08
  -H "Authorization: Bearer $OPENAI_API_KEY"
```

Response

```
{
    "id": "cfile_682e0e8a43c88191a7978f477a09bdf5",
    "object": "container.file.deleted",
    "deleted": true
}
```

The container file object

bytes integer
Size of the file in bytes.

container_id string
The container this file belongs to.

created_at integer
Unix timestamp (in seconds) when the file was created.

id string
Unique identifier for the file.

object string
The type of this object (`container.file`).

path string
Path of the file in the container.

source string
Source of the file (e.g., `user`, `assistant`).

OBJECT The container file object

```
{
    "id": "cfile_682e0e8a43c88191a7978f477a09bdf5",
    "object": "container.file",
    "created_at": 1747848842,
    "bytes": 880,
    "container_id": "cntr_682e0e7318108198aa783fd921ff305e08e78805b9fdbb04",
    "path": "/mnt/data/88e12fa445d32636f190a0b33daed6cb-tsconfig.json",
    "source": "user"
}
```

Assistants Beta

Build assistants that can call models and use tools to perform tasks.

Get started with the Assistants API

Create assistant Beta

`POST https://api.openai.com/v1/assistants`

Create an assistant with a model and instructions.

Request body

model string Required
ID of the model to use. You can use the List models API to see all of your available models, or see our Model overview for descriptions of them.

description string or null Optional

The description of the assistant. The maximum length is 512 characters.

`instructions` string or null Optional
The system instructions that the assistant uses. The maximum length is 256,000 characters.

`metadata` map Optional
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

`name` string or null Optional
The name of the assistant. The maximum length is 256 characters.

`reasoning_effort` string or null Optional Defaults to medium
o-series models only

Constrains effort on reasoning for reasoning models. Currently supported values are `low`, `medium`, and `high`. Reducing reasoning effort can result in faster responses and fewer tokens used on reasoning in a response.

`response_format` "auto" or object Optional
Specifies the format that the model must output. Compatible with GPT-4o, GPT-4 Turbo, and all GPT-3.5 Turbo models since `gpt-3.5-turbo-1106`.

Setting to `{ "type": "json_schema", "json_schema": {...} }` enables Structured Outputs which ensures the model will match your supplied JSON schema. Learn more in the Structured Outputs guide.

Setting to `{ "type": "json_object" }` enables JSON mode, which ensures the message the model generates is valid JSON.

Important: when using JSON mode, you **must** also instruct the model to produce JSON yourself via a system or user message. Without this, the model may generate an unending stream of whitespace until the generation reaches the token limit, resulting in a long-running and seemingly "stuck" request. Also note that the message content may be partially cut off if `finish_reason="length"`, which indicates the generation exceeded `max_tokens` or the conversation exceeded the max context length.

^ Hide possible types

string
`auto` is the default value

Text object
Default response format. Used to generate text responses.

Hide properties

type string Required
The type of response format being defined. Always `text`.

JSON object object
JSON object response format. An older method of generating JSON responses. Using `json_schema` is recommended for models that support it. Note that the model will not generate JSON without a system or user message instructing it to do so.

Hide properties

type string Required
The type of response format being defined. Always `json_object`.

JSON schema object
JSON Schema response format. Used to generate structured JSON responses. Learn more about Structured Outputs.

Hide properties

json_schema object Required
Structured Outputs configuration options, including a JSON Schema.

Hide properties

name string Required
The name of the response format. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

description string Optional
A description of what the response format is for, used by the model to determine how to respond in the format.

schema object Optional
The schema for the response format, described as a JSON Schema object. Learn how to build JSON schemas here.

strict boolean or null Optional Defaults to false
Whether to enable strict schema adherence when generating the output. If set to true, the model will always follow the exact schema defined in the `schema` field. Only a subset of JSON Schema is supported when `strict` is `true`. To learn more, read the Structured Outputs guide.

type string Required

The type of response format being defined. Always `json_schema`.

temperature number or null Optional Defaults to 1
What sampling temperature to use, between 0 and 2. Higher values like 0.8 will make the output more random, while lower values like 0.2 will make it more focused and deterministic.

tool_resources object or null Optional
A set of resources that are used by the assistant's tools. The resources are specific to the type of tool. For example, the `code_interpreter` tool requires a list of file IDs, while the `file_search` tool requires a list of vector store IDs.

Hide properties

code_interpreter object Optional

Hide properties

file_ids array Optional Defaults to []
A list of file IDs made available to the `code_interpreter` tool. There can be a maximum of 20 files associated with the tool.

file_search Optional

Hide possible types

tools array Optional Defaults to []
A list of tool enabled on the assistant. There can be a maximum of 128 tools per assistant. Tools can be of types `code_interpreter`, `file_search`, or `function`.

Hide possible types

Code interpreter tool object

Hide properties

type string Required
The type of tool being defined: `code_interpreter`

FileSearch tool object

Hide properties

type string Required

The type of tool being defined: `file_search`

`file_search` object Optional
Overrides for the file search tool.

Hide properties

`max_num_results` integer Optional
The maximum number of results the file search tool should output. The default is 20 for `gpt-4*` models and 5 for `gpt-3.5-turbo`. This number should be between 1 and 50 inclusive.

Note that the file search tool may output fewer than `max_num_results` results. See the file search tool documentation for more information.

`ranking_options` object Optional
The ranking options for the file search. If not specified, the file search tool will use the `auto` ranker and a score_threshold of 0.

See the file search tool documentation for more information.

Hide properties

`score_threshold` number Required
The score threshold for the file search. All values must be a floating point number between 0 and 1.

`ranker` string Optional
The ranker to use for the file search. If not specified will use the `auto` ranker.

Function tool object

Hide properties

`function` object Required

Hide properties

`name` string Required
The name of the function to be called. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

`description` string Optional
A description of what the function does, used by the model to choose when and how to call the function.

`parameters` object Optional

The parameters the functions accepts, described as a JSON Schema object. See the guide for examples, and the JSON Schema reference for documentation about the format.

Omitting `parameters` defines a function with an empty parameter list.

strict boolean or null Optional Defaults to false

Whether to enable strict schema adherence when generating the function call. If set to true, the model will follow the exact schema defined in the `parameters` field. Only a subset of JSON Schema is supported when `strict` is `true`. Learn more about Structured Outputs in the function calling guide.

type string Required

The type of tool being defined: `function`

top_p number or null Optional Defaults to 1

An alternative to sampling with temperature, called nucleus sampling, where the model considers the results of the tokens with top_p probability mass. So 0.1 means only the tokens comprising the top 10% probability mass are considered.

We generally recommend altering this or temperature but not both.

Returns

An assistant object.

Code Interpreter Files

Example request curl

```curl
curl "https://api.openai.com/v1/assistants" \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "OpenAI-Beta: assistants=v2" \
  -d '{
    "instructions": "You are a personal math tutor. When asked a question, write and run P
    "name": "Math Tutor",
    "tools": [{"type": "code_interpreter"}],
    "model": "gpt-4o"
  }'
```

Response

```
{
  "id": "asst_abc123",
  "object": "assistant",
  "created_at": 1698984975,
  "name": "Math Tutor",
  "description": null,
  "model": "gpt-4o",
  "instructions": "You are a personal math tutor. When asked a question, write and run Pyt
  "tools": [
    {
      "type": "code_interpreter"
    }
  ],
  "metadata": {},
  "top_p": 1.0,
  "temperature": 1.0,
  "response_format": "auto"
}
```

List assistants Beta

`GET https://api.openai.com/v1/assistants`

Returns a list of assistants.

Query parameters

after string Optional

A cursor for use in pagination. `after` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, ending with obj_foo, your subsequent call can include after=obj_foo in order to fetch the next page of the list.

before string Optional

A cursor for use in pagination. `before` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, starting with obj_foo, your subsequent call can include before=obj_foo in order to fetch the previous page of the list.

limit integer Optional Defaults to 20

A limit on the number of objects to be returned. Limit can range between 1 and 100, and the default is 20.

order string Optional Defaults to desc

Sort order by the `created_at` timestamp of the objects. `asc` for ascending order and `desc` for descending order.

Returns

A list of assistant objects.

Example request curl

```
curl "https://api.openai.com/v1/assistants?order=desc&limit=20" \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "OpenAI-Beta: assistants=v2"
```

Response

```
{
  "object": "list",
  "data": [
    {
      "id": "asst_abc123",
      "object": "assistant",
      "created_at": 1698982736,
      "name": "Coding Tutor",
      "description": null,
      "model": "gpt-4o",
      "instructions": "You are a helpful assistant designed to make me better at coding!",
      "tools": [],
      "tool_resources": {},
      "metadata": {},
      "top_p": 1.0,
      "temperature": 1.0,
      "response_format": "auto"
    },
    {
      "id": "asst_abc456",
      "object": "assistant",
      "created_at": 1698982718,
      "name": "My Assistant",

```

```
      "description": null,
      "model": "gpt-4o",
      "instructions": "You are a helpful assistant designed to make me better at coding!",
      "tools": [],
      "tool_resources": {},
      "metadata": {},
      "top_p": 1.0,
      "temperature": 1.0,
      "response_format": "auto"
    },
    {
      "id": "asst_abc789",
      "object": "assistant",
      "created_at": 1698982643,
      "name": null,
      "description": null,
      "model": "gpt-4o",
      "instructions": null,
      "tools": [],
      "tool_resources": {},
      "metadata": {},
      "top_p": 1.0,
      "temperature": 1.0,
      "response_format": "auto"
    }
  ],
  "first_id": "asst_abc123",
  "last_id": "asst_abc789",
  "has_more": false
}
```

Retrieve assistant Beta

```
GET https://api.openai.com/v1/assistants/{assistant_id}
```

Retrieves an assistant.

Path parameters

assistant_id string Required

The ID of the assistant to retrieve.

Returns

The assistant object matching the specified ID.

Example request curl

```
curl https://api.openai.com/v1/assistants/asst_abc123 \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "OpenAI-Beta: assistants=v2"
```

Response

```
{
  "id": "asst_abc123",
  "object": "assistant",
  "created_at": 1699009709,
  "name": "HR Helper",
  "description": null,
  "model": "gpt-4o",
  "instructions": "You are an HR bot, and you have access to files to answer employee ques
  "tools": [
    {
      "type": "file_search"
    }
  ],
  "metadata": {},
  "top_p": 1.0,
  "temperature": 1.0,
  "response_format": "auto"
}
```

Modify assistant Beta

`POST https://api.openai.com/v1/assistants/{assistant_id}`

Modifies an assistant.

Path parameters

assistant_id string Required
The ID of the assistant to modify.

Request body

description string or null Optional
The description of the assistant. The maximum length is 512 characters.

instructions string or null Optional
The system instructions that the assistant uses. The maximum length is 256,000 characters.

metadata map Optional
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

model string Optional
ID of the model to use. You can use the List models API to see all of your available models, or see our Model overview for descriptions of them.

name string or null Optional
The name of the assistant. The maximum length is 256 characters.

reasoning_effort string or null Optional Defaults to medium
o-series models only

Constrains effort on reasoning for reasoning models. Currently supported values are `low`, `medium`, and `high`. Reducing reasoning effort can result in faster responses and fewer tokens used on reasoning in a response.

response_format "auto" or object Optional
Specifies the format that the model must output. Compatible with GPT-4o, GPT-4 Turbo, and all GPT-3.5 Turbo models since `gpt-3.5-turbo-1106`.

Setting to `{ "type": "json_schema", "json_schema": {...} }` enables Structured Outputs which ensures the model will match your supplied JSON schema. Learn more in the Structured Outputs guide.

Setting to `{ "type": "json_object" }` enables JSON mode, which ensures the message the model generates is valid JSON.

Important: when using JSON mode, you **must** also instruct the model to produce JSON yourself via a system

or user message. Without this, the model may generate an unending stream of whitespace until the generation reaches the token limit, resulting in a long-running and seemingly "stuck" request. Also note that the message content may be partially cut off if `finish_reason="length"`, which indicates the generation exceeded `max_tokens` or the conversation exceeded the max context length.

Hide possible types

string

`auto` is the default value

Text object

Default response format. Used to generate text responses.

Hide properties

type string Required

The type of response format being defined. Always `text`.

JSON object object

JSON object response format. An older method of generating JSON responses. Using `json_schema` is recommended for models that support it. Note that the model will not generate JSON without a system or user message instructing it to do so.

Hide properties

type string Required

The type of response format being defined. Always `json_object`.

JSON schema object

JSON Schema response format. Used to generate structured JSON responses. Learn more about Structured Outputs.

Hide properties

json_schema object Required

Structured Outputs configuration options, including a JSON Schema.

Hide properties

name string Required

The name of the response format. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

description string Optional

A description of what the response format is for, used by the model to determine how to respond in the format.

schema object Optional
The schema for the response format, described as a JSON Schema object. Learn how to build JSON schemas here.

strict boolean or null Optional Defaults to false
Whether to enable strict schema adherence when generating the output. If set to true, the model will always follow the exact schema defined in the `schema` field. Only a subset of JSON Schema is supported when `strict` is `true`. To learn more, read the Structured Outputs guide.

type string Required
The type of response format being defined. Always `json_schema`.

temperature number or null Optional Defaults to 1
What sampling temperature to use, between 0 and 2. Higher values like 0.8 will make the output more random, while lower values like 0.2 will make it more focused and deterministic.

tool_resources object or null Optional
A set of resources that are used by the assistant's tools. The resources are specific to the type of tool. For example, the `code_interpreter` tool requires a list of file IDs, while the `file_search` tool requires a list of vector store IDs.

Hide properties

code_interpreter object Optional

Hide properties

file_ids array Optional Defaults to []
Overrides the list of file IDs made available to the `code_interpreter` tool. There can be a maximum of 20 files associated with the tool.

file_search object Optional

Hide properties

vector_store_ids array Optional
Overrides the vector store attached to this assistant. There can be a maximum of 1 vector store attached to the assistant.

tools array Optional Defaults to []
A list of tool enabled on the assistant. There can be a maximum of 128 tools per assistant. Tools can be of

types `code_interpreter`, `file_search`, or `function`.

Hide possible types

Code interpreter tool object

Hide properties

type string Required
The type of tool being defined: `code_interpreter`

FileSearch tool object

Hide properties

type string Required
The type of tool being defined: `file_search`

file_search object Optional
Overrides for the file search tool.

Hide properties

max_num_results integer Optional
The maximum number of results the file search tool should output. The default is 20 for `gpt-4*` models and 5 for `gpt-3.5-turbo`. This number should be between 1 and 50 inclusive.

Note that the file search tool may output fewer than `max_num_results` results. See the file search tool documentation for more information.

ranking_options object Optional
The ranking options for the file search. If not specified, the file search tool will use the `auto` ranker and a score_threshold of 0.

See the file search tool documentation for more information.

Hide properties

score_threshold number Required
The score threshold for the file search. All values must be a floating point number between 0 and 1.

ranker string Optional
The ranker to use for the file search. If not specified will use the `auto` ranker.

Function tool object

Hide properties

function object Required

Hide properties

name string Required
The name of the function to be called. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

description string Optional
A description of what the function does, used by the model to choose when and how to call the function.

parameters object Optional
The parameters the functions accepts, described as a JSON Schema object. See the guide for examples, and the JSON Schema reference for documentation about the format.

Omitting `parameters` defines a function with an empty parameter list.

strict boolean or null Optional Defaults to false
Whether to enable strict schema adherence when generating the function call. If set to true, the model will follow the exact schema defined in the `parameters` field. Only a subset of JSON Schema is supported when `strict` is `true`. Learn more about Structured Outputs in the function calling guide.

type string Required
The type of tool being defined: `function`

top_p number or null Optional Defaults to 1
An alternative to sampling with temperature, called nucleus sampling, where the model considers the results of the tokens with top_p probability mass. So 0.1 means only the tokens comprising the top 10% probability mass are considered.

We generally recommend altering this or temperature but not both.

Returns

The modified assistant object.

Example request curl

```
curl https://api.openai.com/v1/assistants/asst_abc123 \
```

```
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "OpenAI-Beta: assistants=v2" \
  -d '{
      "instructions": "You are an HR bot, and you have access to files to answer employee c
      "tools": [{"type": "file_search"}],
      "model": "gpt-4o"
    }'
```

Response

```
{
  "id": "asst_123",
  "object": "assistant",
  "created_at": 1699009709,
  "name": "HR Helper",
  "description": null,
  "model": "gpt-4o",
  "instructions": "You are an HR bot, and you have access to files to answer employee ques
  "tools": [
    {
      "type": "file_search"
    }
  ],
  "tool_resources": {
    "file_search": {
      "vector_store_ids": []
    }
  },
  "metadata": {},
  "top_p": 1.0,
  "temperature": 1.0,
  "response_format": "auto"
}
```

Delete assistant Beta

```
DELETE https://api.openai.com/v1/assistants/{assistant_id}
```

Delete an assistant.

Path parameters

assistant_id string Required
The ID of the assistant to delete.

Returns

Deletion status

Example request curl

```
curl https://api.openai.com/v1/assistants/asst_abc123 \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "OpenAI-Beta: assistants=v2" \
  -X DELETE
```

Response

```
{
  "id": "asst_abc123",
  "object": "assistant.deleted",
  "deleted": true
}
```

The assistant object Beta

Represents an `assistant` that can call the model and use tools.

created_at integer
The Unix timestamp (in seconds) for when the assistant was created.

description string or null
The description of the assistant. The maximum length is 512 characters.

id string
The identifier, which can be referenced in API endpoints.

instructions string or null

The system instructions that the assistant uses. The maximum length is 256,000 characters.

`metadata` map

Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

`model` string

ID of the model to use. You can use the List models API to see all of your available models, or see our Model overview for descriptions of them.

`name` string or null

The name of the assistant. The maximum length is 256 characters.

`object` string

The object type, which is always `assistant`.

`response_format` "auto" or object

Specifies the format that the model must output. Compatible with GPT-4o, GPT-4 Turbo, and all GPT-3.5 Turbo models since `gpt-3.5-turbo-1106`.

Setting to `{ "type": "json_schema", "json_schema": {...} }` enables Structured Outputs which ensures the model will match your supplied JSON schema. Learn more in the Structured Outputs guide.

Setting to `{ "type": "json_object" }` enables JSON mode, which ensures the message the model generates is valid JSON.

Important: when using JSON mode, you **must** also instruct the model to produce JSON yourself via a system or user message. Without this, the model may generate an unending stream of whitespace until the generation reaches the token limit, resulting in a long-running and seemingly "stuck" request. Also note that the message content may be partially cut off if `finish_reason="length"`, which indicates the generation exceeded `max_tokens` or the conversation exceeded the max context length.

Hide possible types

string

`auto` is the default value

Text object

Default response format. Used to generate text responses.

Hide properties

`type` string

The type of response format being defined. Always `text`.

JSON object object

JSON object response format. An older method of generating JSON responses. Using `json_schema` is recommended for models that support it. Note that the model will not generate JSON without a system or user message instructing it to do so.

Hide properties

type string

The type of response format being defined. Always `json_object`.

JSON schema object

JSON Schema response format. Used to generate structured JSON responses. Learn more about Structured Outputs.

Hide properties

json_schema object

Structured Outputs configuration options, including a JSON Schema.

Hide properties

name string

The name of the response format. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

description string

A description of what the response format is for, used by the model to determine how to respond in the format.

schema object

The schema for the response format, described as a JSON Schema object. Learn how to build JSON schemas here.

strict boolean or null

Whether to enable strict schema adherence when generating the output. If set to true, the model will always follow the exact schema defined in the `schema` field. Only a subset of JSON Schema is supported when `strict` is `true`. To learn more, read the Structured Outputs guide.

type string

The type of response format being defined. Always `json_schema`.

temperature number or null
What sampling temperature to use, between 0 and 2. Higher values like 0.8 will make the output more random, while lower values like 0.2 will make it more focused and deterministic.

tool_resources object or null
A set of resources that are used by the assistant's tools. The resources are specific to the type of tool. For example, the `code_interpreter` tool requires a list of file IDs, while the `file_search` tool requires a list of vector store IDs.

Hide properties

code_interpreter object

Hide properties

file_ids array
A list of file IDs made available to the `code_interpreter`` tool. There can be a maximum of 20 files associated with the tool.

file_search object

Hide properties

vector_store_ids array
The ID of the vector store attached to this assistant. There can be a maximum of 1 vector store attached to the assistant.

tools array
A list of tool enabled on the assistant. There can be a maximum of 128 tools per assistant. Tools can be of types `code_interpreter`, `file_search`, or `function`.

Hide possible types

Code interpreter tool object

Hide properties

type string
The type of tool being defined: `code_interpreter`

FileSearch tool object

Hide properties

type string
The type of tool being defined: `file_search`

file_search object
Overrides for the file search tool.

Hide properties

max_num_results integer
The maximum number of results the file search tool should output. The default is 20 for `gpt-4*` models and 5 for `gpt-3.5-turbo`. This number should be between 1 and 50 inclusive.

Note that the file search tool may output fewer than `max_num_results` results. See the file search tool documentation for more information.

ranking_options object
The ranking options for the file search. If not specified, the file search tool will use the `auto` ranker and a score_threshold of 0.

See the file search tool documentation for more information.

Hide properties

score_threshold number
The score threshold for the file search. All values must be a floating point number between 0 and 1.

ranker string
The ranker to use for the file search. If not specified will use the `auto` ranker.

Function tool object

Hide properties

function object

Hide properties

name string
The name of the function to be called. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

description string
A description of what the function does, used by the model to choose when and how to call the function.

parameters object
The parameters the functions accepts, described as a JSON Schema object. See the guide for examples, and the JSON Schema reference for documentation about the format.

Omitting `parameters` defines a function with an empty parameter list.

strict boolean or null
Whether to enable strict schema adherence when generating the function call. If set to true, the model will follow the exact schema defined in the `parameters` field. Only a subset of JSON Schema is supported when `strict` is `true`. Learn more about Structured Outputs in the function calling guide.

type string
The type of tool being defined: `function`

top_p number or null
An alternative to sampling with temperature, called nucleus sampling, where the model considers the results of the tokens with top_p probability mass. So 0.1 means only the tokens comprising the top 10% probability mass are considered.

We generally recommend altering this or temperature but not both.

OBJECT The assistant object

```
{
  "id": "asst_abc123",
  "object": "assistant",
  "created_at": 1698984975,
  "name": "Math Tutor",
  "description": null,
  "model": "gpt-4o",
  "instructions": "You are a personal math tutor. When asked a question, write and run Pyt
  "tools": [
    {
      "type": "code_interpreter"
    }
  ],
  "metadata": {},
  "top_p": 1.0,
  "temperature": 1.0,
  "response_format": "auto"
}
```

Threads Beta

Create threads that assistants can interact with.

Related guide: Assistants

Create thread Beta

`POST https://api.openai.com/v1/threads`

Create a thread.

Request body

---

messages array Optional
A list of messages to start the thread with.

Hide properties

content string or array Required

Hide possible types

Text content string
The text contents of the message.

Array of content parts array
An array of content parts with a defined type, each can be of type `text` or images can be passed with `image_url` or `image_file`. Image types are only supported on Vision-compatible models.

Hide possible types

Image file object
References an image File in the content of a message.

Hide properties

image_file object Required

Hide properties

file_id string Required
The File ID of the image in the message content. Set `purpose="vision"` when uploading the File if you need to later display the file content.

detail string Optional Defaults to auto
Specifies the detail level of the image if specified by the user. `low` uses fewer tokens, you can opt in to high resolution using `high`.

type string Required
Always `image_file`.

Image URL object
References an image URL in the content of a message.

Hide properties

image_url object Required

Hide properties

url string Required
The external URL of the image, must be a supported image types: jpeg, jpg, png, gif, webp.

detail string Optional Defaults to auto
Specifies the detail level of the image. `low` uses fewer tokens, you can opt in to high resolution using `high`. Default value is `auto`

type string Required
The type of the content part.

Text object
The text content that is part of a message.

Hide properties

text string Required
Text content to be sent to the model

type string Required
Always `text`.

role string Required
The role of the entity that is creating the message. Allowed values include:

`user`: Indicates the message is sent by an actual user and should be used in most cases to

represent user-generated messages.

`assistant` : Indicates the message is generated by the assistant. Use this value to insert messages from the assistant into the conversation.

attachments array or null Optional

A list of files attached to the message, and the tools they should be added to.

Hide properties

file_id string Optional

The ID of the file to attach to the message.

tools array Optional

The tools to add this file to.

Hide possible types

Code interpreter tool object

Hide properties

type string Required

The type of tool being defined: `code_interpreter`

FileSearch tool object

Hide properties

type string Required

The type of tool being defined: `file_search`

metadata map Optional

Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

metadata map Optional

Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

tool_resources object or null Optional

A set of resources that are made available to the assistant's tools in this thread. The resources are specific to the type of tool. For example, the `code_interpreter` tool requires a list of file IDs, while the `file_search` tool requires a list of vector store IDs.

Hide properties

code_interpreter object Optional

Hide properties

file_ids array Optional Defaults to []

A list of file IDs made available to the `code_interpreter` tool. There can be a maximum of 20 files associated with the tool.

file_search Optional

Hide possible types

Returns

A thread object.

Empty Messages

Example request curl

```
curl https://api.openai.com/v1/threads \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "OpenAI-Beta: assistants=v2" \
  -d ''
```

Response

```
{
  "id": "thread_abc123",
  "object": "thread",
  "created_at": 1699012949,
  "metadata": {},
```

```
  "tool_resources": {}
}
```

Retrieve thread Beta

```
GET https://api.openai.com/v1/threads/{thread_id}
```

Retrieves a thread.

Path parameters

thread_id string Required
The ID of the thread to retrieve.

Returns

The thread object matching the specified ID.

Example request curl

```curl
curl https://api.openai.com/v1/threads/thread_abc123 \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "OpenAI-Beta: assistants=v2"
```

Response

```response
{
  "id": "thread_abc123",
  "object": "thread",
  "created_at": 1699014083,
  "metadata": {},
  "tool_resources": {
    "code_interpreter": {
      "file_ids": []
    }
  }
}
```

Modify thread Beta

```
POST https://api.openai.com/v1/threads/{thread_id}
```

Modifies a thread.

Path parameters

thread_id string Required
The ID of the thread to modify. Only the `metadata` can be modified.

Request body

metadata map Optional
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

tool_resources object or null Optional
A set of resources that are made available to the assistant's tools in this thread. The resources are specific to the type of tool. For example, the `code_interpreter` tool requires a list of file IDs, while the `file_search` tool requires a list of vector store IDs.

Hide properties

code_interpreter object Optional

Hide properties

file_ids array Optional Defaults to []
A list of file IDs made available to the `code_interpreter` tool. There can be a maximum of 20 files associated with the tool.

file_search object Optional

Hide properties

vector_store_ids array Optional
The vector store attached to this thread. There can be a maximum of 1 vector store attached to the thread.

Returns

The modified thread object matching the specified ID.

Example request curl

```curl
curl https://api.openai.com/v1/threads/thread_abc123 \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "OpenAI-Beta: assistants=v2" \
  -d '{
      "metadata": {
        "modified": "true",
        "user": "abc123"
      }
    }'
```

Response

```response
{
  "id": "thread_abc123",
  "object": "thread",
  "created_at": 1699014083,
  "metadata": {
    "modified": "true",
    "user": "abc123"
  },
  "tool_resources": {}
}
```

Delete thread Beta

`DELETE https://api.openai.com/v1/threads/{thread_id}`

Delete a thread.

Path parameters

thread_id string Required
The ID of the thread to delete.

Returns

Deletion status

Example request curl

```
curl https://api.openai.com/v1/threads/thread_abc123 \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "OpenAI-Beta: assistants=v2" \
  -X DELETE
```

Response

```
{
  "id": "thread_abc123",
  "object": "thread.deleted",
  "deleted": true
}
```

The thread object Beta

Represents a thread that contains messages.

created_at integer
The Unix timestamp (in seconds) for when the thread was created.

id string
The identifier, which can be referenced in API endpoints.

metadata map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512

characters.

object string
The object type, which is always `thread`.

tool_resources object or null
A set of resources that are made available to the assistant's tools in this thread. The resources are specific to the type of tool. For example, the `code_interpreter` tool requires a list of file IDs, while the `file_search` tool requires a list of vector store IDs.

Hide properties

code_interpreter object

Hide properties

file_ids array
A list of file IDs made available to the `code_interpreter` tool. There can be a maximum of 20 files associated with the tool.

file_search object

Hide properties

vector_store_ids array
The vector store attached to this thread. There can be a maximum of 1 vector store attached to the thread.

OBJECT The thread object

```
{
  "id": "thread_abc123",
  "object": "thread",
  "created_at": 1698107661,
  "metadata": {}
}
```

Messages Beta

Create messages within threads

Related guide: Assistants

Create message Beta

```
POST https://api.openai.com/v1/threads/{thread_id}/messages
```

Create a message.

Path parameters

thread_id string Required
The ID of the thread to create a message for.

Request body

content string or array Required

Hide possible types

Text content string
The text contents of the message.

Array of content parts array
An array of content parts with a defined type, each can be of type `text` or images can be passed with `image_url` or `image_file`. Image types are only supported on Vision-compatible models.

Hide possible types

Image file object
References an image File in the content of a message.

Hide properties

image_file object Required

Hide properties

file_id string Required

The File ID of the image in the message content. Set `purpose="vision"` when uploading the File if you need to later display the file content.

`detail` string Optional Defaults to auto
Specifies the detail level of the image if specified by the user. `low` uses fewer tokens, you can opt in to high resolution using `high`.

`type` string Required
Always `image_file`.

Image URL object
References an image URL in the content of a message.

Hide properties

`image_url` object Required

Hide properties

`url` string Required
The external URL of the image, must be a supported image types: jpeg, jpg, png, gif, webp.

`detail` string Optional Defaults to auto
Specifies the detail level of the image. `low` uses fewer tokens, you can opt in to high resolution using `high`. Default value is `auto`

`type` string Required
The type of the content part.

Text object
The text content that is part of a message.

Hide properties

`text` string Required
Text content to be sent to the model

`type` string Required
Always `text`.

`role` string Required
The role of the entity that is creating the message. Allowed values include:

`user` : Indicates the message is sent by an actual user and should be used in most cases to represent user-generated messages.

`assistant` : Indicates the message is generated by the assistant. Use this value to insert messages from the assistant into the conversation.

---

attachments array or null Optional

A list of files attached to the message, and the tools they should be added to.

Hide properties

file_id string Optional

The ID of the file to attach to the message.

tools array Optional

The tools to add this file to.

Hide possible types

Code interpreter tool object

Hide properties

type string Required

The type of tool being defined: `code_interpreter`

FileSearch tool object

Hide properties

type string Required

The type of tool being defined: `file_search`

---

metadata map Optional

Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

Returns

A message object.

Example request curl

```curl
curl https://api.openai.com/v1/threads/thread_abc123/messages \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "OpenAI-Beta: assistants=v2" \
  -d '{
      "role": "user",
      "content": "How does AI work? Explain it in simple terms."
    }'
```

Response

```response_object_1
{
  "id": "msg_abc123",
  "object": "thread.message",
  "created_at": 1713226573,
  "assistant_id": null,
  "thread_id": "thread_abc123",
  "run_id": null,
  "role": "user",
  "content": [
    {
      "type": "text",
      "text": {
        "value": "How does AI work? Explain it in simple terms.",
        "annotations": []
      }
    }
  ],
  "attachments": [],
  "metadata": {}
}
```

List messages Beta

GET https://api.openai.com/v1/threads/{thread_id}/messages

Returns a list of messages for a given thread.

Path parameters

thread_id string Required
The ID of the thread the messages belong to.

Query parameters

after string Optional
A cursor for use in pagination. `after` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, ending with obj_foo, your subsequent call can include after=obj_foo in order to fetch the next page of the list.

before string Optional
A cursor for use in pagination. `before` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, starting with obj_foo, your subsequent call can include before=obj_foo in order to fetch the previous page of the list.

limit integer Optional Defaults to 20
A limit on the number of objects to be returned. Limit can range between 1 and 100, and the default is 20.

order string Optional Defaults to desc
Sort order by the `created_at` timestamp of the objects. `asc` for ascending order and `desc` for descending order.

run_id string Optional
Filter messages by the run ID that generated them.

Returns

A list of message objects.

Example request curl

```curl
curl https://api.openai.com/v1/threads/thread_abc123/messages \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "OpenAI-Beta: assistants=v2"
```

Response


```
{
  "object": "list",
  "data": [
    {
      "id": "msg_abc123",
      "object": "thread.message",
      "created_at": 1699016383,
      "assistant_id": null,
      "thread_id": "thread_abc123",
      "run_id": null,
      "role": "user",
      "content": [
        {
          "type": "text",
          "text": {
            "value": "How does AI work? Explain it in simple terms.",
            "annotations": []
          }
        }
      ],
      "attachments": [],
      "metadata": {}
    },
    {
      "id": "msg_abc456",
      "object": "thread.message",
      "created_at": 1699016383,
      "assistant_id": null,
      "thread_id": "thread_abc123",
      "run_id": null,
      "role": "user",
      "content": [
        {
          "type": "text",
          "text": {
            "value": "Hello, what is AI?",
            "annotations": []
          }
        }
      ],
      "attachments": [],
      "metadata": {}
    }
  ],
```

```
  "first_id": "msg_abc123",
  "last_id": "msg_abc456",
  "has_more": false
}
```

Retrieve message Beta

```
GET https://api.openai.com/v1/threads/{thread_id}/messages/{message_id}
```

Retrieve a message.

Path parameters

message_id string Required
The ID of the message to retrieve.

thread_id string Required
The ID of the thread to which this message belongs.

Returns

The message object matching the specified ID.

Example request curl

```
curl https://api.openai.com/v1/threads/thread_abc123/messages/msg_abc123 \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "OpenAI-Beta: assistants=v2"
```

Response

```
{
  "id": "msg_abc123",
  "object": "thread.message",
  "created_at": 1699017614,
  "assistant_id": null,
  "thread_id": "thread_abc123",
```

```
  "run_id": null,
  "role": "user",
  "content": [
    {
      "type": "text",
      "text": {
        "value": "How does AI work? Explain it in simple terms.",
        "annotations": []
      }
    }
  ],
  "attachments": [],
  "metadata": {}
}
```

Modify message Beta

```
POST https://api.openai.com/v1/threads/{thread_id}/messages/{message_id}
```

Modifies a message.

Path parameters

`message_id` string Required
The ID of the message to modify.

`thread_id` string Required
The ID of the thread to which this message belongs.

Request body

`metadata` map Optional
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

Returns

The modified message object.

Example request curl

```curl
curl https://api.openai.com/v1/threads/thread_abc123/messages/msg_abc123 \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "OpenAI-Beta: assistants=v2" \
  -d '{
      "metadata": {
        "modified": "true",
        "user": "abc123"
      }
    }'
```

Response

```response_object_example
{
  "id": "msg_abc123",
  "object": "thread.message",
  "created_at": 1699017614,
  "assistant_id": null,
  "thread_id": "thread_abc123",
  "run_id": null,
  "role": "user",
  "content": [
    {
      "type": "text",
      "text": {
        "value": "How does AI work? Explain it in simple terms.",
        "annotations": []
      }
    }
  ],
  "file_ids": [],
  "metadata": {
    "modified": "true",
    "user": "abc123"
  }
}
```

Delete message Beta

```
DELETE https://api.openai.com/v1/threads/{thread_id}/messages/{message_id}
```

Deletes a message.

Path parameters

message_id string Required
The ID of the message to delete.

thread_id string Required
The ID of the thread to which this message belongs.

Returns

Deletion status

Example request curl

```curl
curl -X DELETE https://api.openai.com/v1/threads/thread_abc123/messages/msg_abc123 \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "OpenAI-Beta: assistants=v2"
```

Response

```
{
  "id": "msg_abc123",
  "object": "thread.message.deleted",
  "deleted": true
}
```

The message object Beta

Represents a message within a thread.

assistant_id string or null
If applicable, the ID of the assistant that authored this message.

attachments array or null
A list of files attached to the message, and the tools they were added to.

Hide properties

file_id string
The ID of the file to attach to the message.

tools array
The tools to add this file to.

Hide possible types

Code interpreter tool object

Hide properties

type string
The type of tool being defined: `code_interpreter`

FileSearch tool object

Hide properties

type string
The type of tool being defined: `file_search`

completed_at integer or null
The Unix timestamp (in seconds) for when the message was completed.

content array
The content of the message in array of text and/or images.

Hide possible types

Image file object
References an image File in the content of a message.

Hide properties

image_file object

Hide properties

file_id string
The File ID of the image in the message content. Set `purpose="vision"` when uploading the File if you need to later display the file content.

detail string
Specifies the detail level of the image if specified by the user. `low` uses fewer tokens, you can opt in to high resolution using `high`.

type string
Always `image_file`.

Image URL object
References an image URL in the content of a message.

Hide properties

image_url object

Hide properties

url string
The external URL of the image, must be a supported image types: jpeg, jpg, png, gif, webp.

detail string
Specifies the detail level of the image. `low` uses fewer tokens, you can opt in to high resolution using `high`. Default value is `auto`

type string
The type of the content part.

Text object
The text content that is part of a message.

Hide properties

text object

Hide properties

annotations array

Hide possible types

File citation object
A citation within the message that points to a specific quote from a specific File associated

with the assistant or the message. Generated when the assistant uses the "file_search" tool to search files.

Hide properties

end_index integer

file_citation object

Hide properties

file_id string
The ID of the specific File the citation is from.

start_index integer

text string
The text in the message content that needs to be replaced.

type string
Always `file_citation`.

File path object
A URL for the file that's generated when the assistant used the `code_interpreter` tool to generate a file.

Hide properties

end_index integer

file_path object

Hide properties

file_id string
The ID of the file that was generated.

start_index integer

text string
The text in the message content that needs to be replaced.

type string
Always `file_path`.

value string
The data that makes up the text.

type string
Always `text`.

Refusal object
The refusal content generated by the assistant.

Hide properties

refusal string

type string
Always `refusal`.

created_at integer
The Unix timestamp (in seconds) for when the message was created.

id string
The identifier, which can be referenced in API endpoints.

incomplete_at integer or null
The Unix timestamp (in seconds) for when the message was marked as incomplete.

incomplete_details object or null
On an incomplete message, details about why the message is incomplete.

Hide properties

reason string
The reason the message is incomplete.

metadata map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

object string

The object type, which is always `thread.message`.

role string
The entity that produced the message. One of `user` or `assistant`.

run_id string or null
The ID of the run associated with the creation of this message. Value is `null` when messages are created manually using the create message or create thread endpoints.

status string
The status of the message, which can be either `in_progress`, `incomplete`, or `completed`.

thread_id string
The thread ID that this message belongs to.

OBJECT The message object

```
{
  "id": "msg_abc123",
  "object": "thread.message",
  "created_at": 1698983503,
  "thread_id": "thread_abc123",
  "role": "assistant",
  "content": [
    {
      "type": "text",
      "text": {
        "value": "Hi! How can I help you today?",
        "annotations": []
      }
    }
  ],
  "assistant_id": "asst_abc123",
  "run_id": "run_abc123",
  "attachments": [],
  "metadata": {}
}
```

Runs Beta

Represents an execution run on a thread.

Related guide: Assistants

Create run Beta

`POST https://api.openai.com/v1/threads/{thread_id}/runs`

Create a run.

Path parameters

`thread_id` string Required
The ID of the thread to run.

Query parameters

`include[]` array Optional
A list of additional fields to include in the response. Currently the only supported value is `step_details.tool_calls[*].file_search.results[*].content` to fetch the file search result content.

See the file search tool documentation for more information.

Request body

`assistant_id` string Required
The ID of the assistant to use to execute this run.

`additional_instructions` string or null Optional
Appends additional instructions at the end of the instructions for the run. This is useful for modifying the behavior on a per-run basis without overriding other instructions.

`additional_messages` array or null Optional
Adds additional messages to the thread before creating the run.

Hide properties

`content` string or array Required

Hide possible types

Text content string

The text contents of the message.

Array of content parts array

An array of content parts with a defined type, each can be of type `text` or images can be passed with `image_url` or `image_file`. Image types are only supported on Vision-compatible models.

Hide possible types

Image file object

References an image File in the content of a message.

Hide properties

image_file object Required

Hide properties

file_id string Required

The File ID of the image in the message content. Set `purpose="vision"` when uploading the File if you need to later display the file content.

detail string Optional Defaults to auto

Specifies the detail level of the image if specified by the user. `low` uses fewer tokens, you can opt in to high resolution using `high`.

type string Required

Always `image_file`.

Image URL object

References an image URL in the content of a message.

Hide properties

image_url object Required

Hide properties

url string Required

The external URL of the image, must be a supported image types: jpeg, jpg, png, gif, webp.

detail string Optional Defaults to auto

Specifies the detail level of the image. `low` uses fewer tokens, you can opt in to high resolution using `high`. Default value is `auto`

type string Required

The type of the content part.

Text object

The text content that is part of a message.

Hide properties

text string Required

Text content to be sent to the model

type string Required

Always `text`.

role string Required

The role of the entity that is creating the message. Allowed values include:

`user`: Indicates the message is sent by an actual user and should be used in most cases to represent user-generated messages.

`assistant`: Indicates the message is generated by the assistant. Use this value to insert messages from the assistant into the conversation.

attachments array or null Optional

A list of files attached to the message, and the tools they should be added to.

Hide properties

file_id string Optional

The ID of the file to attach to the message.

tools array Optional

The tools to add this file to.

Hide possible types

Code interpreter tool object

Hide properties

type string Required

The type of tool being defined: `code_interpreter`

FileSearch tool object

Hide properties

type string Required
The type of tool being defined: `file_search`

metadata map Optional
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

instructions string or null Optional
Overrides the instructions of the assistant. This is useful for modifying the behavior on a per-run basis.

max_completion_tokens integer or null Optional
The maximum number of completion tokens that may be used over the course of the run. The run will make a best effort to use only the number of completion tokens specified, across multiple turns of the run. If the run exceeds the number of completion tokens specified, the run will end with status `incomplete`. See `incomplete_details` for more info.

max_prompt_tokens integer or null Optional
The maximum number of prompt tokens that may be used over the course of the run. The run will make a best effort to use only the number of prompt tokens specified, across multiple turns of the run. If the run exceeds the number of prompt tokens specified, the run will end with status `incomplete`. See `incomplete_details` for more info.

metadata map Optional
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

model string Optional
The ID of the Model to be used to execute this run. If a value is provided here, it will override the model associated with the assistant. If not, the model associated with the assistant will be used.

parallel_tool_calls boolean Optional Defaults to true
Whether to enable parallel function calling during tool use.

reasoning_effort string or null Optional Defaults to medium

o-series models only

Constrains effort on reasoning for reasoning models. Currently supported values are `low`, `medium`, and `high`. Reducing reasoning effort can result in faster responses and fewer tokens used on reasoning in a response.

`response_format` "auto" or object Optional

Specifies the format that the model must output. Compatible with GPT-4o, GPT-4 Turbo, and all GPT-3.5 Turbo models since `gpt-3.5-turbo-1106`.

Setting to `{ "type": "json_schema", "json_schema": {...} }` enables Structured Outputs which ensures the model will match your supplied JSON schema. Learn more in the Structured Outputs guide.

Setting to `{ "type": "json_object" }` enables JSON mode, which ensures the message the model generates is valid JSON.

Important: when using JSON mode, you **must** also instruct the model to produce JSON yourself via a system or user message. Without this, the model may generate an unending stream of whitespace until the generation reaches the token limit, resulting in a long-running and seemingly "stuck" request. Also note that the message content may be partially cut off if `finish_reason="length"`, which indicates the generation exceeded `max_tokens` or the conversation exceeded the max context length.

^ Hide possible types

string

`auto` is the default value

Text object

Default response format. Used to generate text responses.

^ Hide properties

`type` string Required

The type of response format being defined. Always `text`.

JSON object object

JSON object response format. An older method of generating JSON responses. Using `json_schema` is recommended for models that support it. Note that the model will not generate JSON without a system or user message instructing it to do so.

^ Hide properties

`type` string Required

The type of response format being defined. Always `json_object`.

JSON schema object

JSON Schema response format. Used to generate structured JSON responses. Learn more about

Structured Outputs.

Hide properties

`json_schema` object Required

Structured Outputs configuration options, including a JSON Schema.

Hide properties

`name` string Required

The name of the response format. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

`description` string Optional

A description of what the response format is for, used by the model to determine how to respond in the format.

`schema` object Optional

The schema for the response format, described as a JSON Schema object. Learn how to build JSON schemas here.

`strict` boolean or null Optional Defaults to false

Whether to enable strict schema adherence when generating the output. If set to true, the model will always follow the exact schema defined in the `schema` field. Only a subset of JSON Schema is supported when `strict` is `true`. To learn more, read the Structured Outputs guide.

`type` string Required

The type of response format being defined. Always `json_schema`.

`stream` boolean or null Optional

If `true`, returns a stream of events that happen during the Run as server-sent events, terminating when the Run enters a terminal state with a `data: [DONE]` message.

`temperature` number or null Optional Defaults to 1

What sampling temperature to use, between 0 and 2. Higher values like 0.8 will make the output more random, while lower values like 0.2 will make it more focused and deterministic.

`tool_choice` string or object Optional

Controls which (if any) tool is called by the model. `none` means the model will not call any tools and instead generates a message. `auto` is the default value and means the model can pick between generating a message or calling one or more tools. `required` means the model must call one or more tools before responding to the user. Specifying a particular tool like `{"type": "file_search"}` or `{"type":`

`"function", "function": {"name": "my_function"}}` forces the model to call that tool.

Hide possible types

string

`none` means the model will not call any tools and instead generates a message. `auto` means the model can pick between generating a message or calling one or more tools. `required` means the model must call one or more tools before responding to the user.

object

Specifies a tool the model should use. Use to force the model to call a specific tool.

Hide properties

type string Required

The type of the tool. If type is `function`, the function name must be set

function object Optional

Hide properties

name string Required

The name of the function to call.

tools array or null Optional

Override the tools the assistant can use for this run. This is useful for modifying the behavior on a per-run basis.

Hide possible types

Code interpreter tool object

Hide properties

type string Required

The type of tool being defined: `code_interpreter`

FileSearch tool object

Hide properties

type string Required

The type of tool being defined: `file_search`

file_search object Optional

Overrides for the file search tool.

Hide properties

max_num_results integer Optional
The maximum number of results the file search tool should output. The default is 20 for `gpt-4*` models and 5 for `gpt-3.5-turbo`. This number should be between 1 and 50 inclusive.

Note that the file search tool may output fewer than `max_num_results` results. See the file search tool documentation for more information.

ranking_options object Optional
The ranking options for the file search. If not specified, the file search tool will use the `auto` ranker and a score_threshold of 0.

See the file search tool documentation for more information.

Hide properties

score_threshold number Required
The score threshold for the file search. All values must be a floating point number between 0 and 1.

ranker string Optional
The ranker to use for the file search. If not specified will use the `auto` ranker.

Function tool object

Hide properties

function object Required

Hide properties

name string Required
The name of the function to be called. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

description string Optional
A description of what the function does, used by the model to choose when and how to call the function.

parameters object Optional
The parameters the functions accepts, described as a JSON Schema object. See the guide for examples, and the JSON Schema reference for documentation about the format.

Omitting `parameters` defines a function with an empty parameter list.

strict boolean or null Optional Defaults to false
Whether to enable strict schema adherence when generating the function call. If set to true, the model will follow the exact schema defined in the `parameters` field. Only a subset of JSON Schema is supported when `strict` is `true`. Learn more about Structured Outputs in the function calling guide.

type string Required
The type of tool being defined: `function`

top_p number or null Optional Defaults to 1
An alternative to sampling with temperature, called nucleus sampling, where the model considers the results of the tokens with top_p probability mass. So 0.1 means only the tokens comprising the top 10% probability mass are considered.

We generally recommend altering this or temperature but not both.

truncation_strategy object or null Optional
Controls for how a thread will be truncated prior to the run. Use this to control the intial context window of the run.

Hide properties

type string Required
The truncation strategy to use for the thread. The default is `auto`. If set to `last_messages`, the thread will be truncated to the n most recent messages in the thread. When set to `auto`, messages in the middle of the thread will be dropped to fit the context length of the model, `max_prompt_tokens`.

last_messages integer or null Optional
The number of most recent messages from the thread when constructing the context for the run.

Returns

A run object.

Default Streaming Streaming with Functions

Example request curl

```
curl https://api.openai.com/v1/threads/thread_abc123/runs \
```

```
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -H "OpenAI-Beta: assistants=v2" \
  -d '{
    "assistant_id": "asst_abc123"
  }'
```

Response

```
{
  "id": "run_abc123",
  "object": "thread.run",
  "created_at": 1699063290,
  "assistant_id": "asst_abc123",
  "thread_id": "thread_abc123",
  "status": "queued",
  "started_at": 1699063290,
  "expires_at": null,
  "cancelled_at": null,
  "failed_at": null,
  "completed_at": 1699063291,
  "last_error": null,
  "model": "gpt-4o",
  "instructions": null,
  "incomplete_details": null,
  "tools": [
    {
      "type": "code_interpreter"
    }
  ],
  "metadata": {},
  "usage": null,
  "temperature": 1.0,
  "top_p": 1.0,
  "max_prompt_tokens": 1000,
  "max_completion_tokens": 1000,
  "truncation_strategy": {
    "type": "auto",
    "last_messages": null
  },
  "response_format": "auto",
  "tool_choice": "auto",
  "parallel_tool_calls": true
}
```

Create thread and run Beta

```
POST https://api.openai.com/v1/threads/runs
```

Create a thread and run it in one request.

Request body

assistant_id string Required
The ID of the assistant to use to execute this run.

instructions string or null Optional
Override the default system message of the assistant. This is useful for modifying the behavior on a per-run basis.

max_completion_tokens integer or null Optional
The maximum number of completion tokens that may be used over the course of the run. The run will make a best effort to use only the number of completion tokens specified, across multiple turns of the run. If the run exceeds the number of completion tokens specified, the run will end with status `incomplete`. See `incomplete_details` for more info.

max_prompt_tokens integer or null Optional
The maximum number of prompt tokens that may be used over the course of the run. The run will make a best effort to use only the number of prompt tokens specified, across multiple turns of the run. If the run exceeds the number of prompt tokens specified, the run will end with status `incomplete`. See `incomplete_details` for more info.

metadata map Optional
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

model string Optional
The ID of the Model to be used to execute this run. If a value is provided here, it will override the model associated with the assistant. If not, the model associated with the assistant will be used.

parallel_tool_calls boolean Optional Defaults to true
Whether to enable parallel function calling during tool use.

`response_format` "auto" or object Optional

Specifies the format that the model must output. Compatible with GPT-4o, GPT-4 Turbo, and all GPT-3.5 Turbo models since `gpt-3.5-turbo-1106`.

Setting to `{ "type": "json_schema", "json_schema": {...} }` enables Structured Outputs which ensures the model will match your supplied JSON schema. Learn more in the Structured Outputs guide.

Setting to `{ "type": "json_object" }` enables JSON mode, which ensures the message the model generates is valid JSON.

Important: when using JSON mode, you **must** also instruct the model to produce JSON yourself via a system or user message. Without this, the model may generate an unending stream of whitespace until the generation reaches the token limit, resulting in a long-running and seemingly "stuck" request. Also note that the message content may be partially cut off if `finish_reason="length"`, which indicates the generation exceeded `max_tokens` or the conversation exceeded the max context length.

Hide possible types

string

`auto` is the default value

Text object

Default response format. Used to generate text responses.

Hide properties

`type` string Required

The type of response format being defined. Always `text`.

JSON object object

JSON object response format. An older method of generating JSON responses. Using `json_schema` is recommended for models that support it. Note that the model will not generate JSON without a system or user message instructing it to do so.

Hide properties

`type` string Required

The type of response format being defined. Always `json_object`.

JSON schema object

JSON Schema response format. Used to generate structured JSON responses. Learn more about Structured Outputs.

Hide properties

`json_schema` object Required

Structured Outputs configuration options, including a JSON Schema.

Hide properties

name string Required
The name of the response format. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

description string Optional
A description of what the response format is for, used by the model to determine how to respond in the format.

schema object Optional
The schema for the response format, described as a JSON Schema object. Learn how to build JSON schemas here.

strict boolean or null Optional Defaults to false
Whether to enable strict schema adherence when generating the output. If set to true, the model will always follow the exact schema defined in the `schema` field. Only a subset of JSON Schema is supported when `strict` is `true`. To learn more, read the Structured Outputs guide.

type string Required
The type of response format being defined. Always `json_schema`.

stream boolean or null Optional
If `true`, returns a stream of events that happen during the Run as server-sent events, terminating when the Run enters a terminal state with a `data: [DONE]` message.

temperature number or null Optional Defaults to 1
What sampling temperature to use, between 0 and 2. Higher values like 0.8 will make the output more random, while lower values like 0.2 will make it more focused and deterministic.

thread object Optional
Options to create a new thread. If no thread is provided when running a request, an empty thread will be created.

Hide properties

messages array Optional
A list of messages to start the thread with.

Hide properties

content string or array Required

Hide possible types

Text content string
The text contents of the message.

Array of content parts array
An array of content parts with a defined type, each can be of type `text` or images can be passed with `image_url` or `image_file`. Image types are only supported on Vision-compatible models.

Hide possible types

Image file object
References an image File in the content of a message.

Hide properties

image_file object Required

Hide properties

file_id string Required
The File ID of the image in the message content. Set `purpose="vision"` when uploading the File if you need to later display the file content.

detail string Optional Defaults to auto
Specifies the detail level of the image if specified by the user. `low` uses fewer tokens, you can opt in to high resolution using `high`.

type string Required
Always `image_file`.

Image URL object
References an image URL in the content of a message.

Hide properties

image_url object Required

Hide properties

url string Required
The external URL of the image, must be a supported image types: jpeg, jpg, png, gif, webp.

detail string Optional Defaults to auto
Specifies the detail level of the image. `low` uses fewer tokens, you can opt in to high

resolution using `high`. Default value is `auto`

type string Required
The type of the content part.

Text object
The text content that is part of a message.

^ Hide properties

text string Required
Text content to be sent to the model

type string Required
Always `text`.

role string Required
The role of the entity that is creating the message. Allowed values include:

`user`: Indicates the message is sent by an actual user and should be used in most cases to represent user-generated messages.

`assistant`: Indicates the message is generated by the assistant. Use this value to insert messages from the assistant into the conversation.

attachments array or null Optional
A list of files attached to the message, and the tools they should be added to.

^ Hide properties

file_id string Optional
The ID of the file to attach to the message.

tools array Optional
The tools to add this file to.

^ Hide possible types

Code interpreter tool object

^ Hide properties

type string Required
The type of tool being defined: `code_interpreter`

FileSearch tool object

Hide properties

type string Required

The type of tool being defined: `file_search`

metadata map Optional

Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

metadata map Optional

Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

tool_resources object or null Optional

A set of resources that are made available to the assistant's tools in this thread. The resources are specific to the type of tool. For example, the `code_interpreter` tool requires a list of file IDs, while the `file_search` tool requires a list of vector store IDs.

Hide properties

code_interpreter object Optional

Hide properties

file_ids array Optional Defaults to []

A list of file IDs made available to the `code_interpreter` tool. There can be a maximum of 20 files associated with the tool.

file_search Optional

Hide possible types

tool_choice string or object Optional

Controls which (if any) tool is called by the model. `none` means the model will not call any tools and instead generates a message. `auto` is the default value and means the model can pick between generating a message or calling one or more tools. `required` means the model must call one or more tools before responding to the user. Specifying a particular tool like `{"type": "file_search"}` or `{"type": "function", "function": {"name": "my_function"}}` forces the model to call that tool.

Hide possible types

string

`none` means the model will not call any tools and instead generates a message. `auto` means the model can pick between generating a message or calling one or more tools. `required` means the model must call one or more tools before responding to the user.

object

Specifies a tool the model should use. Use to force the model to call a specific tool.

Hide properties

type string Required

The type of the tool. If type is `function`, the function name must be set

function object Optional

Hide properties

name string Required

The name of the function to call.

tool_resources object or null Optional

A set of resources that are used by the assistant's tools. The resources are specific to the type of tool. For example, the `code_interpreter` tool requires a list of file IDs, while the `file_search` tool requires a list of vector store IDs.

Hide properties

code_interpreter object Optional

Hide properties

file_ids array Optional Defaults to []

A list of file IDs made available to the `code_interpreter` tool. There can be a maximum of 20 files associated with the tool.

file_search object Optional

Hide properties

vector_store_ids array Optional
The ID of the vector store attached to this assistant. There can be a maximum of 1 vector store attached to the assistant.

tools array or null Optional
Override the tools the assistant can use for this run. This is useful for modifying the behavior on a per-run basis.

Hide possible types

Code interpreter tool object

Hide properties

type string Required
The type of tool being defined: `code_interpreter`

FileSearch tool object

Hide properties

type string Required
The type of tool being defined: `file_search`

file_search object Optional
Overrides for the file search tool.

Hide properties

max_num_results integer Optional
The maximum number of results the file search tool should output. The default is 20 for `gpt-4*` models and 5 for `gpt-3.5-turbo`. This number should be between 1 and 50 inclusive.

Note that the file search tool may output fewer than `max_num_results` results. See the file search tool documentation for more information.

ranking_options object Optional
The ranking options for the file search. If not specified, the file search tool will use the `auto` ranker and a score_threshold of 0.

See the file search tool documentation for more information.

Hide properties

score_threshold number Required
The score threshold for the file search. All values must be a floating point number between 0

and 1.

ranker string Optional
The ranker to use for the file search. If not specified will use the `auto` ranker.

Function tool object

Hide properties

function object Required

Hide properties

name string Required
The name of the function to be called. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

description string Optional
A description of what the function does, used by the model to choose when and how to call the function.

parameters object Optional
The parameters the functions accepts, described as a JSON Schema object. See the guide for examples, and the JSON Schema reference for documentation about the format.

Omitting `parameters` defines a function with an empty parameter list.

strict boolean or null Optional Defaults to false
Whether to enable strict schema adherence when generating the function call. If set to true, the model will follow the exact schema defined in the `parameters` field. Only a subset of JSON Schema is supported when `strict` is `true`. Learn more about Structured Outputs in the function calling guide.

type string Required
The type of tool being defined: `function`

top_p number or null Optional Defaults to 1
An alternative to sampling with temperature, called nucleus sampling, where the model considers the results of the tokens with top_p probability mass. So 0.1 means only the tokens comprising the top 10% probability mass are considered.

We generally recommend altering this or temperature but not both.

truncation_strategy object or null Optional

Controls for how a thread will be truncated prior to the run. Use this to control the intial context window of the run.

Hide properties

type string Required

The truncation strategy to use for the thread. The default is `auto`. If set to `last_messages`, the thread will be truncated to the n most recent messages in the thread. When set to `auto`, messages in the middle of the thread will be dropped to fit the context length of the model, `max_prompt_tokens`.

last_messages integer or null Optional

The number of most recent messages from the thread when constructing the context for the run.

Returns

A run object.

Default Streaming Streaming with Functions

Example request curl

```
curl https://api.openai.com/v1/threads/runs \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -H "OpenAI-Beta: assistants=v2" \
  -d '{
      "assistant_id": "asst_abc123",
      "thread": {
        "messages": [
          {"role": "user", "content": "Explain deep learning to a 5 year old."}
        ]
      }
    }'
```

Response

```
{
  "id": "run_abc123",
  "object": "thread.run",
  "created_at": 1699076792,

```

```
  "assistant_id": "asst_abc123",
  "thread_id": "thread_abc123",
  "status": "queued",
  "started_at": null,
  "expires_at": 1699077392,
  "cancelled_at": null,
  "failed_at": null,
  "completed_at": null,
  "required_action": null,
  "last_error": null,
  "model": "gpt-4o",
  "instructions": "You are a helpful assistant.",
  "tools": [],
  "tool_resources": {},
  "metadata": {},
  "temperature": 1.0,
  "top_p": 1.0,
  "max_completion_tokens": null,
  "max_prompt_tokens": null,
  "truncation_strategy": {
    "type": "auto",
    "last_messages": null
  },
  "incomplete_details": null,
  "usage": null,
  "response_format": "auto",
  "tool_choice": "auto",
  "parallel_tool_calls": true
}
```

List runs Beta

```
GET https://api.openai.com/v1/threads/{thread_id}/runs
```

Returns a list of runs belonging to a thread.

Path parameters

thread_id string Required

The ID of the thread the run belongs to.

Query parameters

after string Optional
A cursor for use in pagination. `after` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, ending with obj_foo, your subsequent call can include after=obj_foo in order to fetch the next page of the list.

before string Optional
A cursor for use in pagination. `before` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, starting with obj_foo, your subsequent call can include before=obj_foo in order to fetch the previous page of the list.

limit integer Optional Defaults to 20
A limit on the number of objects to be returned. Limit can range between 1 and 100, and the default is 20.

order string Optional Defaults to desc
Sort order by the `created_at` timestamp of the objects. `asc` for ascending order and `desc` for descending order.

Returns

A list of run objects.

Example request curl

```
curl https://api.openai.com/v1/threads/thread_abc123/runs \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -H "OpenAI-Beta: assistants=v2"
```

Response

```
{
  "object": "list",
  "data": [
    {
      "id": "run_abc123",
      "object": "thread.run",
      "created_at": 1699075072,
      "assistant_id": "asst_abc123",
```

```
      "thread_id": "thread_abc123",
      "status": "completed",
      "started_at": 1699075072,
      "expires_at": null,
      "cancelled_at": null,
      "failed_at": null,
      "completed_at": 1699075073,
      "last_error": null,
      "model": "gpt-4o",
      "instructions": null,
      "incomplete_details": null,
      "tools": [
        {
          "type": "code_interpreter"
        }
      ],
      "tool_resources": {
        "code_interpreter": {
          "file_ids": [
            "file-abc123",
            "file-abc456"
          ]
        }
      },
      "metadata": {},
      "usage": {
        "prompt_tokens": 123,
        "completion_tokens": 456,
        "total_tokens": 579
      },
      "temperature": 1.0,
      "top_p": 1.0,
      "max_prompt_tokens": 1000,
      "max_completion_tokens": 1000,
      "truncation_strategy": {
        "type": "auto",
        "last_messages": null
      },
      "response_format": "auto",
      "tool_choice": "auto",
      "parallel_tool_calls": true
    },
    {
      "id": "run_abc456",
      "object": "thread.run",
      "created_at": 1699063290,
      "assistant_id": "asst_abc123",
```

```
      "thread_id": "thread_abc123",
      "status": "completed",
      "started_at": 1699063290,
      "expires_at": null,
      "cancelled_at": null,
      "failed_at": null,
      "completed_at": 1699063291,
      "last_error": null,
      "model": "gpt-4o",
      "instructions": null,
      "incomplete_details": null,
      "tools": [
        {
          "type": "code_interpreter"
        }
      ],
      "tool_resources": {
        "code_interpreter": {
          "file_ids": [
            "file-abc123",
            "file-abc456"
          ]
        }
      },
      "metadata": {},
      "usage": {
        "prompt_tokens": 123,
        "completion_tokens": 456,
        "total_tokens": 579
      },
      "temperature": 1.0,
      "top_p": 1.0,
      "max_prompt_tokens": 1000,
      "max_completion_tokens": 1000,
      "truncation_strategy": {
        "type": "auto",
        "last_messages": null
      },
      "response_format": "auto",
      "tool_choice": "auto",
      "parallel_tool_calls": true
    }
  ],
  "first_id": "run_abc123",
  "last_id": "run_abc456",
  "has_more": false
}
```

Retrieve run Beta

```
GET https://api.openai.com/v1/threads/{thread_id}/runs/{run_id}
```

Retrieves a run.

Path parameters

run_id string Required
The ID of the run to retrieve.

thread_id string Required
The ID of the thread that was run.

Returns

The run object matching the specified ID.

Example request curl

```curl
curl https://api.openai.com/v1/threads/thread_abc123/runs/run_abc123 \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "OpenAI-Beta: assistants=v2"
```

Response

```
{
  "id": "run_abc123",
  "object": "thread.run",
  "created_at": 1699075072,
  "assistant_id": "asst_abc123",
  "thread_id": "thread_abc123",
  "status": "completed",
  "started_at": 1699075072,
  "expires_at": null,
  "cancelled_at": null,
  "failed_at": null,
  "completed_at": 1699075073,

```

```
  "last_error": null,
  "model": "gpt-4o",
  "instructions": null,
  "incomplete_details": null,
  "tools": [
    {
      "type": "code_interpreter"
    }
  ],
  "metadata": {},
  "usage": {
    "prompt_tokens": 123,
    "completion_tokens": 456,
    "total_tokens": 579
  },
  "temperature": 1.0,
  "top_p": 1.0,
  "max_prompt_tokens": 1000,
  "max_completion_tokens": 1000,
  "truncation_strategy": {
    "type": "auto",
    "last_messages": null
  },
  "response_format": "auto",
  "tool_choice": "auto",
  "parallel_tool_calls": true
}
```

Modify run Beta

```
POST https://api.openai.com/v1/threads/{thread_id}/runs/{run_id}
```

Modifies a run.

Path parameters

run_id string Required
The ID of the run to modify.

thread_id string Required
The ID of the thread that was run.

Request body

metadata map Optional

Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

Returns

The modified run object matching the specified ID.

Example request curl

```
curl https://api.openai.com/v1/threads/thread_abc123/runs/run_abc123 \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -H "OpenAI-Beta: assistants=v2" \
  -d '{
    "metadata": {
      "user_id": "user_abc123"
    }
  }'
```

Response

```
{
  "id": "run_abc123",
  "object": "thread.run",
  "created_at": 1699075072,
  "assistant_id": "asst_abc123",
  "thread_id": "thread_abc123",
  "status": "completed",
  "started_at": 1699075072,
  "expires_at": null,
  "cancelled_at": null,
  "failed_at": null,
  "completed_at": 1699075073,
  "last_error": null,
  "model": "gpt-4o",
  "instructions": null,
  "incomplete_details": null,
```

```
  "tools": [
    {
      "type": "code_interpreter"
    }
  ],
  "tool_resources": {
    "code_interpreter": {
      "file_ids": [
        "file-abc123",
        "file-abc456"
      ]
    }
  },
  "metadata": {
    "user_id": "user_abc123"
  },
  "usage": {
    "prompt_tokens": 123,
    "completion_tokens": 456,
    "total_tokens": 579
  },
  "temperature": 1.0,
  "top_p": 1.0,
  "max_prompt_tokens": 1000,
  "max_completion_tokens": 1000,
  "truncation_strategy": {
    "type": "auto",
    "last_messages": null
  },
  "response_format": "auto",
  "tool_choice": "auto",
  "parallel_tool_calls": true
}
```

Submit tool outputs to run Beta

```
POST https://api.openai.com/v1/threads/{thread_id}/runs/{run_id}/submit_tool_outputs
```

When a run has the `status: "requires_action"` and `required_action.type` is `submit_tool_outputs`, this endpoint can be used to submit the outputs from the tool calls once they're all completed. All outputs must be submitted in a single request.

Path parameters

run_id string Required
The ID of the run that requires the tool output submission.

thread_id string Required
The ID of the thread to which this run belongs.

Request body

tool_outputs array Required
A list of tools for which the outputs are being submitted.

Hide properties

output string Optional
The output of the tool call to be submitted to continue the run.

tool_call_id string Optional
The ID of the tool call in the `required_action` object within the run object the output is being submitted for.

stream boolean or null Optional
If `true`, returns a stream of events that happen during the Run as server-sent events, terminating when the Run enters a terminal state with a `data: [DONE]` message.

Returns

The modified run object matching the specified ID.

Default Streaming

Example request curl

```curl
curl https://api.openai.com/v1/threads/thread_123/runs/run_123/submit_tool_outputs \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -H "OpenAI-Beta: assistants=v2" \
  -d '{
    "tool_outputs": [
```

```
      {
        "tool_call_id": "call_001",
        "output": "70 degrees and sunny."
      }
    ]
  }'
```

Response

```
{
  "id": "run_123",
  "object": "thread.run",
  "created_at": 1699075592,
  "assistant_id": "asst_123",
  "thread_id": "thread_123",
  "status": "queued",
  "started_at": 1699075592,
  "expires_at": 1699076192,
  "cancelled_at": null,
  "failed_at": null,
  "completed_at": null,
  "last_error": null,
  "model": "gpt-4o",
  "instructions": null,
  "tools": [
    {
      "type": "function",
      "function": {
        "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"
            },
            "unit": {
              "type": "string",
              "enum": ["celsius", "fahrenheit"]
            }
          },
          "required": ["location"]
        }
      }
    }
```

```
  ],
  "metadata": {},
  "usage": null,
  "temperature": 1.0,
  "top_p": 1.0,
  "max_prompt_tokens": 1000,
  "max_completion_tokens": 1000,
  "truncation_strategy": {
    "type": "auto",
    "last_messages": null
  },
  "response_format": "auto",
  "tool_choice": "auto",
  "parallel_tool_calls": true
}
```

Cancel a run Beta

`POST https://api.openai.com/v1/threads/{thread_id}/runs/{run_id}/cancel`

Cancels a run that is `in_progress`.

Path parameters

run_id string Required
The ID of the run to cancel.

thread_id string Required
The ID of the thread to which this run belongs.

Returns

The modified run object matching the specified ID.

Example request curl

```
curl https://api.openai.com/v1/threads/thread_abc123/runs/run_abc123/cancel \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "OpenAI-Beta: assistants=v2" \
```

```
4   -X POST
```

Response

```
1  {
2    "id": "run_abc123",
3    "object": "thread.run",
4    "created_at": 1699076126,
5    "assistant_id": "asst_abc123",
6    "thread_id": "thread_abc123",
7    "status": "cancelling",
8    "started_at": 1699076126,
9    "expires_at": 1699076726,
10   "cancelled_at": null,
11   "failed_at": null,
12   "completed_at": null,
13   "last_error": null,
14   "model": "gpt-4o",
15   "instructions": "You summarize books.",
16   "tools": [
17     {
18       "type": "file_search"
19     }
20   ],
21   "tool_resources": {
22     "file_search": {
23       "vector_store_ids": ["vs_123"]
24     }
25   },
26   "metadata": {},
27   "usage": null,
28   "temperature": 1.0,
29   "top_p": 1.0,
30   "response_format": "auto",
31   "tool_choice": "auto",
32   "parallel_tool_calls": true
33 }
```

The run object Beta

Represents an execution run on a thread.

assistant_id string
The ID of the assistant used for execution of this run.

cancelled_at integer or null
The Unix timestamp (in seconds) for when the run was cancelled.

completed_at integer or null
The Unix timestamp (in seconds) for when the run was completed.

created_at integer
The Unix timestamp (in seconds) for when the run was created.

expires_at integer or null
The Unix timestamp (in seconds) for when the run will expire.

failed_at integer or null
The Unix timestamp (in seconds) for when the run failed.

id string
The identifier, which can be referenced in API endpoints.

incomplete_details object or null
Details on why the run is incomplete. Will be `null` if the run is not incomplete.

^ Hide properties

reason string
The reason why the run is incomplete. This will point to which specific token limit was reached over the course of the run.

instructions string
The instructions that the assistant used for this run.

last_error object or null
The last error associated with this run. Will be `null` if there are no errors.

^ Hide properties

code string
One of `server_error`, `rate_limit_exceeded`, or `invalid_prompt`.

message string

A human-readable description of the error.

---

max_completion_tokens integer or null
The maximum number of completion tokens specified to have been used over the course of the run.

---

max_prompt_tokens integer or null
The maximum number of prompt tokens specified to have been used over the course of the run.

---

metadata map
Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

---

model string
The model that the assistant used for this run.

---

object string
The object type, which is always `thread.run`.

---

parallel_tool_calls boolean
Whether to enable parallel function calling during tool use.

---

required_action object or null
Details on the action required to continue the run. Will be `null` if no action is required.

Hide properties

submit_tool_outputs object
Details on the tool outputs needed for this run to continue.

Hide properties

tool_calls array
A list of the relevant tool calls.

Hide properties

function object
The function definition.

Hide properties

arguments string
The arguments that the model expects you to pass to the function.

name string
The name of the function.

id string
The ID of the tool call. This ID must be referenced when you submit the tool outputs in using the Submit tool outputs to run endpoint.

type string
The type of tool call the output is required for. For now, this is always `function`.

type string
For now, this is always `submit_tool_outputs`.

response_format "auto" or object
Specifies the format that the model must output. Compatible with GPT-4o, GPT-4 Turbo, and all GPT-3.5 Turbo models since `gpt-3.5-turbo-1106`.

Setting to `{ "type": "json_schema", "json_schema": {...} }` enables Structured Outputs which ensures the model will match your supplied JSON schema. Learn more in the Structured Outputs guide.

Setting to `{ "type": "json_object" }` enables JSON mode, which ensures the message the model generates is valid JSON.

Important: when using JSON mode, you **must** also instruct the model to produce JSON yourself via a system or user message. Without this, the model may generate an unending stream of whitespace until the generation reaches the token limit, resulting in a long-running and seemingly "stuck" request. Also note that the message content may be partially cut off if `finish_reason="length"`, which indicates the generation exceeded `max_tokens` or the conversation exceeded the max context length.

^ Hide possible types

string
`auto` is the default value

Text object
Default response format. Used to generate text responses.

^ Hide properties

type string
The type of response format being defined. Always `text`.

JSON object object
JSON object response format. An older method of generating JSON responses. Using `json_schema` is recommended for models that support it. Note that the model will not generate JSON without a system or user message instructing it to do so.

Hide properties

type string
The type of response format being defined. Always `json_object`.

JSON schema object
JSON Schema response format. Used to generate structured JSON responses. Learn more about Structured Outputs.

Hide properties

json_schema object
Structured Outputs configuration options, including a JSON Schema.

Hide properties

name string
The name of the response format. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

description string
A description of what the response format is for, used by the model to determine how to respond in the format.

schema object
The schema for the response format, described as a JSON Schema object. Learn how to build JSON schemas here.

strict boolean or null
Whether to enable strict schema adherence when generating the output. If set to true, the model will always follow the exact schema defined in the `schema` field. Only a subset of JSON Schema is supported when `strict` is `true`. To learn more, read the Structured Outputs guide.

type string
The type of response format being defined. Always `json_schema`.

started_at integer or null
The Unix timestamp (in seconds) for when the run was started.

status string
The status of the run, which can be either `queued`, `in_progress`, `requires_action`, `cancelling`, `cancelled`, `failed`, `completed`, `incomplete`, or `expired`.

temperature number or null
The sampling temperature used for this run. If not set, defaults to 1.

thread_id string
The ID of the thread that was executed on as a part of this run.

tool_choice string or object
Controls which (if any) tool is called by the model. `none` means the model will not call any tools and instead generates a message. `auto` is the default value and means the model can pick between generating a message or calling one or more tools. `required` means the model must call one or more tools before responding to the user. Specifying a particular tool like `{"type": "file_search"}` or `{"type": "function", "function": {"name": "my_function"}}` forces the model to call that tool.

Hide possible types

string
`none` means the model will not call any tools and instead generates a message. `auto` means the model can pick between generating a message or calling one or more tools. `required` means the model must call one or more tools before responding to the user.

object
Specifies a tool the model should use. Use to force the model to call a specific tool.

Hide properties

type string
The type of the tool. If type is `function`, the function name must be set

function object

Hide properties

name string
The name of the function to call.

tools array

The list of tools that the assistant used for this run.

Hide possible types

Code interpreter tool object

Hide properties

type string

The type of tool being defined: `code_interpreter`

FileSearch tool object

Hide properties

type string

The type of tool being defined: `file_search`

file_search object

Overrides for the file search tool.

Hide properties

max_num_results integer

The maximum number of results the file search tool should output. The default is 20 for `gpt-4*` models and 5 for `gpt-3.5-turbo`. This number should be between 1 and 50 inclusive.

Note that the file search tool may output fewer than `max_num_results` results. See the file search tool documentation for more information.

ranking_options object

The ranking options for the file search. If not specified, the file search tool will use the `auto` ranker and a score_threshold of 0.

See the file search tool documentation for more information.

Hide properties

score_threshold number

The score threshold for the file search. All values must be a floating point number between 0 and 1.

ranker string

The ranker to use for the file search. If not specified will use the `auto` ranker.

Function tool object

Hide properties

function object

Hide properties

name string
The name of the function to be called. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

description string
A description of what the function does, used by the model to choose when and how to call the function.

parameters object
The parameters the functions accepts, described as a JSON Schema object. See the guide for examples, and the JSON Schema reference for documentation about the format.

Omitting `parameters` defines a function with an empty parameter list.

strict boolean or null
Whether to enable strict schema adherence when generating the function call. If set to true, the model will follow the exact schema defined in the `parameters` field. Only a subset of JSON Schema is supported when `strict` is `true`. Learn more about Structured Outputs in the function calling guide.

type string
The type of tool being defined: `function`

top_p number or null
The nucleus sampling value used for this run. If not set, defaults to 1.

truncation_strategy object or null
Controls for how a thread will be truncated prior to the run. Use this to control the intial context window of the run.

Hide properties

type string
The truncation strategy to use for the thread. The default is `auto`. If set to `last_messages`, the thread will be truncated to the n most recent messages in the thread. When set to `auto`, messages in the middle of the thread will be dropped to fit the context length of the model, `max_prompt_tokens`.

last_messages integer or null
The number of most recent messages from the thread when constructing the context for the run.

usage object or null
Usage statistics related to the run. This value will be `null` if the run is not in a terminal state (i.e. `in_progress`, `queued`, etc.).

^ Hide properties

completion_tokens integer
Number of completion tokens used over the course of the run.

prompt_tokens integer
Number of prompt tokens used over the course of the run.

total_tokens integer
Total number of tokens used (prompt + completion).

OBJECT The run object

```
{
  "id": "run_abc123",
  "object": "thread.run",
  "created_at": 1698107661,
  "assistant_id": "asst_abc123",
  "thread_id": "thread_abc123",
  "status": "completed",
  "started_at": 1699073476,
  "expires_at": null,
  "cancelled_at": null,
  "failed_at": null,
  "completed_at": 1699073498,
  "last_error": null,
  "model": "gpt-4o",
  "instructions": null,
  "tools": [{"type": "file_search"}, {"type": "code_interpreter"}],
  "metadata": {},
  "incomplete_details": null,

```

```
  "usage": {
    "prompt_tokens": 123,
    "completion_tokens": 456,
    "total_tokens": 579
  },
  "temperature": 1.0,
  "top_p": 1.0,
  "max_prompt_tokens": 1000,
  "max_completion_tokens": 1000,
  "truncation_strategy": {
    "type": "auto",
    "last_messages": null
  },
  "response_format": "auto",
  "tool_choice": "auto",
  "parallel_tool_calls": true
}
```

Run steps Beta

Represents the steps (model and tool calls) taken during the run.

Related guide: Assistants

List run steps Beta

```
GET https://api.openai.com/v1/threads/{thread_id}/runs/{run_id}/steps
```

Returns a list of run steps belonging to a run.

Path parameters

run_id string Required
The ID of the run the run steps belong to.

thread_id string Required
The ID of the thread the run and run steps belong to.

Query parameters

after string Optional
A cursor for use in pagination. `after` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, ending with obj_foo, your subsequent call can include after=obj_foo in order to fetch the next page of the list.

before string Optional
A cursor for use in pagination. `before` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, starting with obj_foo, your subsequent call can include before=obj_foo in order to fetch the previous page of the list.

include[] array Optional
A list of additional fields to include in the response. Currently the only supported value is `step_details.tool_calls[*].file_search.results[*].content` to fetch the file search result content.

See the file search tool documentation for more information.

limit integer Optional Defaults to 20
A limit on the number of objects to be returned. Limit can range between 1 and 100, and the default is 20.

order string Optional Defaults to desc
Sort order by the `created_at` timestamp of the objects. `asc` for ascending order and `desc` for descending order.

Returns

A list of run step objects.

Example request curl

```curl
curl https://api.openai.com/v1/threads/thread_abc123/runs/run_abc123/steps \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -H "OpenAI-Beta: assistants=v2"
```

Response

```
{
  "object": "list",
```

```
  "data": [
    {
      "id": "step_abc123",
      "object": "thread.run.step",
      "created_at": 1699063291,
      "run_id": "run_abc123",
      "assistant_id": "asst_abc123",
      "thread_id": "thread_abc123",
      "type": "message_creation",
      "status": "completed",
      "cancelled_at": null,
      "completed_at": 1699063291,
      "expired_at": null,
      "failed_at": null,
      "last_error": null,
      "step_details": {
        "type": "message_creation",
        "message_creation": {
          "message_id": "msg_abc123"
        }
      },
      "usage": {
        "prompt_tokens": 123,
        "completion_tokens": 456,
        "total_tokens": 579
      }
    }
  ],
  "first_id": "step_abc123",
  "last_id": "step_abc456",
  "has_more": false
}
```

Retrieve run step Beta

```
GET https://api.openai.com/v1/threads/{thread_id}/runs/{run_id}/steps/{step_id}
```

Retrieves a run step.

Path parameters

run_id string Required

The ID of the run to which the run step belongs.

step_id string Required
The ID of the run step to retrieve.

thread_id string Required
The ID of the thread to which the run and run step belongs.

Query parameters

include[] array Optional
A list of additional fields to include in the response. Currently the only supported value is `step_details.tool_calls[*].file_search.results[*].content` to fetch the file search result content.

See the file search tool documentation for more information.

Returns

The run step object matching the specified ID.

Example request curl

```curl
curl https://api.openai.com/v1/threads/thread_abc123/runs/run_abc123/steps/step_abc123 \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -H "OpenAI-Beta: assistants=v2"
```

Response

```
{
  "id": "step_abc123",
  "object": "thread.run.step",
  "created_at": 1699063291,
  "run_id": "run_abc123",
  "assistant_id": "asst_abc123",
  "thread_id": "thread_abc123",
  "type": "message_creation",
  "status": "completed",
  "cancelled_at": null,
  "completed_at": 1699063291,

```

```
  "expired_at": null,
  "failed_at": null,
  "last_error": null,
  "step_details": {
    "type": "message_creation",
    "message_creation": {
      "message_id": "msg_abc123"
    }
  },
  "usage": {
    "prompt_tokens": 123,
    "completion_tokens": 456,
    "total_tokens": 579
  }
}
```

The run step object Beta

Represents a step in execution of a run.

`assistant_id` string
The ID of the assistant associated with the run step.

`cancelled_at` integer or null
The Unix timestamp (in seconds) for when the run step was cancelled.

`completed_at` integer or null
The Unix timestamp (in seconds) for when the run step completed.

`created_at` integer
The Unix timestamp (in seconds) for when the run step was created.

`expired_at` integer or null
The Unix timestamp (in seconds) for when the run step expired. A step is considered expired if the parent run is expired.

`failed_at` integer or null
The Unix timestamp (in seconds) for when the run step failed.

`id` string

The identifier of the run step, which can be referenced in API endpoints.

`last_error` object or null

The last error associated with this run step. Will be `null` if there are no errors.

Hide properties

`code` string

One of `server_error` or `rate_limit_exceeded`.

`message` string

A human-readable description of the error.

`metadata` map

Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

`object` string

The object type, which is always `thread.run.step`.

`run_id` string

The ID of the run that this run step is a part of.

`status` string

The status of the run step, which can be either `in_progress`, `cancelled`, `failed`, `completed`, or `expired`.

`step_details` object

The details of the run step.

Hide possible types

Message creation object

Details of the message creation by the run step.

Hide properties

`message_creation` object

Hide properties

message_id string
The ID of the message that was created by this run step.

type string
Always `message_creation`.

Tool calls object
Details of the tool call.

Hide properties

tool_calls array
An array of tool calls the run step was involved in. These can be associated with one of three types of tools: `code_interpreter`, `file_search`, or `function`.

Hide possible types

Code Interpreter tool call object
Details of the Code Interpreter tool call the run step was involved in.

Hide properties

code_interpreter object
The Code Interpreter tool call definition.

Hide properties

input string
The input to the Code Interpreter tool call.

outputs array
The outputs from the Code Interpreter tool call. Code Interpreter can output one or more items, including text (`logs`) or images (`image`). Each of these are represented by a different object type.

Hide possible types

Code Interpreter log output object

Text output from the Code Interpreter tool call as part of a run step.

Hide properties

logs string

The text output from the Code Interpreter tool call.

type string

Always `logs`.

Code Interpreter image output object

Hide properties

image object

Hide properties

file_id string

The file ID of the image.

type string

Always `image`.

id string

The ID of the tool call.

type string

The type of tool call. This is always going to be `code_interpreter` for this type of tool call.

File search tool call object

Hide properties

file_search map

For now, this is always going to be an empty object.

Hide properties

ranking_options object

The ranking options for the file search.

Hide properties

ranker string
The ranker to use for the file search. If not specified will use the `auto` ranker.

score_threshold number
The score threshold for the file search. All values must be a floating point number between 0 and 1.

results array
The results of the file search.

Hide properties

file_id string
The ID of the file that result was found in.

file_name string
The name of the file that result was found in.

score number
The score of the result. All values must be a floating point number between 0 and 1.

content array
The content of the result that was found. The content is only included if requested via the include query parameter.

Hide properties

text string
The text content of the file.

type string
The type of the content.

id string
The ID of the tool call object.

type string
The type of tool call. This is always going to be `file_search` for this type of tool call.

Function tool call object

Hide properties

function object
The definition of the function that was called.

^ Hide properties

arguments string
The arguments passed to the function.

name string
The name of the function.

output string or null
The output of the function. This will be `null` if the outputs have not been submitted yet.

id string
The ID of the tool call object.

type string
The type of tool call. This is always going to be `function` for this type of tool call.

type string
Always `tool_calls`.

thread_id string
The ID of the thread that was run.

type string
The type of run step, which can be either `message_creation` or `tool_calls`.

usage object or null
Usage statistics related to the run step. This value will be `null` while the run step's status is `in_progress`.

^ Hide properties

completion_tokens integer
Number of completion tokens used over the course of the run step.

prompt_tokens integer
Number of prompt tokens used over the course of the run step.

total_tokens integer
Total number of tokens used (prompt + completion).

OBJECT The run step object

```
{
  "id": "step_abc123",
  "object": "thread.run.step",
  "created_at": 1699063291,
  "run_id": "run_abc123",
  "assistant_id": "asst_abc123",
  "thread_id": "thread_abc123",
  "type": "message_creation",
  "status": "completed",
  "cancelled_at": null,
  "completed_at": 1699063291,
  "expired_at": null,
  "failed_at": null,
  "last_error": null,
  "step_details": {
    "type": "message_creation",
    "message_creation": {
      "message_id": "msg_abc123"
    }
  },
  "usage": {
    "prompt_tokens": 123,
    "completion_tokens": 456,
    "total_tokens": 579
  }
}
```

Streaming Beta

Stream the result of executing a Run or resuming a Run after submitting tool outputs. You can stream events from the Create Thread and Run, Create Run, and Submit Tool Outputs endpoints by passing `"stream": true`. The response will be a Server-Sent events stream. Our Node and Python SDKs provide helpful utilities to make streaming easy. Reference the Assistants API quickstart to learn more.

The message delta object Beta

Represents a message delta i.e. any changed fields on a message during streaming.

delta object
The delta containing the fields that have changed on the Message.

Hide properties

content array
The content of the message in array of text and/or images.

Hide possible types

Image file object
References an image File in the content of a message.

Hide properties

index integer
The index of the content part in the message.

type string
Always `image_file`.

image_file object

Hide properties

detail string
Specifies the detail level of the image if specified by the user. `low` uses fewer tokens, you can opt in to high resolution using `high`.

file_id string
The File ID of the image in the message content. Set `purpose="vision"` when uploading the File if you need to later display the file content.

Text object
The text content that is part of a message.

Hide properties

index integer
The index of the content part in the message.

type string

Always `text`.

text object

Hide properties

annotations array

Hide possible types

File citation object

A citation within the message that points to a specific quote from a specific File associated with the assistant or the message. Generated when the assistant uses the "file_search" tool to search files.

Hide properties

index integer

The index of the annotation in the text content part.

type string

Always `file_citation`.

end_index integer

file_citation object

Hide properties

file_id string

The ID of the specific File the citation is from.

quote string

The specific quote in the file.

start_index integer

text string

The text in the message content that needs to be replaced.

File path object

A URL for the file that's generated when the assistant used the `code_interpreter` tool to generate a file.

Hide properties

index integer

The index of the annotation in the text content part.

type string
Always `file_path`.

end_index integer

file_path object
Hide properties

file_id string
The ID of the file that was generated.

start_index integer

text string
The text in the message content that needs to be replaced.

value string
The data that makes up the text.

Refusal object
The refusal content that is part of a message.
Hide properties

index integer
The index of the refusal part in the message.

type string
Always `refusal`.

refusal string

Image URL object
References an image URL in the content of a message.
Hide properties

index integer
The index of the content part in the message.

type string
Always `image_url`.

image_url object

Hide properties

detail string
Specifies the detail level of the image. `low` uses fewer tokens, you can opt in to high resolution using `high`.

url string
The URL of the image, must be a supported image types: jpeg, jpg, png, gif, webp.

role string
The entity that produced the message. One of `user` or `assistant`.

id string
The identifier of the message, which can be referenced in API endpoints.

object string
The object type, which is always `thread.message.delta`.

OBJECT The message delta object

```
{
  "id": "msg_123",
  "object": "thread.message.delta",
  "delta": {
    "content": [
      {
        "index": 0,
        "type": "text",
        "text": { "value": "Hello", "annotations": [] }
      }
    ]
  }
}
```

The run step delta object Beta

Represents a run step delta i.e. any changed fields on a run step during streaming.

delta object
The delta containing the fields that have changed on the run step.

Hide properties

step_details object
The details of the run step.

Hide possible types

Message creation object
Details of the message creation by the run step.

Hide properties

type string
Always `message_creation`.

message_creation object

Hide properties

message_id string
The ID of the message that was created by this run step.

Tool calls object
Details of the tool call.

Hide properties

type string
Always `tool_calls`.

tool_calls array
An array of tool calls the run step was involved in. These can be associated with one of three types of tools: `code_interpreter`, `file_search`, or `function`.

Hide possible types

Code interpreter tool call object
Details of the Code Interpreter tool call the run step was involved in.

Hide properties

index integer
The index of the tool call in the tool calls array.

type string
The type of tool call. This is always going to be `code_interpreter` for this type of tool call.

code_interpreter object
The Code Interpreter tool call definition.

Hide properties

input string
The input to the Code Interpreter tool call.

outputs array
The outputs from the Code Interpreter tool call. Code Interpreter can output one or more items, including text (`logs`) or images (`image`). Each of these are represented by a different object type.

Hide possible types

Code interpreter log output object
Text output from the Code Interpreter tool call as part of a run step.

Hide properties

index integer
The index of the output in the outputs array.

type string
Always `logs` .

logs string
The text output from the Code Interpreter tool call.

Code interpreter image output object

Hide properties

index integer
The index of the output in the outputs array.

type string
Always `image` .

image object

Hide properties

file_id string
The file ID of the image.

id string
The ID of the tool call.

File search tool call object

Hide properties

file_search map
For now, this is always going to be an empty object.

index integer
The index of the tool call in the tool calls array.

type string
The type of tool call. This is always going to be `file_search` for this type of tool call.

id string
The ID of the tool call object.

Function tool call object

Hide properties

index integer
The index of the tool call in the tool calls array.

type string
The type of tool call. This is always going to be `function` for this type of tool call.

function object
The definition of the function that was called.

Hide properties

arguments string
The arguments passed to the function.

name string
The name of the function.

output string or null
The output of the function. This will be `null` if the outputs have not been submitted yet.

id string
The ID of the tool call object.

id string
The identifier of the run step, which can be referenced in API endpoints.

object string
The object type, which is always `thread.run.step.delta`.

OBJECT The run step delta object

```
{
  "id": "step_123",
  "object": "thread.run.step.delta",
  "delta": {
    "step_details": {
      "type": "tool_calls",
      "tool_calls": [
        {
          "index": 0,
          "id": "call_123",
          "type": "code_interpreter",
          "code_interpreter": { "input": "", "outputs": [] }
        }
      ]
    }
  }
}
```

Assistant stream events Beta

Represents an event emitted when streaming a Run.

Each event in a server-sent events stream has an `event` and `data` property:

```
event: thread.created
data: {"id": "thread_123", "object": "thread", ...}
```

We emit events whenever a new object is created, transitions to a new state, or is being streamed in parts (deltas). For example, we emit `thread.run.created` when a new run is created, `thread.run.completed` when a run completes, and so on. When an Assistant chooses to create a message during a run, we emit a `thread.message.created event`, a `thread.message.in_progress` event, many `thread.message.delta` events, and finally a `thread.message.completed` event.

We may add additional events over time, so we recommend handling unknown events gracefully in your code. See the Assistants API quickstart to learn how to integrate the Assistants API with streaming.

---

done `data` is `[DONE]`

Occurs when a stream ends.

---

error `data` is an error

Occurs when an error occurs. This can happen due to an internal server error or a timeout.

---

thread.created `data` is a thread

Occurs when a new thread is created.

---

thread.message.completed `data` is a message

Occurs when a message is completed.

---

thread.message.created `data` is a message

Occurs when a message is created.

---

thread.message.delta `data` is a message delta

Occurs when parts of a Message are being streamed.

---

thread.message.in_progress `data` is a message

Occurs when a message moves to an `in_progress` state.

`thread.message.incomplete` `data` is a message

Occurs when a message ends before it is completed.

`thread.run.cancelled` `data` is a run

Occurs when a run is cancelled.

`thread.run.cancelling` `data` is a run

Occurs when a run moves to a `cancelling` status.

`thread.run.completed` `data` is a run

Occurs when a run is completed.

`thread.run.created` `data` is a run

Occurs when a new run is created.

`thread.run.expired` `data` is a run

Occurs when a run expires.

`thread.run.failed` `data` is a run

Occurs when a run fails.

`thread.run.in_progress` `data` is a run

Occurs when a run moves to an `in_progress` status.

`thread.run.incomplete` `data` is a run

Occurs when a run ends with status `incomplete`.

`thread.run.queued` `data` is a run

Occurs when a run moves to a `queued` status.

`thread.run.requires_action` `data` is a run

Occurs when a run moves to a `requires_action` status.

`thread.run.step.cancelled` `data` is a run step

Occurs when a run step is cancelled.

`thread.run.step.completed` `data` is a run step

Occurs when a run step is completed.

`thread.run.step.created` `data` is a run step

Occurs when a run step is created.

`thread.run.step.delta` `data` is a run step delta

Occurs when parts of a run step are being streamed.

`thread.run.step.expired` `data` is a run step

Occurs when a run step expires.

`thread.run.step.failed` `data` is a run step

Occurs when a run step fails.

`thread.run.step.in_progress` `data` is a run step

Occurs when a run step moves to an `in_progress` state.

Administration

Programmatically manage your organization. The Audit Logs endpoint provides a log of all actions taken in the organization for security and monitoring purposes. To access these endpoints please generate an Admin API Key through the API Platform Organization overview. Admin API keys cannot be used for non-administration endpoints. For best practices on setting up your organization, please refer to this guide

Admin API Keys

Admin API keys enable Organization Owners to programmatically manage various aspects of their organization, including users, projects, and API keys. These keys provide administrative capabilities, such as creating, updating, and deleting users; managing projects; and

overseeing API key lifecycles.

Key Features of Admin API Keys:

- User Management: Invite new users, update roles, and remove users from the organization.
- Project Management: Create, update, archive projects, and manage user assignments within projects.
- API Key Oversight: List, retrieve, and delete API keys associated with projects.

Only Organization Owners have the authority to create and utilize Admin API keys. To manage these keys, Organization Owners can navigate to the Admin Keys section of their API Platform dashboard.

For direct access to the Admin Keys management page, Organization Owners can use the following link:

https://platform.openai.com/settings/organization/admin-keys

It's crucial to handle Admin API keys with care due to their elevated permissions. Adhering to best practices, such as regular key rotation and assigning appropriate permissions, enhances security and ensures proper governance within the organization.

List all organization and project API keys.

```
GET https://api.openai.com/v1/organization/admin_api_keys
```

List organization API keys

Query parameters

after string or null Optional

limit integer Optional Defaults to 20

order string Optional Defaults to asc

Returns

A list of admin and project API key objects.

Example request curl

```
curl https://api.openai.com/v1/organization/admin_api_keys?after=key_abc&limit=20 \
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
  "object": "list",
  "data": [
    {
      "object": "organization.admin_api_key",
      "id": "key_abc",
      "name": "Main Admin Key",
      "redacted_value": "sk-admin...def",
      "created_at": 1711471533,
      "last_used_at": 1711471534,
      "owner": {
        "type": "service_account",
        "object": "organization.service_account",
        "id": "sa_456",
        "name": "My Service Account",
        "created_at": 1711471533,
        "role": "member"
      }
    }
  ],
  "first_id": "key_abc",
  "last_id": "key_abc",
  "has_more": false
}
```

Create admin API key

```
POST https://api.openai.com/v1/organization/admin_api_keys
```

Create an organization admin API key

Request body

name string Required

Returns

The created AdminApiKey object.

Example request curl

```
curl -X POST https://api.openai.com/v1/organization/admin_api_keys \
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json" \
  -d '{
      "name": "New Admin Key"
  }'
```

Response

```
{
  "object": "organization.admin_api_key",
  "id": "key_xyz",
  "name": "New Admin Key",
  "redacted_value": "sk-admin...xyz",
  "created_at": 1711471533,
  "last_used_at": 1711471534,
  "owner": {
    "type": "user",
    "object": "organization.user",
    "id": "user_123",
    "name": "John Doe",
    "created_at": 1711471533,
    "role": "owner"
  },
  "value": "sk-admin-1234abcd"
}
```

Retrieve admin API key

```
GET https://api.openai.com/v1/organization/admin_api_keys/{key_id}
```

Retrieve a single organization API key

Path parameters

key_id string Required

Returns

The requested AdminApiKey object.

Example request curl

```
curl https://api.openai.com/v1/organization/admin_api_keys/key_abc \
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
  "object": "organization.admin_api_key",
  "id": "key_abc",
  "name": "Main Admin Key",
  "redacted_value": "sk-admin...xyz",
  "created_at": 1711471533,
  "last_used_at": 1711471534,
  "owner": {
    "type": "user",
    "object": "organization.user",
    "id": "user_123",
    "name": "John Doe",
    "created_at": 1711471533,
    "role": "owner"
  }
}
```

Delete admin API key

```
DELETE https://api.openai.com/v1/organization/admin_api_keys/{key_id}
```

Delete an organization admin API key

Path parameters

key_id string Required

Returns

A confirmation object indicating the key was deleted.

Example request curl

```curl
curl -X DELETE https://api.openai.com/v1/organization/admin_api_keys/key_abc \
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json"
```

Response

```json
{
  "id": "key_abc",
  "object": "organization.admin_api_key.deleted",
  "deleted": true
}
```

The admin API key object

Represents an individual Admin API key in an org.

created_at integer
The Unix timestamp (in seconds) of when the API key was created

id string
The identifier, which can be referenced in API endpoints

last_used_at integer or null
The Unix timestamp (in seconds) of when the API key was last used

name string
The name of the API key

object string
The object type, which is always `organization.admin_api_key`

owner object

Hide properties

created_at integer
The Unix timestamp (in seconds) of when the user was created

id string
The identifier, which can be referenced in API endpoints

name string
The name of the user

object string
The object type, which is always organization.user

role string
Always `owner`

type string
Always `user`

redacted_value string
The redacted value of the API key

value string
The value of the API key. Only shown on create.

```
OBJECT The admin API key object
```

```
{
  "object": "organization.admin_api_key",
  "id": "key_abc",
  "name": "Main Admin Key",
  "redacted_value": "sk-admin...xyz",
  "created_at": 1711471533,
  "last_used_at": 1711471534,
  "owner": {
    "type": "user",
    "object": "organization.user",
    "id": "user_123",
    "name": "John Doe",
    "created_at": 1711471533,
    "role": "owner"
  }
}
```

Invites

Invite and manage invitations for an organization.

List invites

`GET https://api.openai.com/v1/organization/invites`

Returns a list of invites in the organization.

Query parameters

`after` string Optional

A cursor for use in pagination. `after` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, ending with obj_foo, your subsequent call can include after=obj_foo in order to fetch the next page of the list.

`limit` integer Optional Defaults to 20

A limit on the number of objects to be returned. Limit can range between 1 and 100, and the default is 20.

Returns

A list of Invite objects.

Example request curl

```
curl https://api.openai.com/v1/organization/invites?after=invite-abc&limit=20 \
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
  "object": "list",
  "data": [
    {
      "object": "organization.invite",
      "id": "invite-abc",
      "email": "user@example.com",
      "role": "owner",
      "status": "accepted",
      "invited_at": 1711471533,
      "expires_at": 1711471533,
      "accepted_at": 1711471533
    }
  ],
  "first_id": "invite-abc",
  "last_id": "invite-abc",
  "has_more": false
}
```

Create invite

`POST https://api.openai.com/v1/organization/invites`

Create an invite for a user to the organization. The invite must be accepted by the user before they have access to the organization.

Request body

email string Required
Send an email to this address

role string Required
`owner` or `reader`

projects array Optional
An array of projects to which membership is granted at the same time the org invite is accepted. If omitted, the user will be invited to the default project for compatibility with legacy behavior.

Hide properties

id string Required
Project's public ID

role string Required
Project membership role

Returns

The created Invite object.

Example request curl

```
curl -X POST https://api.openai.com/v1/organization/invites \
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json" \
  -d '{
      "email": "anotheruser@example.com",
      "role": "reader",
      "projects": [
        {
          "id": "project-xyz",
          "role": "member"
        },
        {
          "id": "project-abc",
          "role": "owner"
        }
      ]
  }'
```

Response

```
{
  "object": "organization.invite",
  "id": "invite-def",
  "email": "anotheruser@example.com",
  "role": "reader",
  "status": "pending",
  "invited_at": 1711471533,
  "expires_at": 1711471533,
  "accepted_at": null,
  "projects": [
    {
      "id": "project-xyz",
      "role": "member"
    },
    {
      "id": "project-abc",
      "role": "owner"
    }
  ]
}
```

Retrieve invite

`GET https://api.openai.com/v1/organization/invites/{invite_id}`

Retrieves an invite.

Path parameters

invite_id string Required
The ID of the invite to retrieve.

Returns

The Invite object matching the specified ID.

Example request curl

```
curl https://api.openai.com/v1/organization/invites/invite-abc \
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
    "object": "organization.invite",
    "id": "invite-abc",
    "email": "user@example.com",
    "role": "owner",
    "status": "accepted",
    "invited_at": 1711471533,
    "expires_at": 1711471533,
    "accepted_at": 1711471533
}
```

Delete invite

`DELETE https://api.openai.com/v1/organization/invites/{invite_id}`

Delete an invite. If the invite has already been accepted, it cannot be deleted.

Path parameters

invite_id string Required
The ID of the invite to delete.

Returns

Confirmation that the invite has been deleted

Example request curl

```
curl -X DELETE https://api.openai.com/v1/organization/invites/invite-abc \
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
    "object": "organization.invite.deleted",
    "id": "invite-abc",
    "deleted": true
}
```

The invite object

Represents an individual `invite` to the organization.

`accepted_at` integer
The Unix timestamp (in seconds) of when the invite was accepted.

`email` string
The email address of the individual to whom the invite was sent

`expires_at` integer
The Unix timestamp (in seconds) of when the invite expires.

`id` string
The identifier, which can be referenced in API endpoints

`invited_at` integer
The Unix timestamp (in seconds) of when the invite was sent.

`object` string
The object type, which is always `organization.invite`

projects array

The projects that were granted membership upon acceptance of the invite.

Hide properties

id string

Project's public ID

role string

Project membership role

role string

`owner` or `reader`

status string

`accepted`, `expired`, or `pending`

OBJECT The invite object

```
{
  "object": "organization.invite",
  "id": "invite-abc",
  "email": "user@example.com",
  "role": "owner",
  "status": "accepted",
  "invited_at": 1711471533,
  "expires_at": 1711471533,
  "accepted_at": 1711471533,
  "projects": [
    {
      "id": "project-xyz",
      "role": "member"
    }
  ]
}
```

Users

Manage users and their role in an organization.

List users

`GET https://api.openai.com/v1/organization/users`

Lists all of the users in the organization.

Query parameters

after string Optional
A cursor for use in pagination. `after` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, ending with obj_foo, your subsequent call can include after=obj_foo in order to fetch the next page of the list.

emails array Optional
Filter by the email address of users.

limit integer Optional Defaults to 20
A limit on the number of objects to be returned. Limit can range between 1 and 100, and the default is 20.

Returns

A list of User objects.

Example request curl

```curl
curl https://api.openai.com/v1/organization/users?after=user_abc&limit=20 \
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
    "object": "list",
    "data": [
        {
            "object": "organization.user",
            "id": "user_abc",
            "name": "First Last",
            "email": "user@example.com",

```

```
            "role": "owner",
            "added_at": 1711471533
        }
    ],
    "first_id": "user-abc",
    "last_id": "user-xyz",
    "has_more": false
}
```

Modify user

`POST https://api.openai.com/v1/organization/users/{user_id}`

Modifies a user's role in the organization.

Path parameters

user_id string Required
The ID of the user.

Request body

role string Required
`owner` or `reader`

Returns

The updated User object.

Example request curl

```
curl -X POST https://api.openai.com/v1/organization/users/user_abc \
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json" \
  -d '{
      "role": "owner"
  }'
```

Response

```
{
    "object": "organization.user",
    "id": "user_abc",
    "name": "First Last",
    "email": "user@example.com",
    "role": "owner",
    "added_at": 1711471533
}
```

Retrieve user

`GET https://api.openai.com/v1/organization/users/{user_id}`

Retrieves a user by their identifier.

Path parameters

user_id string Required
The ID of the user.

Returns

The User object matching the specified ID.

Example request curl

```
curl https://api.openai.com/v1/organization/users/user_abc \
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
    "object": "organization.user",
    "id": "user_abc",
    "name": "First Last",
```

```
    "email": "user@example.com",
    "role": "owner",
    "added_at": 1711471533
}
```

Delete user

```
DELETE https://api.openai.com/v1/organization/users/{user_id}
```

Deletes a user from the organization.

Path parameters

user_id string Required
The ID of the user.

Returns

Confirmation of the deleted user

Example request curl

```
curl -X DELETE https://api.openai.com/v1/organization/users/user_abc \
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
    "object": "organization.user.deleted",
    "id": "user_abc",
    "deleted": true
}
```

The user object

Represents an individual `user` within an organization.

`added_at` integer
The Unix timestamp (in seconds) of when the user was added.

`email` string
The email address of the user

`id` string
The identifier, which can be referenced in API endpoints

`name` string
The name of the user

`object` string
The object type, which is always `organization.user`

`role` string
`owner` or `reader`

OBJECT The user object

```
{
    "object": "organization.user",
    "id": "user_abc",
    "name": "First Last",
    "email": "user@example.com",
    "role": "owner",
    "added_at": 1711471533
}
```

Projects

Manage the projects within an orgnanization includes creation, updating, and archiving or projects. The Default project cannot be archived.

List projects

```
GET https://api.openai.com/v1/organization/projects
```

Returns a list of projects.

Query parameters

after string Optional
A cursor for use in pagination. `after` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, ending with obj_foo, your subsequent call can include after=obj_foo in order to fetch the next page of the list.

include_archived boolean Optional Defaults to false
If `true` returns all projects including those that have been `archived`. Archived projects are not included by default.

limit integer Optional Defaults to 20
A limit on the number of objects to be returned. Limit can range between 1 and 100, and the default is 20.

Returns

A list of Project objects.

Example request curl

```curl
curl https://api.openai.com/v1/organization/projects?after=proj_abc&limit=20&include_archiv
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
    "object": "list",
    "data": [
        {
            "id": "proj_abc",
            "object": "organization.project",
            "name": "Project example",
            "created_at": 1711471533,
            "archived_at": null,
            "status": "active"
```

```
        }
    ],
    "first_id": "proj-abc",
    "last_id": "proj-xyz",
    "has_more": false
}
```

Create project

`POST https://api.openai.com/v1/organization/projects`

Create a new project in the organization. Projects can be created and archived, but cannot be deleted.

Request body

name string Required
The friendly name of the project, this name appears in reports.

Returns

The created Project object.

Example request curl

```
curl -X POST https://api.openai.com/v1/organization/projects \
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json" \
  -d '{
      "name": "Project ABC"
  }'
```

Response

```
{
    "id": "proj_abc",
    "object": "organization.project",
    "name": "Project ABC",
```

```
    "created_at": 1711471533,
    "archived_at": null,
    "status": "active"
}
```

Retrieve project

```
GET https://api.openai.com/v1/organization/projects/{project_id}
```

Retrieves a project.

Path parameters

project_id string Required
The ID of the project.

Returns

The Project object matching the specified ID.

Example request curl

```
curl https://api.openai.com/v1/organization/projects/proj_abc \
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
    "id": "proj_abc",
    "object": "organization.project",
    "name": "Project example",
    "created_at": 1711471533,
    "archived_at": null,
    "status": "active"
}
```

Modify project

`POST https://api.openai.com/v1/organization/projects/{project_id}`

Modifies a project in the organization.

Path parameters

project_id string Required
The ID of the project.

Request body

name string Required
The updated name of the project, this name appears in reports.

Returns

The updated Project object.

Example request curl

```curl
curl -X POST https://api.openai.com/v1/organization/projects/proj_abc \
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json" \
  -d '{
      "name": "Project DEF"
  }'
```

Archive project

`POST https://api.openai.com/v1/organization/projects/{project_id}/archive`

Archives a project in the organization. Archived projects cannot be used or updated.

Path parameters

project_id string Required
The ID of the project.

Returns

The archived Project object.

Example request curl

```
curl -X POST https://api.openai.com/v1/organization/projects/proj_abc/archive \
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
    "id": "proj_abc",
    "object": "organization.project",
    "name": "Project DEF",
    "created_at": 1711471533,
    "archived_at": 1711471533,
    "status": "archived"
}
```

The project object

Represents an individual project.

archived_at integer or null
The Unix timestamp (in seconds) of when the project was archived or `null`.

created_at integer
The Unix timestamp (in seconds) of when the project was created.

id string
The identifier, which can be referenced in API endpoints

name string
The name of the project. This appears in reporting.

object string
The object type, which is always `organization.project`

status string
`active` or `archived`

OBJECT The project object

```
{
    "id": "proj_abc",
    "object": "organization.project",
    "name": "Project example",
    "created_at": 1711471533,
    "archived_at": null,
    "status": "active"
}
```

Project users

Manage users within a project, including adding, updating roles, and removing users.

List project users

```
GET https://api.openai.com/v1/organization/projects/{project_id}/users
```

Returns a list of users in the project.

Path parameters

project_id string Required
The ID of the project.

Query parameters

after string Optional

A cursor for use in pagination. `after` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, ending with obj_foo, your subsequent call can include after=obj_foo in order to fetch the next page of the list.

limit integer Optional Defaults to 20

A limit on the number of objects to be returned. Limit can range between 1 and 100, and the default is 20.

Returns

A list of ProjectUser objects.

Example request curl

```
curl https://api.openai.com/v1/organization/projects/proj_abc/users?after=user_abc&limit=20
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
    "object": "list",
    "data": [
        {
            "object": "organization.project.user",
            "id": "user_abc",
            "name": "First Last",
            "email": "user@example.com",
            "role": "owner",
            "added_at": 1711471533
        }
    ],
    "first_id": "user-abc",
    "last_id": "user-xyz",
    "has_more": false
}
```

Create project user

```
POST https://api.openai.com/v1/organization/projects/{project_id}/users
```

Adds a user to the project. Users must already be members of the organization to be added to a project.

Path parameters

project_id string Required
The ID of the project.

Request body

role string Required
`owner` or `member`

user_id string Required
The ID of the user.

Returns

The created ProjectUser object.

Example request curl

```curl
curl -X POST https://api.openai.com/v1/organization/projects/proj_abc/users \
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json" \
  -d '{
      "user_id": "user_abc",
      "role": "member"
  }'
```

Response

```
{
    "object": "organization.project.user",
    "id": "user_abc",
    "email": "user@example.com",
    "role": "owner",
```

```
    "added_at": 1711471533
}
```

Retrieve project user

```
GET https://api.openai.com/v1/organization/projects/{project_id}/users/{user_id}
```

Retrieves a user in the project.

Path parameters

project_id string Required
The ID of the project.

user_id string Required
The ID of the user.

Returns

The ProjectUser object matching the specified ID.

Example request curl

```
curl https://api.openai.com/v1/organization/projects/proj_abc/users/user_abc \
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
    "object": "organization.project.user",
    "id": "user_abc",
    "name": "First Last",
    "email": "user@example.com",
    "role": "owner",
    "added_at": 1711471533
}
```

Modify project user

```
POST https://api.openai.com/v1/organization/projects/{project_id}/users/{user_id}
```

Modifies a user's role in the project.

Path parameters

`project_id` string Required
The ID of the project.

`user_id` string Required
The ID of the user.

Request body

`role` string Required
`owner` or `member`

Returns

The updated ProjectUser object.

Example request curl

```
curl -X POST https://api.openai.com/v1/organization/projects/proj_abc/users/user_abc \
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json" \
  -d '{
      "role": "owner"
  }'
```

Response

```
{
    "object": "organization.project.user",
    "id": "user_abc",
    "name": "First Last",
    "email": "user@example.com",
    "role": "owner",
    "added_at": 1711471533
```

```
}
```

Delete project user

```
DELETE https://api.openai.com/v1/organization/projects/{project_id}/users/{user_id}
```

Deletes a user from the project.

Path parameters

project_id string Required
The ID of the project.

user_id string Required
The ID of the user.

Returns

Confirmation that project has been deleted or an error in case of an archived project, which has no users

Example request curl

```curl
curl -X DELETE https://api.openai.com/v1/organization/projects/proj_abc/users/user_abc \
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
    "object": "organization.project.user.deleted",
    "id": "user_abc",
    "deleted": true
}
```

The project user object

Represents an individual user in a project.

added_at integer
The Unix timestamp (in seconds) of when the project was added.

email string
The email address of the user

id string
The identifier, which can be referenced in API endpoints

name string
The name of the user

object string
The object type, which is always `organization.project.user`

role string
`owner` or `member`

OBJECT The project user object

```
{
    "object": "organization.project.user",
    "id": "user_abc",
    "name": "First Last",
    "email": "user@example.com",
    "role": "owner",
    "added_at": 1711471533
}
```

Project service accounts

Manage service accounts within a project. A service account is a bot user that is not associated with a user. If a user leaves an organization, their keys and membership in projects will no longer work. Service accounts do not have this limitation. However, service accounts can also be deleted from a project.

List project service accounts

```
GET https://api.openai.com/v1/organization/projects/{project_id}/service_accounts
```

Returns a list of service accounts in the project.

Path parameters

`project_id` string Required
The ID of the project.

Query parameters

`after` string Optional
A cursor for use in pagination. `after` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, ending with obj_foo, your subsequent call can include after=obj_foo in order to fetch the next page of the list.

`limit` integer Optional Defaults to 20
A limit on the number of objects to be returned. Limit can range between 1 and 100, and the default is 20.

Returns

A list of ProjectServiceAccount objects.

Example request curl

```curl
curl https://api.openai.com/v1/organization/projects/proj_abc/service_accounts?after=custom
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
    "object": "list",
    "data": [
        {
            "object": "organization.project.service_account",
```

```
            "id": "svc_acct_abc",
            "name": "Service Account",
            "role": "owner",
            "created_at": 1711471533
        }
    ],
    "first_id": "svc_acct_abc",
    "last_id": "svc_acct_xyz",
    "has_more": false
}
```

Create project service account

`POST https://api.openai.com/v1/organization/projects/{project_id}/service_accounts`

Creates a new service account in the project. This also returns an unredacted API key for the service account.

Path parameters

project_id string Required
The ID of the project.

Request body

name string Required
The name of the service account being created.

Returns

The created ProjectServiceAccount object.

Example request curl

```
curl -X POST https://api.openai.com/v1/organization/projects/proj_abc/service_accounts \
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json" \
```

```
4   -d '{
5       "name": "Production App"
6   }'
```

Response

```
1  {
2      "object": "organization.project.service_account",
3      "id": "svc_acct_abc",
4      "name": "Production App",
5      "role": "member",
6      "created_at": 1711471533,
7      "api_key": {
8          "object": "organization.project.service_account.api_key",
9          "value": "sk-abcdefghijklmnop123",
10         "name": "Secret Key",
11         "created_at": 1711471533,
12         "id": "key_abc"
13     }
14 }
```

Retrieve project service account

```
GET https://api.openai.com/v1/organization/projects/{project_id}/service_accounts/{service_account_id}
```

Retrieves a service account in the project.

Path parameters

`project_id` string Required
The ID of the project.

`service_account_id` string Required
The ID of the service account.

Returns

The ProjectServiceAccount object matching the specified ID.

Example request curl

```
curl https://api.openai.com/v1/organization/projects/proj_abc/service_accounts/svc_acct_abc
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
    "object": "organization.project.service_account",
    "id": "svc_acct_abc",
    "name": "Service Account",
    "role": "owner",
    "created_at": 1711471533
}
```

Delete project service account

```
DELETE https://api.openai.com/v1/organization/projects/{project_id}/service_accounts/{service
       _account_id}
```

Deletes a service account from the project.

Path parameters

project_id string Required
The ID of the project.

service_account_id string Required
The ID of the service account.

Returns

Confirmation of service account being deleted, or an error in case of an archived project, which has no service accounts

Example request curl

```
curl -X DELETE https://api.openai.com/v1/organization/projects/proj_abc/service_accounts/sv
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
    "object": "organization.project.service_account.deleted",
    "id": "svc_acct_abc",
    "deleted": true
}
```

The project service account object

Represents an individual service account in a project.

created_at integer
The Unix timestamp (in seconds) of when the service account was created

id string
The identifier, which can be referenced in API endpoints

name string
The name of the service account

object string
The object type, which is always `organization.project.service_account`

role string
`owner` or `member`

OBJECT The project service account object

```
{
    "object": "organization.project.service_account",
    "id": "svc_acct_abc",
    "name": "Service Account",
    "role": "owner",
    "created_at": 1711471533
}
```

Project API keys

Manage API keys for a given project. Supports listing and deleting keys for users. This API does not allow issuing keys for users, as users need to authorize themselves to generate keys.

List project API keys

`GET https://api.openai.com/v1/organization/projects/{project_id}/api_keys`

Returns a list of API keys in the project.

Path parameters

`project_id` string Required
The ID of the project.

Query parameters

`after` string Optional
A cursor for use in pagination. `after` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, ending with obj_foo, your subsequent call can include after=obj_foo in order to fetch the next page of the list.

`limit` integer Optional Defaults to 20
A limit on the number of objects to be returned. Limit can range between 1 and 100, and the default is 20.

Returns

A list of ProjectApiKey objects.

Example request curl

```
curl https://api.openai.com/v1/organization/projects/proj_abc/api_keys?after=key_abc&limit=
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
    "object": "list",
    "data": [
        {
            "object": "organization.project.api_key",
            "redacted_value": "sk-abc...def",
            "name": "My API Key",
            "created_at": 1711471533,
            "last_used_at": 1711471534,
            "id": "key_abc",
            "owner": {
                "type": "user",
                "user": {
                    "object": "organization.project.user",
                    "id": "user_abc",
                    "name": "First Last",
                    "email": "user@example.com",
                    "role": "owner",
                    "added_at": 1711471533
                }
            }
        }
    ],
    "first_id": "key_abc",
    "last_id": "key_xyz",
    "has_more": false
}
```

Retrieve project API key

```
GET https://api.openai.com/v1/organization/projects/{project_id}/api_keys/{key_id}
```

Retrieves an API key in the project.

Path parameters

key_id string Required
The ID of the API key.

project_id string Required
The ID of the project.

Returns

The ProjectApiKey object matching the specified ID.

Example request curl

```curl
curl https://api.openai.com/v1/organization/projects/proj_abc/api_keys/key_abc \
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
    "object": "organization.project.api_key",
    "redacted_value": "sk-abc...def",
    "name": "My API Key",
    "created_at": 1711471533,
    "last_used_at": 1711471534,
    "id": "key_abc",
    "owner": {
        "type": "user",
        "user": {
            "object": "organization.project.user",
            "id": "user_abc",
            "name": "First Last",
            "email": "user@example.com",
            "role": "owner",
            "added_at": 1711471533
        }
    }
}
```

Delete project API key

`DELETE https://api.openai.com/v1/organization/projects/{project_id}/api_keys/{key_id}`

Deletes an API key from the project.

Path parameters

key_id string Required
The ID of the API key.

project_id string Required
The ID of the project.

Returns

Confirmation of the key's deletion or an error if the key belonged to a service account

Example request curl

```curl
curl -X DELETE https://api.openai.com/v1/organization/projects/proj_abc/api_keys/key_abc \
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
    "object": "organization.project.api_key.deleted",
    "id": "key_abc",
    "deleted": true
}
```

The project API key object

Represents an individual API key in a project.

created_at integer
The Unix timestamp (in seconds) of when the API key was created

id string
The identifier, which can be referenced in API endpoints

last_used_at integer
The Unix timestamp (in seconds) of when the API key was last used.

name string
The name of the API key

object string
The object type, which is always `organization.project.api_key`

owner object

^ Hide properties

service_account object

Represents an individual service account in a project.

Hide properties

created_at integer

The Unix timestamp (in seconds) of when the service account was created

id string

The identifier, which can be referenced in API endpoints

name string

The name of the service account

object string

The object type, which is always `organization.project.service_account`

role string

`owner` or `member`

type string

`user` or `service_account`

user object

Represents an individual user in a project.

Hide properties

added_at integer

The Unix timestamp (in seconds) of when the project was added.

email string

The email address of the user

id string

The identifier, which can be referenced in API endpoints

name string

The name of the user

object string

The object type, which is always `organization.project.user`

role string

`owner` or `member`

redacted_value string
The redacted value of the API key

OBJECT The project API key object

```
{
    "object": "organization.project.api_key",
    "redacted_value": "sk-abc...def",
    "name": "My API Key",
    "created_at": 1711471533,
    "last_used_at": 1711471534,
    "id": "key_abc",
    "owner": {
        "type": "user",
        "user": {
            "object": "organization.project.user",
            "id": "user_abc",
            "name": "First Last",
            "email": "user@example.com",
            "role": "owner",
            "created_at": 1711471533
        }
    }
}
```

Project rate limits

Manage rate limits per model for projects. Rate limits may be configured to be equal to or lower than the organization's rate limits.

List project rate limits

`GET https://api.openai.com/v1/organization/projects/{project_id}/rate_limits`

Returns the rate limits per model for a project.

Path parameters

project_id string Required
The ID of the project.

Query parameters

after string Optional
A cursor for use in pagination. `after` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, ending with obj_foo, your subsequent call can include after=obj_foo in order to fetch the next page of the list.

before string Optional
A cursor for use in pagination. `before` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, beginning with obj_foo, your subsequent call can include before=obj_foo in order to fetch the previous page of the list.

limit integer Optional Defaults to 100
A limit on the number of objects to be returned. The default is 100.

Returns

A list of ProjectRateLimit objects.

Example request curl

```
curl https://api.openai.com/v1/organization/projects/proj_abc/rate_limits?after=rl_xxx&limi
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
    "object": "list",
    "data": [
        {
          "object": "project.rate_limit",
          "id": "rl-ada",
          "model": "ada",
```

```
            "max_requests_per_1_minute": 600,
            "max_tokens_per_1_minute": 150000,
            "max_images_per_1_minute": 10
        }
    ],
    "first_id": "rl-ada",
    "last_id": "rl-ada",
    "has_more": false
}
```

Modify project rate limit

```
POST https://api.openai.com/v1/organization/projects/{project_id}/rate_limits/{rate_limit_id}
```

Updates a project rate limit.

Path parameters

`project_id` string Required
The ID of the project.

`rate_limit_id` string Required
The ID of the rate limit.

Request body

`batch_1_day_max_input_tokens` integer Optional
The maximum batch input tokens per day. Only relevant for certain models.

`max_audio_megabytes_per_1_minute` integer Optional
The maximum audio megabytes per minute. Only relevant for certain models.

`max_images_per_1_minute` integer Optional
The maximum images per minute. Only relevant for certain models.

`max_requests_per_1_day` integer Optional
The maximum requests per day. Only relevant for certain models.

max_requests_per_1_minute integer Optional
The maximum requests per minute.

max_tokens_per_1_minute integer Optional
The maximum tokens per minute.

Returns

The updated ProjectRateLimit object.

Example request curl

```
curl -X POST https://api.openai.com/v1/organization/projects/proj_abc/rate_limits/rl_xxx \
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json" \
  -d '{
      "max_requests_per_1_minute": 500
  }'
```

Response

```
{
    "object": "project.rate_limit",
    "id": "rl-ada",
    "model": "ada",
    "max_requests_per_1_minute": 600,
    "max_tokens_per_1_minute": 150000,
    "max_images_per_1_minute": 10
  }
```

The project rate limit object

Represents a project rate limit config.

batch_1_day_max_input_tokens integer
The maximum batch input tokens per day. Only present for relevant models.

id string

The identifier, which can be referenced in API endpoints.

max_audio_megabytes_per_1_minute integer
The maximum audio megabytes per minute. Only present for relevant models.

max_images_per_1_minute integer
The maximum images per minute. Only present for relevant models.

max_requests_per_1_day integer
The maximum requests per day. Only present for relevant models.

max_requests_per_1_minute integer
The maximum requests per minute.

max_tokens_per_1_minute integer
The maximum tokens per minute.

model string
The model this rate limit applies to.

object string
The object type, which is always `project.rate_limit`

OBJECT The project rate limit object

```
{
    "object": "project.rate_limit",
    "id": "rl_ada",
    "model": "ada",
    "max_requests_per_1_minute": 600,
    "max_tokens_per_1_minute": 150000,
    "max_images_per_1_minute": 10
}
```

Audit logs

Logs of user actions and configuration changes within this organization. To log events, an Organization Owner must activate logging in the Data Controls Settings. Once activated, for security reasons, logging cannot be deactivated.

List audit logs

```
GET https://api.openai.com/v1/organization/audit_logs
```

List user actions and configuration changes within this organization.

Query parameters

`actor_emails[]` array Optional
Return only events performed by users with these emails.

`actor_ids[]` array Optional
Return only events performed by these actors. Can be a user ID, a service account ID, or an api key tracking ID.

`after` string Optional
A cursor for use in pagination. `after` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, ending with obj_foo, your subsequent call can include after=obj_foo in order to fetch the next page of the list.

`before` string Optional
A cursor for use in pagination. `before` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, starting with obj_foo, your subsequent call can include before=obj_foo in order to fetch the previous page of the list.

`effective_at` object Optional
Return only events whose `effective_at` (Unix seconds) is in this range.

^ Hide properties

`gt` integer Optional
Return only events whose `effective_at` (Unix seconds) is greater than this value.

gte integer Optional
Return only events whose `effective_at` (Unix seconds) is greater than or equal to this value.

lt integer Optional
Return only events whose `effective_at` (Unix seconds) is less than this value.

lte integer Optional
Return only events whose `effective_at` (Unix seconds) is less than or equal to this value.

event_types[] array Optional
Return only events with a `type` in one of these values. For example, `project.created`. For all options, see the documentation for the audit log object.

limit integer Optional Defaults to 20
A limit on the number of objects to be returned. Limit can range between 1 and 100, and the default is 20.

project_ids[] array Optional
Return only events for these projects.

resource_ids[] array Optional
Return only events performed on these targets. For example, a project ID updated.

Returns

A list of paginated Audit Log objects.

Example request curl

```curl
curl https://api.openai.com/v1/organization/audit_logs \
-H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
-H "Content-Type: application/json"
```

Response

```json
{
    "object": "list",
    "data": [
        {
            "id": "audit_log-xxx_yyyymmdd",
```

```
            "type": "project.archived",
            "effective_at": 1722461446,
            "actor": {
                "type": "api_key",
                "api_key": {
                    "type": "user",
                    "user": {
                        "id": "user-xxx",
                        "email": "user@example.com"
                    }
                }
            },
            "project.archived": {
                "id": "proj_abc"
            },
        },
        {
            "id": "audit_log-yyy__20240101",
            "type": "api_key.updated",
            "effective_at": 1720804190,
            "actor": {
                "type": "session",
                "session": {
                    "user": {
                        "id": "user-xxx",
                        "email": "user@example.com"
                    },
                    "ip_address": "127.0.0.1",
                    "user_agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/5
                    "ja3": "a497151ce4338a12c4418c44d375173e",
                    "ja4": "q13d0313h3_55b375c5d22e_c7319ce65786",
                    "ip_address_details": {
                      "country": "US",
                      "city": "San Francisco",
                      "region": "California",
                      "region_code": "CA",
                      "asn": "1234",
                      "latitude": "37.77490",
                      "longitude": "-122.41940"
                    }
                }
            },
            "api_key.updated": {
                "id": "key_xxxx",
                "data": {
                    "scopes": ["resource_2.operation_2"]
                }
```

```
            },
        }
    ],
    "first_id": "audit_log-xxx__20240101",
    "last_id": "audit_log_yyy__20240101",
    "has_more": true
}
```

The audit log object

A log of a user action or configuration change within this organization.

actor object

The actor who performed the audit logged action.

Hide properties

api_key object

The API Key used to perform the audit logged action.

Hide properties

id string

The tracking id of the API key.

service_account object

The service account that performed the audit logged action.

Hide properties

id string

The service account id.

type string

The type of API key. Can be either `user` or `service_account`.

user object

The user who performed the audit logged action.

Hide properties

email string

The user email.

id string
The user id.

session object
The session in which the audit logged action was performed.

Hide properties

ip_address string
The IP address from which the action was performed.

user object
The user who performed the audit logged action.

Hide properties

email string
The user email.

id string
The user id.

type string
The type of actor. Is either `session` or `api_key`.

api_key.created object
The details for events with this `type`.

Hide properties

data object
The payload used to create the API key.

Hide properties

scopes array
A list of scopes allowed for the API key, e.g. `["api.model.request"]`

id string
The tracking ID of the API key.

api_key.deleted object

The details for events with this `type`.

Hide properties

id string
The tracking ID of the API key.

api_key.updated object
The details for events with this `type`.

Hide properties

changes_requested object
The payload used to update the API key.

Hide properties

scopes array
A list of scopes allowed for the API key, e.g. `["api.model.request"]`

id string
The tracking ID of the API key.

certificate.created object
The details for events with this `type`.

Hide properties

id string
The certificate ID.

name string
The name of the certificate.

certificate.deleted object
The details for events with this `type`.

Hide properties

certificate string
The certificate content in PEM format.

id string
The certificate ID.

name string
The name of the certificate.

certificate.updated object
The details for events with this `type`.

Hide properties

id string
The certificate ID.

name string
The name of the certificate.

certificates.activated object
The details for events with this `type`.

Hide properties

certificates array

Hide properties

id string
The certificate ID.

name string
The name of the certificate.

certificates.deactivated object
The details for events with this `type`.

Hide properties

certificates array

Hide properties

id string
The certificate ID.

name string
The name of the certificate.

checkpoint_permission.created object

The project and fine-tuned model checkpoint that the checkpoint permission was created for.

Hide properties

data object

The payload used to create the checkpoint permission.

Hide properties

fine_tuned_model_checkpoint string

The ID of the fine-tuned model checkpoint.

project_id string

The ID of the project that the checkpoint permission was created for.

id string

The ID of the checkpoint permission.

checkpoint_permission.deleted object

The details for events with this `type`.

Hide properties

id string

The ID of the checkpoint permission.

effective_at integer

The Unix timestamp (in seconds) of the event.

id string

The ID of this log.

invite.accepted object

The details for events with this `type`.

Hide properties

id string

The ID of the invite.

invite.deleted object

The details for events with this `type`.

Hide properties

id string
The ID of the invite.

invite.sent object
The details for events with this `type`.

Hide properties

data object
The payload used to create the invite.

Hide properties

email string
The email invited to the organization.

role string
The role the email was invited to be. Is either `owner` or `member`.

id string
The ID of the invite.

login.failed object
The details for events with this `type`.

Hide properties

error_code string
The error code of the failure.

error_message string
The error message of the failure.

logout.failed object
The details for events with this `type`.

Hide properties

error_code string
The error code of the failure.

error_message string

The error message of the failure.

organization.updated object

The details for events with this `type`.

Hide properties

changes_requested object

The payload used to update the organization settings.

Hide properties

api_call_logging string

How your organization logs data from supported API calls. One of `disabled`, `enabled_per_call`, `enabled_for_all_projects`, or `enabled_for_selected_projects`

api_call_logging_project_ids string

The list of project ids if api_call_logging is set to `enabled_for_selected_projects`

description string

The organization description.

name string

The organization name.

threads_ui_visibility string

Visibility of the threads page which shows messages created with the Assistants API and Playground. One of `ANY_ROLE`, `OWNERS`, or `NONE`.

title string

The organization title.

usage_dashboard_visibility string

Visibility of the usage dashboard which shows activity and costs for your organization. One of `ANY_ROLE` or `OWNERS`.

id string

The organization ID.

project object

The project that the action was scoped to. Absent for actions not scoped to projects.

Hide properties

id string
The project ID.

name string
The project title.

project.archived object
The details for events with this `type`.

Hide properties

id string
The project ID.

project.created object
The details for events with this `type`.

Hide properties

data object
The payload used to create the project.

Hide properties

name string
The project name.

title string
The title of the project as seen on the dashboard.

id string
The project ID.

project.updated object
The details for events with this `type`.

Hide properties

changes_requested object
The payload used to update the project.

Hide properties

title string
The title of the project as seen on the dashboard.

id string
The project ID.

rate_limit.deleted object
The details for events with this `type`.

Hide properties

id string
The rate limit ID

rate_limit.updated object
The details for events with this `type`.

Hide properties

changes_requested object
The payload used to update the rate limits.

Hide properties

batch_1_day_max_input_tokens integer
The maximum batch input tokens per day. Only relevant for certain models.

max_audio_megabytes_per_1_minute integer
The maximum audio megabytes per minute. Only relevant for certain models.

max_images_per_1_minute integer
The maximum images per minute. Only relevant for certain models.

max_requests_per_1_day integer
The maximum requests per day. Only relevant for certain models.

max_requests_per_1_minute integer
The maximum requests per minute.

max_tokens_per_1_minute integer
The maximum tokens per minute.

id string

The rate limit ID

service_account.created object

The details for events with this `type`.

Hide properties

data object

The payload used to create the service account.

Hide properties

role string

The role of the service account. Is either `owner` or `member`.

id string

The service account ID.

service_account.deleted object

The details for events with this `type`.

Hide properties

id string

The service account ID.

service_account.updated object

The details for events with this `type`.

Hide properties

changes_requested object

The payload used to updated the service account.

Hide properties

role string

The role of the service account. Is either `owner` or `member`.

id string

The service account ID.

type string
The event type.

user.added object
The details for events with this `type`.

Hide properties

data object
The payload used to add the user to the project.

Hide properties

role string
The role of the user. Is either `owner` or `member`.

id string
The user ID.

user.deleted object
The details for events with this `type`.

Hide properties

id string
The user ID.

user.updated object
The details for events with this `type`.

Hide properties

changes_requested object
The payload used to update the user.

Hide properties

role string
The role of the user. Is either `owner` or `member`.

id string
The project ID.

```
OBJECT The audit log object
```

```
{
    "id": "req_xxx_20240101",
    "type": "api_key.created",
    "effective_at": 1720804090,
    "actor": {
        "type": "session",
        "session": {
            "user": {
                "id": "user-xxx",
                "email": "user@example.com"
            },
            "ip_address": "127.0.0.1",
            "user_agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (K
        }
    },
    "api_key.created": {
        "id": "key_xxxx",
        "data": {
            "scopes": ["resource.operation"]
        }
    }
}
```

Usage

The **Usage API** provides detailed insights into your activity across the OpenAI API. It also includes a separate Costs endpoint, which offers visibility into your spend, breaking down consumption by invoice line items and project IDs.

While the Usage API delivers granular usage data, it may not always reconcile perfectly with the Costs due to minor differences in how usage and spend are recorded. For financial purposes, we recommend using the Costs endpoint or the Costs tab in the Usage Dashboard, which will reconcile back to your billing invoice.

Completions

`GET https://api.openai.com/v1/organization/usage/completions`

Get completions usage details for the organization.

Query parameters

start_time integer Required
Start time (Unix seconds) of the query time range, inclusive.

api_key_ids array Optional
Return only usage for these API keys.

batch boolean Optional
If `true`, return batch jobs only. If `false`, return non-batch jobs only. By default, return both.

bucket_width string Optional Defaults to 1d
Width of each time bucket in response. Currently `1m`, `1h` and `1d` are supported, default to `1d`.

end_time integer Optional
End time (Unix seconds) of the query time range, exclusive.

group_by array Optional
Group the usage data by the specified fields. Support fields include `project_id`, `user_id`, `api_key_id`, `model`, `batch` or any combination of them.

limit integer Optional
Specifies the number of buckets to return.

- `bucket_width=1d`: default: 7, max: 31
- `bucket_width=1h`: default: 24, max: 168
- `bucket_width=1m`: default: 60, max: 1440

models array Optional
Return only usage for these models.

page string Optional
A cursor for use in pagination. Corresponding to the `next_page` field from the previous response.

project_ids array Optional
Return only usage for these projects.

user_ids array Optional
Return only usage for these users.

Returns

A list of paginated, time bucketed Completions usage objects.

Example request curl

```
curl "https://api.openai.com/v1/organization/usage/completions?start_time=1730419200&limit=
-H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
-H "Content-Type: application/json"
```

Response

```
{
    "object": "page",
    "data": [
        {
            "object": "bucket",
            "start_time": 1730419200,
            "end_time": 1730505600,
            "results": [
                {
                    "object": "organization.usage.completions.result",
                    "input_tokens": 1000,
                    "output_tokens": 500,
                    "input_cached_tokens": 800,
                    "input_audio_tokens": 0,
                    "output_audio_tokens": 0,
                    "num_model_requests": 5,
                    "project_id": null,
                    "user_id": null,
                    "api_key_id": null,
                    "model": null,
                    "batch": null
                }
            ]
        }
    ],
    "has_more": true,
    "next_page": "page_AAAAAGdGxdEiJdKOAAAAAGcqsYA="
}
```

Completions usage object

The aggregated completions usage details of the specific time bucket.

api_key_id string or null
When `group_by=api_key_id`, this field provides the API key ID of the grouped usage result.

batch boolean or null
When `group_by=batch`, this field tells whether the grouped usage result is batch or not.

input_audio_tokens integer
The aggregated number of audio input tokens used, including cached tokens.

input_cached_tokens integer
The aggregated number of text input tokens that has been cached from previous requests. For customers subscribe to scale tier, this includes scale tier tokens.

input_tokens integer
The aggregated number of text input tokens used, including cached tokens. For customers subscribe to scale tier, this includes scale tier tokens.

model string or null
When `group_by=model`, this field provides the model name of the grouped usage result.

num_model_requests integer
The count of requests made to the model.

object string

output_audio_tokens integer
The aggregated number of audio output tokens used.

output_tokens integer
The aggregated number of text output tokens used. For customers subscribe to scale tier, this includes scale tier tokens.

project_id string or null
When `group_by=project_id`, this field provides the project ID of the grouped usage result.

user_id string or null

When `group_by=user_id`, this field provides the user ID of the grouped usage result.

OBJECT Completions usage object

```
{
    "object": "organization.usage.completions.result",
    "input_tokens": 5000,
    "output_tokens": 1000,
    "input_cached_tokens": 4000,
    "input_audio_tokens": 300,
    "output_audio_tokens": 200,
    "num_model_requests": 5,
    "project_id": "proj_abc",
    "user_id": "user-abc",
    "api_key_id": "key_abc",
    "model": "gpt-4o-mini-2024-07-18",
    "batch": false
}
```

Embeddings

`GET https://api.openai.com/v1/organization/usage/embeddings`

Get embeddings usage details for the organization.

Query parameters

`start_time` integer Required
Start time (Unix seconds) of the query time range, inclusive.

`api_key_ids` array Optional
Return only usage for these API keys.

`bucket_width` string Optional Defaults to 1d
Width of each time bucket in response. Currently `1m`, `1h` and `1d` are supported, default to `1d`.

`end_time` integer Optional
End time (Unix seconds) of the query time range, exclusive.

`group_by` array Optional

Group the usage data by the specified fields. Support fields include `project_id`, `user_id`, `api_key_id`, `model` or any combination of them.

`limit` integer Optional

Specifies the number of buckets to return.

- `bucket_width=1d` : default: 7, max: 31
- `bucket_width=1h` : default: 24, max: 168
- `bucket_width=1m` : default: 60, max: 1440

`models` array Optional

Return only usage for these models.

`page` string Optional

A cursor for use in pagination. Corresponding to the `next_page` field from the previous response.

`project_ids` array Optional

Return only usage for these projects.

`user_ids` array Optional

Return only usage for these users.

Returns

A list of paginated, time bucketed Embeddings usage objects.

Example request curl

```
curl "https://api.openai.com/v1/organization/usage/embeddings?start_time=1730419200&limit=1
-H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
-H "Content-Type: application/json"
```

Response

```
{
    "object": "page",
    "data": [
        {
            "object": "bucket",
```

```
            "start_time": 1730419200,
            "end_time": 1730505600,
            "results": [
                {
                    "object": "organization.usage.embeddings.result",
                    "input_tokens": 16,
                    "num_model_requests": 2,
                    "project_id": null,
                    "user_id": null,
                    "api_key_id": null,
                    "model": null
                }
            ]
        }
    ],
    "has_more": false,
    "next_page": null
}
```

Embeddings usage object

The aggregated embeddings usage details of the specific time bucket.

api_key_id string or null
When `group_by=api_key_id`, this field provides the API key ID of the grouped usage result.

input_tokens integer
The aggregated number of input tokens used.

model string or null
When `group_by=model`, this field provides the model name of the grouped usage result.

num_model_requests integer
The count of requests made to the model.

object string

project_id string or null
When `group_by=project_id`, this field provides the project ID of the grouped usage result.

user_id string or null
When `group_by=user_id`, this field provides the user ID of the grouped usage result.

OBJECT Embeddings usage object

```
{
    "object": "organization.usage.embeddings.result",
    "input_tokens": 20,
    "num_model_requests": 2,
    "project_id": "proj_abc",
    "user_id": "user-abc",
    "api_key_id": "key_abc",
    "model": "text-embedding-ada-002-v2"
}
```

Moderations

`GET https://api.openai.com/v1/organization/usage/moderations`

Get moderations usage details for the organization.

Query parameters

start_time integer Required
Start time (Unix seconds) of the query time range, inclusive.

api_key_ids array Optional
Return only usage for these API keys.

bucket_width string Optional Defaults to 1d
Width of each time bucket in response. Currently `1m`, `1h` and `1d` are supported, default to `1d`.

end_time integer Optional
End time (Unix seconds) of the query time range, exclusive.

group_by array Optional
Group the usage data by the specified fields. Support fields include `project_id`, `user_id`, `api_key_id`, `model` or any combination of them.

limit integer Optional

Specifies the number of buckets to return.

- `bucket_width=1d` : default: 7, max: 31
- `bucket_width=1h` : default: 24, max: 168
- `bucket_width=1m` : default: 60, max: 1440

models array Optional

Return only usage for these models.

page string Optional

A cursor for use in pagination. Corresponding to the `next_page` field from the previous response.

project_ids array Optional

Return only usage for these projects.

user_ids array Optional

Return only usage for these users.

Returns

A list of paginated, time bucketed Moderations usage objects.

Example request curl

```curl
curl "https://api.openai.com/v1/organization/usage/moderations?start_time=1730419200&limit=
-H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
-H "Content-Type: application/json"
```

Response

```json
{
    "object": "page",
    "data": [
        {
            "object": "bucket",
            "start_time": 1730419200,
            "end_time": 1730505600,
            "results": [
```

```
                {
                    "object": "organization.usage.moderations.result",
                    "input_tokens": 16,
                    "num_model_requests": 2,
                    "project_id": null,
                    "user_id": null,
                    "api_key_id": null,
                    "model": null
                }
            ]
        }
    ],
    "has_more": false,
    "next_page": null
}
```

Moderations usage object

The aggregated moderations usage details of the specific time bucket.

api_key_id string or null
When `group_by=api_key_id`, this field provides the API key ID of the grouped usage result.

input_tokens integer
The aggregated number of input tokens used.

model string or null
When `group_by=model`, this field provides the model name of the grouped usage result.

num_model_requests integer
The count of requests made to the model.

object string

project_id string or null
When `group_by=project_id`, this field provides the project ID of the grouped usage result.

user_id string or null
When `group_by=user_id`, this field provides the user ID of the grouped usage result.

OBJECT Moderations usage object

```
{
    "object": "organization.usage.moderations.result",
    "input_tokens": 20,
    "num_model_requests": 2,
    "project_id": "proj_abc",
    "user_id": "user-abc",
    "api_key_id": "key_abc",
    "model": "text-moderation"
}
```

Images

`GET https://api.openai.com/v1/organization/usage/images`

Get images usage details for the organization.

Query parameters

`start_time` integer Required
Start time (Unix seconds) of the query time range, inclusive.

`api_key_ids` array Optional
Return only usage for these API keys.

`bucket_width` string Optional Defaults to 1d
Width of each time bucket in response. Currently `1m`, `1h` and `1d` are supported, default to `1d`.

`end_time` integer Optional
End time (Unix seconds) of the query time range, exclusive.

`group_by` array Optional
Group the usage data by the specified fields. Support fields include `project_id`, `user_id`, `api_key_id`, `model`, `size`, `source` or any combination of them.

`limit` integer Optional
Specifies the number of buckets to return.

- `bucket_width=1d`: default: 7, max: 31

`bucket_width=1h` : default: 24, max: 168

`bucket_width=1m` : default: 60, max: 1440

`models` array Optional
Return only usage for these models.

`page` string Optional
A cursor for use in pagination. Corresponding to the `next_page` field from the previous response.

`project_ids` array Optional
Return only usage for these projects.

`sizes` array Optional
Return only usages for these image sizes. Possible values are `256x256`, `512x512`, `1024x1024`, `1792x1792`, `1024x1792` or any combination of them.

`sources` array Optional
Return only usages for these sources. Possible values are `image.generation`, `image.edit`, `image.variation` or any combination of them.

`user_ids` array Optional
Return only usage for these users.

Returns

A list of paginated, time bucketed Images usage objects.

Example request curl

```curl
curl "https://api.openai.com/v1/organization/usage/images?start_time=1730419200&limit=1" \
-H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
-H "Content-Type: application/json"
```

Response

```
{
    "object": "page",
    "data": [
```

```
        {
            "object": "bucket",
            "start_time": 1730419200,
            "end_time": 1730505600,
            "results": [
                {
                    "object": "organization.usage.images.result",
                    "images": 2,
                    "num_model_requests": 2,
                    "size": null,
                    "source": null,
                    "project_id": null,
                    "user_id": null,
                    "api_key_id": null,
                    "model": null
                }
            ]
        }
    ],
    "has_more": false,
    "next_page": null
}
```

Images usage object

The aggregated images usage details of the specific time bucket.

api_key_id string or null

When `group_by=api_key_id`, this field provides the API key ID of the grouped usage result.

images integer

The number of images processed.

model string or null

When `group_by=model`, this field provides the model name of the grouped usage result.

num_model_requests integer

The count of requests made to the model.

object string

project_id string or null
When `group_by=project_id`, this field provides the project ID of the grouped usage result.

size string or null
When `group_by=size`, this field provides the image size of the grouped usage result.

source string or null
When `group_by=source`, this field provides the source of the grouped usage result, possible values are `image.generation`, `image.edit`, `image.variation`.

user_id string or null
When `group_by=user_id`, this field provides the user ID of the grouped usage result.

OBJECT Images usage object

```
{
    "object": "organization.usage.images.result",
    "images": 2,
    "num_model_requests": 2,
    "size": "1024x1024",
    "source": "image.generation",
    "project_id": "proj_abc",
    "user_id": "user-abc",
    "api_key_id": "key_abc",
    "model": "dall-e-3"
}
```

Audio speeches

```
GET https://api.openai.com/v1/organization/usage/audio_speeches
```

Get audio speeches usage details for the organization.

Query parameters

start_time integer Required
Start time (Unix seconds) of the query time range, inclusive.

api_key_ids array Optional

Return only usage for these API keys.

bucket_width string Optional Defaults to 1d
Width of each time bucket in response. Currently `1m`, `1h` and `1d` are supported, default to `1d`.

end_time integer Optional
End time (Unix seconds) of the query time range, exclusive.

group_by array Optional
Group the usage data by the specified fields. Support fields include `project_id`, `user_id`, `api_key_id`, `model` or any combination of them.

limit integer Optional
Specifies the number of buckets to return.

- `bucket_width=1d`: default: 7, max: 31
- `bucket_width=1h`: default: 24, max: 168
- `bucket_width=1m`: default: 60, max: 1440

models array Optional
Return only usage for these models.

page string Optional
A cursor for use in pagination. Corresponding to the `next_page` field from the previous response.

project_ids array Optional
Return only usage for these projects.

user_ids array Optional
Return only usage for these users.

Returns

A list of paginated, time bucketed Audio speeches usage objects.

Example request curl

```
curl "https://api.openai.com/v1/organization/usage/audio_speeches?start_time=1730419200&lim
```

```
-H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
-H "Content-Type: application/json"
```

Response

```
{
    "object": "page",
    "data": [
        {
            "object": "bucket",
            "start_time": 1730419200,
            "end_time": 1730505600,
            "results": [
                {
                    "object": "organization.usage.audio_speeches.result",
                    "characters": 45,
                    "num_model_requests": 1,
                    "project_id": null,
                    "user_id": null,
                    "api_key_id": null,
                    "model": null
                }
            ]
        }
    ],
    "has_more": false,
    "next_page": null
}
```

Audio speeches usage object

The aggregated audio speeches usage details of the specific time bucket.

api_key_id string or null

When `group_by=api_key_id`, this field provides the API key ID of the grouped usage result.

characters integer

The number of characters processed.

model string or null

When `group_by=model`, this field provides the model name of the grouped usage result.

num_model_requests integer
The count of requests made to the model.

object string

project_id string or null
When `group_by=project_id`, this field provides the project ID of the grouped usage result.

user_id string or null
When `group_by=user_id`, this field provides the user ID of the grouped usage result.

OBJECT Audio speeches usage object

```
{
    "object": "organization.usage.audio_speeches.result",
    "characters": 45,
    "num_model_requests": 1,
    "project_id": "proj_abc",
    "user_id": "user-abc",
    "api_key_id": "key_abc",
    "model": "tts-1"
}
```

Audio transcriptions

`GET https://api.openai.com/v1/organization/usage/audio_transcriptions`

Get audio transcriptions usage details for the organization.

Query parameters

start_time integer Required
Start time (Unix seconds) of the query time range, inclusive.

api_key_ids array Optional
Return only usage for these API keys.

bucket_width string Optional Defaults to 1d

Width of each time bucket in response. Currently `1m`, `1h` and `1d` are supported, default to `1d`.

`end_time` integer Optional
End time (Unix seconds) of the query time range, exclusive.

`group_by` array Optional
Group the usage data by the specified fields. Support fields include `project_id`, `user_id`, `api_key_id`, `model` or any combination of them.

`limit` integer Optional
Specifies the number of buckets to return.

- `bucket_width=1d`: default: 7, max: 31
- `bucket_width=1h`: default: 24, max: 168
- `bucket_width=1m`: default: 60, max: 1440

`models` array Optional
Return only usage for these models.

`page` string Optional
A cursor for use in pagination. Corresponding to the `next_page` field from the previous response.

`project_ids` array Optional
Return only usage for these projects.

`user_ids` array Optional
Return only usage for these users.

Returns

A list of paginated, time bucketed Audio transcriptions usage objects.

Example request curl

```
curl "https://api.openai.com/v1/organization/usage/audio_transcriptions?start_time=17304192
-H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
-H "Content-Type: application/json"
```

Response

```
{
    "object": "page",
    "data": [
        {
            "object": "bucket",
            "start_time": 1730419200,
            "end_time": 1730505600,
            "results": [
                {
                    "object": "organization.usage.audio_transcriptions.result",
                    "seconds": 20,
                    "num_model_requests": 1,
                    "project_id": null,
                    "user_id": null,
                    "api_key_id": null,
                    "model": null
                }
            ]
        }
    ],
    "has_more": false,
    "next_page": null
}
```

Audio transcriptions usage object

The aggregated audio transcriptions usage details of the specific time bucket.

api_key_id string or null

When `group_by=api_key_id`, this field provides the API key ID of the grouped usage result.

model string or null

When `group_by=model`, this field provides the model name of the grouped usage result.

num_model_requests integer

The count of requests made to the model.

object string

project_id string or null
When `group_by=project_id`, this field provides the project ID of the grouped usage result.

seconds integer
The number of seconds processed.

user_id string or null
When `group_by=user_id`, this field provides the user ID of the grouped usage result.

OBJECT Audio transcriptions usage object

```
{
    "object": "organization.usage.audio_transcriptions.result",
    "seconds": 10,
    "num_model_requests": 1,
    "project_id": "proj_abc",
    "user_id": "user-abc",
    "api_key_id": "key_abc",
    "model": "tts-1"
}
```

Vector stores

`GET https://api.openai.com/v1/organization/usage/vector_stores`

Get vector stores usage details for the organization.

Query parameters

start_time integer Required
Start time (Unix seconds) of the query time range, inclusive.

bucket_width string Optional Defaults to 1d
Width of each time bucket in response. Currently `1m`, `1h` and `1d` are supported, default to `1d`.

end_time integer Optional
End time (Unix seconds) of the query time range, exclusive.

group_by array Optional

Group the usage data by the specified fields. Support fields include `project_id`.

limit integer Optional

Specifies the number of buckets to return.

- `bucket_width=1d`: default: 7, max: 31
- `bucket_width=1h`: default: 24, max: 168
- `bucket_width=1m`: default: 60, max: 1440

page string Optional

A cursor for use in pagination. Corresponding to the `next_page` field from the previous response.

project_ids array Optional

Return only usage for these projects.

Returns

A list of paginated, time bucketed Vector stores usage objects.

Example request curl

```curl
curl "https://api.openai.com/v1/organization/usage/vector_stores?start_time=1730419200&limi
-H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
-H "Content-Type: application/json"
```

Response

```
{
    "object": "page",
    "data": [
        {
            "object": "bucket",
            "start_time": 1730419200,
            "end_time": 1730505600,
            "results": [
                {
                    "object": "organization.usage.vector_stores.result",
                    "usage_bytes": 1024,

```

```
                    "project_id": null
                }
            ]
        }
    ],
    "has_more": false,
    "next_page": null
}
```

Vector stores usage object

The aggregated vector stores usage details of the specific time bucket.

object string

project_id string or null
When `group_by=project_id`, this field provides the project ID of the grouped usage result.

usage_bytes integer
The vector stores usage in bytes.

OBJECT Vector stores usage object

```
{
    "object": "organization.usage.vector_stores.result",
    "usage_bytes": 1024,
    "project_id": "proj_abc"
}
```

Code interpreter sessions

`GET https://api.openai.com/v1/organization/usage/code_interpreter_sessions`

Get code interpreter sessions usage details for the organization.

Query parameters

start_time integer Required
Start time (Unix seconds) of the query time range, inclusive.

bucket_width string Optional Defaults to 1d
Width of each time bucket in response. Currently `1m`, `1h` and `1d` are supported, default to `1d`.

end_time integer Optional
End time (Unix seconds) of the query time range, exclusive.

group_by array Optional
Group the usage data by the specified fields. Support fields include `project_id`.

limit integer Optional
Specifies the number of buckets to return.

- `bucket_width=1d`: default: 7, max: 31
- `bucket_width=1h`: default: 24, max: 168
- `bucket_width=1m`: default: 60, max: 1440

page string Optional
A cursor for use in pagination. Corresponding to the `next_page` field from the previous response.

project_ids array Optional
Return only usage for these projects.

Returns

A list of paginated, time bucketed Code interpreter sessions usage objects.

Example request curl

```
curl "https://api.openai.com/v1/organization/usage/code_interpreter_sessions?start_time=173
-H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
-H "Content-Type: application/json"
```

Response

```
{
```

```
    "object": "page",
    "data": [
        {
            "object": "bucket",
            "start_time": 1730419200,
            "end_time": 1730505600,
            "results": [
                {
                    "object": "organization.usage.code_interpreter_sessions.result",
                    "num_sessions": 1,
                    "project_id": null
                }
            ]
        }
    ],
    "has_more": false,
    "next_page": null
}
```

Code interpreter sessions usage object

The aggregated code interpreter sessions usage details of the specific time bucket.

num_sessions integer
The number of code interpreter sessions.

object string

project_id string or null
When `group_by=project_id`, this field provides the project ID of the grouped usage result.

OBJECT Code interpreter sessions usage object

```
{
    "object": "organization.usage.code_interpreter_sessions.result",
    "num_sessions": 1,
    "project_id": "proj_abc"
}
```

Costs

`GET https://api.openai.com/v1/organization/costs`

Get costs details for the organization.

Query parameters

`start_time` integer Required
Start time (Unix seconds) of the query time range, inclusive.

`bucket_width` string Optional Defaults to 1d
Width of each time bucket in response. Currently only `1d` is supported, default to `1d`.

`end_time` integer Optional
End time (Unix seconds) of the query time range, exclusive.

`group_by` array Optional
Group the costs by the specified fields. Support fields include `project_id`, `line_item` and any combination of them.

`limit` integer Optional Defaults to 7
A limit on the number of buckets to be returned. Limit can range between 1 and 180, and the default is 7.

`page` string Optional
A cursor for use in pagination. Corresponding to the `next_page` field from the previous response.

`project_ids` array Optional
Return only costs for these projects.

Returns

A list of paginated, time bucketed Costs objects.

Example request curl

```curl
curl "https://api.openai.com/v1/organization/costs?start_time=1730419200&limit=1" \
-H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
```

```
-H "Content-Type: application/json"
```

Response

```
{
    "object": "page",
    "data": [
        {
            "object": "bucket",
            "start_time": 1730419200,
            "end_time": 1730505600,
            "results": [
                {
                    "object": "organization.costs.result",
                    "amount": {
                        "value": 0.06,
                        "currency": "usd"
                    },
                    "line_item": null,
                    "project_id": null
                }
            ]
        }
    ],
    "has_more": false,
    "next_page": null
}
```

Costs object

The aggregated costs details of the specific time bucket.

amount object
The monetary value in its associated currency.

Hide properties

currency string
Lowercase ISO-4217 currency e.g. "usd"

value number
The numeric value of the cost.

line_item string or null
When `group_by=line_item`, this field provides the line item of the grouped costs result.

object string

project_id string or null
When `group_by=project_id`, this field provides the project ID of the grouped costs result.

OBJECT Costs object

```
{
    "object": "organization.costs.result",
    "amount": {
      "value": 0.06,
      "currency": "usd"
    },
    "line_item": "Image models",
    "project_id": "proj_abc"
}
```

Certificates Beta

Manage Mutual TLS certificates across your organization and projects.

Learn more about Mutual TLS.

Upload certificate

```
POST https://api.openai.com/v1/organization/certificates
```

Upload a certificate to the organization. This does **not** automatically activate the certificate. Organizations can upload up to 50 certificates.

Request body

content string Required
The certificate content in PEM format

name string Optional
An optional name for the certificate

Returns

A single Certificate object.

Example request curl

```
curl -X POST https://api.openai.com/v1/organization/certificates \
-H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
-H "Content-Type: application/json" \
-d '{
  "name": "My Example Certificate",
  "certificate": "-----BEGIN CERTIFICATE-----\\nMIIDeT...\\n-----END CERTIFICATE-----"
}'
```

Response

```
{
  "object": "certificate",
  "id": "cert_abc",
  "name": "My Example Certificate",
  "created_at": 1234567,
  "certificate_details": {
    "valid_at": 12345667,
    "expires_at": 12345678
  }
}
```

Get certificate

```
GET https://api.openai.com/v1/organization/certificates/{certificate_id}
```

Get a certificate that has been uploaded to the organization.
You can get a certificate regardless of whether it is active or not.

Path parameters

certificate_id string Required
Unique ID of the certificate to retrieve.

Query parameters

include array Optional
A list of additional fields to include in the response. Currently the only supported value is `content` to fetch the PEM content of the certificate.

Returns

A single Certificate object.

Example request curl

```curl
curl "https://api.openai.com/v1/organization/certificates/cert_abc?include[]=content" \
-H "Authorization: Bearer $OPENAI_ADMIN_KEY"
```

Response

```
{
  "object": "certificate",
  "id": "cert_abc",
  "name": "My Example Certificate",
  "created_at": 1234567,
  "certificate_details": {
    "valid_at": 1234567,
    "expires_at": 12345678,
    "content": "-----BEGIN CERTIFICATE-----MIIDeT...-----END CERTIFICATE-----"

```

```
  }
}
```

Modify certificate

`POST https://api.openai.com/v1/organization/certificates/{certificate_id}`

Modify a certificate. Note that only the name can be modified.

Request body

name string Required
The updated name for the certificate

Returns

The updated Certificate object.

Example request curl

```curl
curl -X POST https://api.openai.com/v1/organization/certificates/cert_abc \
-H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
-H "Content-Type: application/json" \
-d '{
  "name": "Renamed Certificate"
}'
```

Response

```
{
  "object": "certificate",
  "id": "cert_abc",
  "name": "Renamed Certificate",
  "created_at": 1234567,
  "certificate_details": {
    "valid_at": 12345667,
    "expires_at": 12345678
  }
}
```

Delete certificate

```
DELETE https://api.openai.com/v1/organization/certificates/{certificate_id}
```

Delete a certificate from the organization.
The certificate must be inactive for the organization and all projects.

Returns

A confirmation object indicating the certificate was deleted.

Example request curl

```
curl -X DELETE https://api.openai.com/v1/organization/certificates/cert_abc \
-H "Authorization: Bearer $OPENAI_ADMIN_KEY"
```

Response

```
{
  "object": "certificate.deleted",
  "id": "cert_abc"
}
```

List organization certificates

```
GET https://api.openai.com/v1/organization/certificates
```

List uploaded certificates for this organization.

Query parameters

after string Optional
A cursor for use in pagination. `after` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, ending with obj_foo, your subsequent call can include after=obj_foo in order to fetch the next page of the list.

limit integer Optional Defaults to 20
A limit on the number of objects to be returned. Limit can range between 1 and 100, and the default is 20.

order string Optional Defaults to desc
Sort order by the `created_at` timestamp of the objects. `asc` for ascending order and `desc` for descending order.

Returns

A list of Certificate objects.

Example request curl

```curl
curl https://api.openai.com/v1/organization/certificates \
-H "Authorization: Bearer $OPENAI_ADMIN_KEY"
```

Response

```
{
  "object": "list",
  "data": [
    {
      "object": "organization.certificate",
      "id": "cert_abc",
      "name": "My Example Certificate",
      "active": true,
      "created_at": 1234567,
      "certificate_details": {
        "valid_at": 12345667,
```

```
12          "expires_at": 12345678
13        }
14      },
15    ],
16    "first_id": "cert_abc",
17    "last_id": "cert_abc",
18    "has_more": false
19  }
```

List project certificates

```
GET https://api.openai.com/v1/organization/projects/{project_id}/certificates
```

List certificates for this project.

Path parameters

`project_id` string Required
The ID of the project.

Query parameters

`after` string Optional
A cursor for use in pagination. `after` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, ending with obj_foo, your subsequent call can include after=obj_foo in order to fetch the next page of the list.

`limit` integer Optional Defaults to 20
A limit on the number of objects to be returned. Limit can range between 1 and 100, and the default is 20.

`order` string Optional Defaults to desc
Sort order by the `created_at` timestamp of the objects. `asc` for ascending order and `desc` for descending order.

Returns

A list of Certificate objects.

Example request curl

```
curl https://api.openai.com/v1/organization/projects/proj_abc/certificates \
-H "Authorization: Bearer $OPENAI_ADMIN_KEY"
```

Response

```
{
  "object": "list",
  "data": [
    {
      "object": "organization.project.certificate",
      "id": "cert_abc",
      "name": "My Example Certificate",
      "active": true,
      "created_at": 1234567,
      "certificate_details": {
        "valid_at": 12345667,
        "expires_at": 12345678
      }
    },
  ],
  "first_id": "cert_abc",
  "last_id": "cert_abc",
  "has_more": false
}
```

Activate certificates for organization

`POST https://api.openai.com/v1/organization/certificates/activate`

Activate certificates at the organization level.
You can atomically and idempotently activate up to 10 certificates at a time.

Request body

certificate_ids array Required

Returns

A list of Certificate objects that were activated.

Example request curl

```curl
curl https://api.openai.com/v1/organization/certificates/activate \
-H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
-H "Content-Type: application/json" \
-d '{
  "data": ["cert_abc", "cert_def"]
}'
```

Response

```json
{
  "object": "organization.certificate.activation",
  "data": [
    {
      "object": "organization.certificate",
      "id": "cert_abc",
      "name": "My Example Certificate",
      "active": true,
      "created_at": 1234567,
      "certificate_details": {
        "valid_at": 12345667,
        "expires_at": 12345678
      }
    },
    {
      "object": "organization.certificate",
      "id": "cert_def",
      "name": "My Example Certificate 2",
      "active": true,
      "created_at": 1234567,
      "certificate_details": {
        "valid_at": 12345667,
        "expires_at": 12345678
      }
    },
  ],
}
```

Deactivate certificates for organization

`POST https://api.openai.com/v1/organization/certificates/deactivate`

Deactivate certificates at the organization level.
You can atomically and idempotently deactivate up to 10 certificates at a time.

Request body

certificate_ids array Required

Returns

A list of Certificate objects that were deactivated.

Example request curl

```
curl https://api.openai.com/v1/organization/certificates/deactivate \
-H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
-H "Content-Type: application/json" \
-d '{
  "data": ["cert_abc", "cert_def"]
}'
```

Response

```
{
  "object": "organization.certificate.deactivation",
  "data": [
    {
      "object": "organization.certificate",
      "id": "cert_abc",
      "name": "My Example Certificate",
      "active": false,
      "created_at": 1234567,
      "certificate_details": {
        "valid_at": 12345667,
        "expires_at": 12345678
      }
    },
    {
```

```
      "object": "organization.certificate",
      "id": "cert_def",
      "name": "My Example Certificate 2",
      "active": false,
      "created_at": 1234567,
      "certificate_details": {
        "valid_at": 12345667,
        "expires_at": 12345678
      }
    },
  ],
}
```

Activate certificates for project

```
POST https://api.openai.com/v1/organization/projects/{project_id}/certificates/activate
```

Activate certificates at the project level.
You can atomically and idempotently activate up to 10 certificates at a time.

Path parameters

project_id string Required
The ID of the project.

Request body

certificate_ids array Required

Returns

A list of Certificate objects that were activated.

Example request curl

```
curl https://api.openai.com/v1/organization/projects/proj_abc/certificates/activate \
-H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
```

```
-H "Content-Type: application/json" \
-d '{
  "data": ["cert_abc", "cert_def"]
}'
```

Response

```
{
  "object": "organization.project.certificate.activation",
  "data": [
    {
      "object": "organization.project.certificate",
      "id": "cert_abc",
      "name": "My Example Certificate",
      "active": true,
      "created_at": 1234567,
      "certificate_details": {
        "valid_at": 12345667,
        "expires_at": 12345678
      }
    },
    {
      "object": "organization.project.certificate",
      "id": "cert_def",
      "name": "My Example Certificate 2",
      "active": true,
      "created_at": 1234567,
      "certificate_details": {
        "valid_at": 12345667,
        "expires_at": 12345678
      }
    },
  ],
}
```

Deactivate certificates for project

`POST https://api.openai.com/v1/organization/projects/{project_id}/certificates/deactivate`

Deactivate certificates at the project level. You can atomically and idempotently deactivate up to 10 certificates at a time.

Path parameters

project_id string Required
The ID of the project.

Request body

certificate_ids array Required

Returns

A list of Certificate objects that were deactivated.

Example request curl

```
curl https://api.openai.com/v1/organization/projects/proj_abc/certificates/deactivate \
-H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
-H "Content-Type: application/json" \
-d '{
  "data": ["cert_abc", "cert_def"]
}'
```

Response

```
{
  "object": "organization.project.certificate.deactivation",
  "data": [
    {
      "object": "organization.project.certificate",
      "id": "cert_abc",
      "name": "My Example Certificate",
      "active": false,
      "created_at": 1234567,
      "certificate_details": {
        "valid_at": 12345667,
        "expires_at": 12345678
      }
    },
    {
      "object": "organization.project.certificate",
      "id": "cert_def",
```

```
      "name": "My Example Certificate 2",
      "active": false,
      "created_at": 1234567,
      "certificate_details": {
        "valid_at": 12345667,
        "expires_at": 12345678
      }
    },
  ],
}
```

The certificate object

Represents an individual `certificate` uploaded to the organization.

`active` boolean
Whether the certificate is currently active at the specified scope. Not returned when getting details for a specific certificate.

`certificate_details` object

Hide properties

`content` string
The content of the certificate in PEM format.

`expires_at` integer
The Unix timestamp (in seconds) of when the certificate expires.

`valid_at` integer
The Unix timestamp (in seconds) of when the certificate becomes valid.

`created_at` integer
The Unix timestamp (in seconds) of when the certificate was uploaded.

`id` string
The identifier, which can be referenced in API endpoints

`name` string
The name of the certificate.

object string

The object type.

If creating, updating, or getting a specific certificate, the object type is `certificate`.

If listing, activating, or deactivating certificates for the organization, the object type is `organization.certificate`.

If listing, activating, or deactivating certificates for a project, the object type is `organization.project.certificate`.

OBJECT The certificate object

```
{
  "object": "certificate",
  "id": "cert_abc",
  "name": "My Certificate",
  "created_at": 1234567,
  "certificate_details": {
    "valid_at": 1234567,
    "expires_at": 12345678,
    "content": "-----BEGIN CERTIFICATE----- MIIGAjCCA...6znFlOW+ -----END CERTIFICATE-----
  }
}
```

Completions Legacy

Given a prompt, the model will return one or more predicted completions along with the probabilities of alternative tokens at each position. Most developer should use our Chat Completions API to leverage our best and newest models.

Create completion Legacy

`POST https://api.openai.com/v1/completions`

Creates a completion for the provided prompt and parameters.

Request body

model string Required

ID of the model to use. You can use the List models API to see all of your available models, or see our Model overview for descriptions of them.

prompt string or array Required

The prompt(s) to generate completions for, encoded as a string, array of strings, array of tokens, or array of token arrays.

Note that <|endoftext|> is the document separator that the model sees during training, so if a prompt is not specified the model will generate as if from the beginning of a new document.

best_of integer or null Optional Defaults to 1

Generates `best_of` completions server-side and returns the "best" (the one with the highest log probability per token). Results cannot be streamed.

When used with `n`, `best_of` controls the number of candidate completions and `n` specifies how many to return – `best_of` must be greater than `n`.

Note: Because this parameter generates many completions, it can quickly consume your token quota. Use carefully and ensure that you have reasonable settings for `max_tokens` and `stop`.

echo boolean or null Optional Defaults to false

Echo back the prompt in addition to the completion

frequency_penalty number or null Optional Defaults to 0

Number between -2.0 and 2.0. Positive values penalize new tokens based on their existing frequency in the text so far, decreasing the model's likelihood to repeat the same line verbatim.

See more information about frequency and presence penalties.

logit_bias map Optional Defaults to null

Modify the likelihood of specified tokens appearing in the completion.

Accepts a JSON object that maps tokens (specified by their token ID in the GPT tokenizer) to an associated bias value from -100 to 100. You can use this tokenizer tool to convert text to token IDs. Mathematically, the bias is added to the logits generated by the model prior to sampling. The exact effect will vary per model, but values between -1 and 1 should decrease or increase likelihood of selection; values like -100 or 100 should result in a ban or exclusive selection of the relevant token.

As an example, you can pass `{"50256": -100}` to prevent the <|endoftext|> token from being generated.

logprobs integer or null Optional Defaults to null

Include the log probabilities on the `logprobs` most likely output tokens, as well the chosen tokens. For example, if `logprobs` is 5, the API will return a list of the 5 most likely tokens. The API will always return the `logprob` of the sampled token, so there may be up to `logprobs+1` elements in the response.

The maximum value for `logprobs` is 5.

`max_tokens` integer or null Optional Defaults to 16

The maximum number of tokens that can be generated in the completion.

The token count of your prompt plus `max_tokens` cannot exceed the model's context length. Example Python code for counting tokens.

`n` integer or null Optional Defaults to 1

How many completions to generate for each prompt.

Note: Because this parameter generates many completions, it can quickly consume your token quota. Use carefully and ensure that you have reasonable settings for `max_tokens` and `stop`.

`presence_penalty` number or null Optional Defaults to 0

Number between -2.0 and 2.0. Positive values penalize new tokens based on whether they appear in the text so far, increasing the model's likelihood to talk about new topics.

See more information about frequency and presence penalties.

`seed` integer or null Optional

If specified, our system will make a best effort to sample deterministically, such that repeated requests with the same `seed` and parameters should return the same result.

Determinism is not guaranteed, and you should refer to the `system_fingerprint` response parameter to monitor changes in the backend.

`stop` string / array / null Optional Defaults to null

Not supported with latest reasoning models `o3` and `o4-mini`.

Up to 4 sequences where the API will stop generating further tokens. The returned text will not contain the stop sequence.

`stream` boolean or null Optional Defaults to false

Whether to stream back partial progress. If set, tokens will be sent as data-only server-sent events as they become available, with the stream terminated by a `data: [DONE]` message. Example Python code.

`stream_options` object or null Optional Defaults to null

Options for streaming response. Only set this when you set `stream: true`.

^ Hide properties

`include_usage` boolean Optional

If set, an additional chunk will be streamed before the `data: [DONE]` message. The `usage` field on this chunk shows the token usage statistics for the entire request, and the `choices` field will always be

an empty array.

All other chunks will also include a `usage` field, but with a null value. **NOTE:** If the stream is interrupted, you may not receive the final usage chunk which contains the total token usage for the request.

`suffix` string or null Optional Defaults to null

The suffix that comes after a completion of inserted text.

This parameter is only supported for `gpt-3.5-turbo-instruct`.

`temperature` number or null Optional Defaults to 1

What sampling temperature to use, between 0 and 2. Higher values like 0.8 will make the output more random, while lower values like 0.2 will make it more focused and deterministic.

We generally recommend altering this or `top_p` but not both.

`top_p` number or null Optional Defaults to 1

An alternative to sampling with temperature, called nucleus sampling, where the model considers the results of the tokens with top_p probability mass. So 0.1 means only the tokens comprising the top 10% probability mass are considered.

We generally recommend altering this or `temperature` but not both.

`user` string Optional

A unique identifier representing your end-user, which can help OpenAI to monitor and detect abuse. Learn more.

Returns

Returns a completion object, or a sequence of completion objects if the request is streamed.

No streaming **Streaming**

Example request

Select... No eligible models available. curl

```
curl https://api.openai.com/v1/completions \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -d '{
    "model": "gpt-3.5-turbo-instruct",
    "prompt": "Say this is a test",
    "max_tokens": 7,
```

```
    "temperature": 0
  }'
```

Response

```
{
  "id": "cmpl-uqkvlQyYK7bGYrRHQ0eXlWi7",
  "object": "text_completion",
  "created": 1589478378,
  "model": "gpt-3.5-turbo-instruct",
  "system_fingerprint": "fp_44709d6fcb",
  "choices": [
    {
      "text": "\n\nThis is indeed a test",
      "index": 0,
      "logprobs": null,
      "finish_reason": "length"
    }
  ],
  "usage": {
    "prompt_tokens": 5,
    "completion_tokens": 7,
    "total_tokens": 12
  }
}
```

The completion object Legacy

Represents a completion response from the API. Note: both the streamed and non-streamed response objects share the same shape (unlike the chat endpoint).

choices array

The list of completion choices the model generated for the input prompt.

Hide properties

finish_reason string

The reason the model stopped generating tokens. This will be `stop` if the model hit a natural stop point or a provided stop sequence, `length` if the maximum number of tokens specified in the request was reached, or `content_filter` if content was omitted due to a flag from our content filters.

index integer

logprobs object or null

Hide properties

text_offset array

token_logprobs array

tokens array

top_logprobs array

Hide properties

text string

created integer
The Unix timestamp (in seconds) of when the completion was created.

id string
A unique identifier for the completion.

model string
The model used for completion.

object string
The object type, which is always "text_completion"

system_fingerprint string
This fingerprint represents the backend configuration that the model runs with.

Can be used in conjunction with the `seed` request parameter to understand when backend changes have been made that might impact determinism.

usage object
Usage statistics for the completion request.

Hide properties

completion_tokens integer
Number of tokens in the generated completion.

prompt_tokens integer

Number of tokens in the prompt.

total_tokens integer
Total number of tokens used in the request (prompt + completion).

completion_tokens_details object
Breakdown of tokens used in a completion.

Hide properties

accepted_prediction_tokens integer
When using Predicted Outputs, the number of tokens in the prediction that appeared in the completion.

audio_tokens integer
Audio input tokens generated by the model.

reasoning_tokens integer
Tokens generated by the model for reasoning.

rejected_prediction_tokens integer
When using Predicted Outputs, the number of tokens in the prediction that did not appear in the completion. However, like reasoning tokens, these tokens are still counted in the total completion tokens for purposes of billing, output, and context window limits.

prompt_tokens_details object
Breakdown of tokens used in the prompt.

Hide properties

audio_tokens integer
Audio input tokens present in the prompt.

cached_tokens integer
Cached tokens present in the prompt.

OBJECT The completion object

```
{
  "id": "cmpl-uqkvlQyYK7bGYrRHQ0eXlWi7",
  "object": "text_completion",
  "created": 1589478378,
  "model": "gpt-4-turbo",
```

```
  "choices": [
    {
      "text": "\n\nThis is indeed a test",
      "index": 0,
      "logprobs": null,
      "finish_reason": "length"
    }
  ],
  "usage": {
    "prompt_tokens": 5,
    "completion_tokens": 7,
    "total_tokens": 12
  }
}
```