

## 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:

```
1 curl https://api.openai.com/v1/models \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "OpenAI-Organization: YOUR_ORG_ID" \
4   -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 `01-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.

---

## 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](#).

✓ Show possible types

---

### **model** string Required

Model ID used to generate the response, like `gpt-4o` or `o1`. 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](#).

✓ Show properties

---

### **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.

✓ Show possible types

---

**functions** Deprecated array Optional

Deprecated in favor of `tools`.

A list of functions the model may generate JSON inputs for.

✓ Show properties

---

**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 [o1 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.

▼ Show possible types

**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.

▼ Show possible types

**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 latency tier to use for processing the request. This parameter is relevant for customers subscribed to the scale tier service:

If set to 'auto', and the Project is Scale tier enabled, the system will utilize scale tier credits until they are exhausted.

If set to 'auto', and the Project is not Scale tier enabled, the request will be processed using the default service tier with a lower uptime SLA and no latency guarantee.

If set to 'default', the request will be processed using the default service tier with a lower uptime SLA and no latency guarantee.

When not set, the default behavior is 'auto'.

When this parameter is set, the response body will include the `service_tier` utilized.

---

**stop** string / array / null Optional Defaults to null

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.

---

**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`.

>Show properties

---

**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.

▼ Show possible types

#### **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.

▼ Show properties

#### **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 unique identifier representing your end-user, which can help OpenAI to monitor and detect 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](#).

▼ Show properties

## 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-4o ⚡ node.js ⚡ [Copy](#)

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const completion = await openai.chat.completions.create({
7     messages: [{ role: "developer", content: "You are a helpful assistant." }],
8     model: "gpt-4o",
9     store: true,
10  });
11
12  console.log(completion.choices[0]);
13 }
14
15 main();
```

## Response



```
1 {
2   "id": "chatcmpl-B9MBs8Cjcv0U2jLn4n570S5qMJKcT",
3   "object": "chat.completion",
4   "created": 1741569952,
5   "model": "gpt-4o-2024-08-06",
6   "choices": [
7     {
8       "index": 0,
9       "message": {
10         "role": "assistant",
11         "content": "Hello! How can I assist you today?",
12         "refusal": null,
13         "annotations": []
14       },
15       "logprobs": null,
16       "finish_reason": "stop"
17     }
18   ],
19   "usage": {
20     "prompt_tokens": 19,
21     "completion_tokens": 10,
22     "total_tokens": 29,
23     "prompt_tokens_details": {
24       "cached_tokens": 0,
25       "audio_tokens": 0
26     },
27     "completion_tokens_details": {
28       "reasoning_tokens": 0,
29       "audio_tokens": 0,
```

```
30     "accepted_prediction_tokens": 0,  
31     "rejected_prediction_tokens": 0  
32   }  
33 },  
34   "service_tier": "default"  
35 }
```

## 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 ⌂

```
1 curl https://api.openai.com/v1/chat/completions/chatcmpl-abc123 \  
2   -H "Authorization: Bearer $OPENAI_API_KEY" \  
3   -H "Content-Type: application/json"
```

#### Response

📋

```
1 {  
2   "object": "chat.completion",  
3   "id": "chatcmpl-abc123",  
4   "model": "gpt-4o-2024-08-06",  
5   "created": 1738960610,  
6   "request_id": "req_ded8ab984ec4bf840f37566c1011c417",  
7   "tool_choice": null,  
8   "usage": {  
9     "total_tokens": 31,
```

```
10     "completion_tokens": 18,
11     "prompt_tokens": 13
12   },
13   "seed": 4944116822809979520,
14   "top_p": 1.0,
15   "temperature": 1.0,
16   "presence_penalty": 0.0,
17   "frequency_penalty": 0.0,
18   "system_fingerprint": "fp_50cad350e4",
19   "input_user": null,
20   "service_tier": "default",
21   "tools": null,
22   "metadata": {},
23   "choices": [
24     {
25       "index": 0,
26       "message": {
27         "content": "Mind of circuits hum, \nLearning patterns in silence- \nFuture",
28         "role": "assistant",
29         "tool_calls": null,
30         "function_call": null
31       },
32       "finish_reason": "stop",
33       "logprobs": null
34     }
35   ],
36   "response_format": null
37 }
```

## 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 ⚡

```
1 curl https://api.openai.com/v1/chat/completions/chat_abc123/messages \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json"
```

## Response

🔗

```
1 {
2   "object": "list",
3   "data": [
4     {
5       "id": "chatmpl-AyPNinnUqUDYo9SAdA52NobMflmj2-0",
6       "role": "user",
7       "content": "write a haiku about ai",
8       "name": null,
9       "content_parts": null
10      }
11    ],
12   "first_id": "chatmpl-AyPNinnUqUDYo9SAdA52NobMflmj2-0",
13   "last_id": "chatmpl-AyPNinnUqUDYo9SAdA52NobMflmj2-0",
14   "has_more": false
15 }
```

# 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 ⌂

```
1 curl https://api.openai.com/v1/chat/completions \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json"
```

### Response

⌂

```
1 {
2   "object": "list",
3   "data": [
```

```
4  {
5      "object": "chat.completion",
6      "id": "chatcmpl-AyPNinnUqUDYo9SAdA52NobMflmj2",
7      "model": "gpt-4o-2024-08-06",
8      "created": 1738960610,
9      "request_id": "req_ded8ab984ec4bf840f37566c1011c417",
10     "tool_choice": null,
11     "usage": {
12         "total_tokens": 31,
13         "completion_tokens": 18,
14         "prompt_tokens": 13
15     },
16     "seed": 4944116822809979520,
17     "top_p": 1.0,
18     "temperature": 1.0,
19     "presence_penalty": 0.0,
20     "frequency_penalty": 0.0,
21     "system_fingerprint": "fp_50cad350e4",
22     "input_user": null,
23     "service_tier": "default",
24     "tools": null,
25     "metadata": {},
26     "choices": [
27         {
28             "index": 0,
29             "message": {
30                 "content": "Mind of circuits hum, \nLearning patterns in silence— \nFuture",
31                 "role": "assistant",
32                 "tool_calls": null,
33                 "function_call": null
34             },
35             "finish_reason": "stop",
36             "logprobs": null
37         }
38     ],
39     "response_format": null
40 }
41 ],
42 "first_id": "chatcmpl-AyPNinnUqUDYo9SAdA52NobMflmj2",
43 "last_id": "chatcmpl-AyPNinnUqUDYo9SAdA52NobMflmj2",
44 "has_more": false
45 }
```

## Update chat completion

POST [https://api.openai.com/v1/chat/completions/{completion\\_id}](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 ⚡

```
1 curl -X POST https://api.openai.com/v1/chat/completions/chat_abc123 \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -d '{"metadata": {"foo": "bar"}}'
```

### Response

🔗

```
1 {
2   "object": "chat.completion",
3   "id": "chatcmpl-AyPNinnUqUDYo9SAdA52NobMflmj2",
4   "model": "gpt-4o-2024-08-06",
5   "created": 1738960610,
6   "request_id": "req_ded8ab984ec4bf840f37566c1011c417",
7   "tool_choice": null,
8   "usage": {
9     "total_tokens": 31,
10    "completion_tokens": 18,
```

```
11     "prompt_tokens": 13
12 },
13 "seed": 4944116822809979520,
14 "top_p": 1.0,
15 "temperature": 1.0,
16 "presence_penalty": 0.0,
17 "frequency_penalty": 0.0,
18 "system_fingerprint": "fp_50cad350e4",
19 "input_user": null,
20 "service_tier": "default",
21 "tools": null,
22 "metadata": {
23     "foo": "bar"
24 },
25 "choices": [
26     {
27         "index": 0,
28         "message": {
29             "content": "Mind of circuits hum, \nLearning patterns in silence— \nFuture",
30             "role": "assistant",
31             "tool_calls": null,
32             "function_call": null
33         },
34         "finish_reason": "stop",
35         "logprobs": null
36     }
37 ],
38 "response_format": null
39 }
```

## 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 ↴



```
1 curl -X DELETE https://api.openai.com/v1/chat/completions/chat_abc123 \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json"
```

#### Response



```
1 {
2   "object": "chat.completion.deleted",
3   "id": "chatcmpl-AyPNinnUqUDYo9SAdA52NobMflmj2",
4   "deleted": true
5 }
```

## 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.

>Show properties

### **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

The service tier used for processing the request.

**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.

▽ Show properties

OBJECT The chat completion object



```
1  {
2      "id": "chatcmpl-B9MHDbslfkBeAs8l4bebGdF0J6PeG",
3      "object": "chat.completion",
4      "created": 1741570283,
5      "model": "gpt-4o-2024-08-06",
6      "choices": [
7          {
8              "index": 0,
9              "message": {
10                  "role": "assistant",
11                  "content": "The image shows a wooden boardwalk path running through a lush green forest. The path is made of light-colored wood planks and leads into the distance, surrounded by tall trees and dense foliage. The lighting suggests it's daytime, with sunlight filtering through the leaves. There are no people or animals visible on the path.", "refusal": null,
12                  "annotations": []
13              },
14              "logprobs": null,
15              "finish_reason": "stop"
16          }
17      ],
18      "usage": {
19          "prompt_tokens": 1117,
20          "completion_tokens": 46,
21          "total_tokens": 1163,
22          "prompt_tokens_details": {
23              "cached_tokens": 0,
24              "audio_tokens": 0
25          },
26          "completion_tokens_details": {
27              "reasoning_tokens": 0,
28              "audio_tokens": 0,
29              "accepted_prediction_tokens": 0,
30              "rejected_prediction_tokens": 0
31          }
32      },
33      "service_tier": "default",
34  }
```

```
35 "system_fingerprint": "fp_fc9f1d7035"  
36 }
```

# The chat completion list object

An object representing a list of Chat Completions.

## **data** array

An array of chat completion objects.

▼ Show properties

## **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



```
1 {  
2   "object": "list",  
3   "data": [  
4     {  
5       "object": "chat.completion",  
6       "id": "chatcmpl-AyPNinnUqUDYo9SAdA52NobMflmj2",  
7       "model": "gpt-4o-2024-08-06",  
8       "created": 1738960610,  
9       "request_id": "req_ded8ab984ec4bf840f37566c1011c417",  
10      "tool_choice": null,  
11      "usage": {  
12        "total_tokens": 31,  
13        "completion_tokens": 18,  
14        "prompt_tokens": 13  
15      },  
16      "seed": 4944116822809979520,  
17      "top_p": 1.0,  
18      "temperature": 1.0,
```

```
19     "presence_penalty": 0.0,
20     "frequency_penalty": 0.0,
21     "system_fingerprint": "fp_50cad350e4",
22     "input_user": null,
23     "service_tier": "default",
24     "tools": null,
25     "metadata": {},
26     "choices": [
27       {
28         "index": 0,
29         "message": {
30           "content": "Mind of circuits hum, \nLearning patterns in silence— \nFuture\n  \"role\": \"assistant\",",
31           "role": "assistant",
32           "tool_calls": null,
33           "function_call": null
34         },
35         "finish_reason": "stop",
36         "logprobs": null
37       }
38     ],
39     "response_format": null
40   }
41 ],
42   "first_id": "chatmpl-AyPNinnUqUDYo9SAdA52NobMflmj2",
43   "last_id": "chatmpl-AyPNinnUqUDYo9SAdA52NobMflmj2",
44   "has_more": false
45 }
```

## The chat completion message list object

An object representing a list of chat completion messages.

### **data** array

An array of chat completion message objects.

∨ Show properties

### **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



```
1  {
2    "object": "list",
3    "data": [
4      {
5        "id": "chatmpl-AyPNinnUqUDYo9SAdA52NobMflmj2-0",
6        "role": "user",
7        "content": "write a haiku about ai",
8        "name": null,
9        "content_parts": null
10      }
11    ],
12    "first_id": "chatmpl-AyPNinnUqUDYo9SAdA52NobMflmj2-0",
13    "last_id": "chatmpl-AyPNinnUqUDYo9SAdA52NobMflmj2-0",
14    "has_more": false
15 }
```

## 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}`.

▼ Show properties

### 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

The service tier used for processing the request.

**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

An optional field that will only be present when you set `stream_options: {"include_usage": true}` in your request. When present, it contains a null value except for the last chunk which contains the token usage statistics for the entire request.

▼ Show properties

OBJECT The chat completion chunk object



```
1 {"id":"chatcmpl-123","object":"chat.completion.chunk","created":1694268190,"model":"gp
2
3 {"id":"chatcmpl-123","object":"chat.completion.chunk","created":1694268190,"model":"gp
4
5 ....
6
7 {"id":"chatcmpl-123","object":"chat.completion.chunk","created":1694268190,"model":"gp
```

## 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](#) or [tts-1-hd](#)

### **voice** string Required

The voice to use when generating the audio. Supported voices are [alloy](#), [ash](#), [coral](#), [echo](#), [fable](#), [onyx](#), [nova](#), [sage](#) and [shimmer](#). Previews of the voices are available in the [Text to speech guide](#).

### **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.

## Returns

The audio file content.

### Example request

curl [▼](#) [Copy](#)

```
1 curl https://api.openai.com/v1/audio/speech \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -d '{
5     "model": "tts-1",
6     "input": "The quick brown fox jumped over the lazy dog.",
7     "voice": "alloy"
```

```
8  }' \
9  --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, mpg, 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.

### **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`.

### **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](#) or a [verbose transcription object](#).

[Default](#) [Word timestamps](#) [Segment timestamps](#)

### Example request

curl ⌂

```
1 curl https://api.openai.com/v1/audio/transcriptions \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: multipart/form-data" \
4   -F file="@/path/to/file/audio.mp3" \
5   -F model="whisper-1"
```

### Response

📋

```
1 {
2   "text": "Imagine the wildest idea that you've ever had, and you're curious about how
3 }
```

## 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) to 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 ↻

```
1 curl https://api.openai.com/v1/audio/translations \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: multipart/form-data" \
4   -F file="@/path/to/file/german.m4a" \
5   -F model="whisper-1"
```

## Response

🔗

```
1 {
2   "text": "Hello, my name is Wolfgang and I come from Germany. Where are you heading to?"
3 }
```

# The transcription object (JSON)

Represents a transcription response returned by model, based on the provided input.

**text** string

The transcribed text.

## OBJECT The transcription object (JSON)



```
1 {  
2   "text": "Imagine the wildest idea that you've ever had, and you're curious about how  
3 }
```

# 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.

✓ Show properties

**text** string

The transcribed text.

**words** array

Extracted words and their corresponding timestamps.

✓ Show properties

## OBJECT The transcription object (Verbose JSON)



```
1 {  
2   "task": "transcribe",  
3   "language": "english",  
4   "duration": 8.470000267028809,  
5   "text": "The beach was a popular spot on a hot summer day. People were swimming in  
6   "segments": [  
7     {  
8       "id": 0,  
9       "seek": 0,  
10      "start": 0.0,
```

```
11     "end": 3.319999933242798,  
12     "text": " The beach was a popular spot on a hot summer day.",  
13     "tokens": [  
14         50364, 440, 7534, 390, 257, 3743, 4008, 322, 257, 2368, 4266, 786, 13, 50530  
15     ],  
16     "temperature": 0.0,  
17     "avg_logprob": -0.2860786020755768,  
18     "compression_ratio": 1.2363636493682861,  
19     "no_speech_prob": 0.00985979475080967  
20 },  
21 ...  
22 ]  
23 }
```

## 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 `text-embedding-ada-002`), cannot be an empty string, and any array must be 2048 dimensions or less. [Example Python code](#) for counting tokens. Some models may also impose a limit on total number of tokens summed across inputs.

▼ Show possible types

#### **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

node.js ▾ 

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const embedding = await openai.embeddings.create({
7     model: "text-embedding-ada-002",
8     input: "The quick brown fox jumped over the lazy dog",
9     encoding_format: "float",
10  });
11
12  console.log(embedding);
13 }
14
15 main();
```

### Response



```
1 {
2   "object": "list",
3   "data": [
4     {
5       "object": "embedding",
6       "embedding": [
7         0.0023064255,
8         -0.009327292,
9         .... (1536 floats total for ada-002)
10        -0.0028842222,
```

```

11     ],
12     "index": 0
13   }
14 ],
15   "model": "text-embedding-ada-002",
16   "usage": {
17     "prompt_tokens": 8,
18     "total_tokens": 8
19   }
20 }
```

## 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

```

1  {
2    "object": "embedding",
3    "embedding": [
4      0.0023064255,
5      -0.009327292,
6      .... (1536 floats total for ada-002)
7      -0.0028842222,
8    ],
9    "index": 0
10 }
```

## 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](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.

▼ Show properties

---

### **integrations** array or null Optional

A list of integrations to enable for your fine-tuning job.

▼ Show properties

---

### **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.

▼ Show properties

---

**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](#)   [Validation file](#)   [DPO](#)   [W&B Integration](#)

Example request

node.js ▾ 

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
```

```
6  const fineTune = await openai.fineTuning.jobs.create({
7    training_file: "file-abc123"
8  });
9
10 console.log(fineTune);
11 }
12
13 main();
```

## Response



```
1  {
2    "object": "fine_tuning.job",
3    "id": "ftjob-abc123",
4    "model": "gpt-4o-mini-2024-07-18",
5    "created_at": 1721764800,
6    "fine_tuned_model": null,
7    "organization_id": "org-123",
8    "result_files": [],
9    "status": "queued",
10   "validation_file": null,
11   "training_file": "file-abc123",
12   "method": {
13     "type": "supervised",
14     "supervised": {
15       "hyperparameters": {
16         "batch_size": "auto",
17         "learning_rate_multiplier": "auto",
18         "n_epochs": "auto",
19       }
20     }
21   },
22   "metadata": null
23 }
```

## List fine-tuning jobs

GET [https://api.openai.com/v1/fine\\_tuning/jobs](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

node.js ↗

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const list = await openai.fineTuning.jobs.list();
7
8   for await (const fineTune of list) {
9     console.log(fineTune);
10  }
11 }
12
13 main();
```

## Response

↗

```
1 {
2   "object": "list",
3   "data": [
4     {
5       "object": "fine_tuning.job",
6       "id": "ftjob-abc123",
7       "model": "gpt-4o-mini-2024-07-18",
8       "created_at": 1721764800,
9       "fine_tuned_model": null,
10      "organization_id": "org-123",
```

```
11     "result_files": [],
12     "status": "queued",
13     "validation_file": null,
14     "training_file": "file-abc123",
15     "metadata": {
16       "key": "value"
17     }
18   },
19   { ... },
20   { ... }
21 ], "has_more": true
22 }
```

## List fine-tuning events

GET [https://api.openai.com/v1/fine\\_tuning/jobs/{fine\\_tuning\\_job\\_id}/events](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

node.js ▼ 

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const list = await openai.fineTuning.list_events(id="ftjob-abc123", limit=2);
7
8   for await (const fineTune of list) {
9     console.log(fineTune);
10  }
11 }
12
13 main();
```

## Response



```
1 {
2   "object": "list",
3   "data": [
4     {
5       "object": "fine_tuning.job.event",
6       "id": "ft-event-ddTJfwuMVpfLXse00Am0Gqjm",
7       "created_at": 1721764800,
8       "level": "info",
9       "message": "Fine tuning job successfully completed",
10      "data": null,
11      "type": "message"
12    },
13    {
14      "object": "fine_tuning.job.event",
15      "id": "ft-event-tyiGuB72evQncpH87xe505Sv",
16      "created_at": 1721764800,
17      "level": "info",
18      "message": "New fine-tuned model created: ft:gpt-4o-mini:openai::7p4lURel",
19      "data": null,
20      "type": "message"
21    }
22  ],
23  "has_more": true
24 }
```

## List fine-tuning checkpoints

GET [https://api.openai.com/v1/fine\\_tuning/jobs/{fine\\_tuning\\_job\\_id}/checkpoints](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 ⌂ 

```
1 curl https://api.openai.com/v1/fine_tuning/jobs/ftjob-abc123/checkpoints \
2   -H "Authorization: Bearer $OPENAI_API_KEY"
```

#### Response



```
1 {
2   "object": "list"
3   "data": [
4     {
5       "object": "fine_tuning.job.checkpoint",
6       "id": "ftckpt_zc4Q7MP6XxulcVzj4MZdwsAB",
7       "created_at": 1721764867,
8       "fine_tuned_model_checkpoint": "ft:gpt-4o-mini-2024-07-18:my-org:custom-suffix:",
9       "metrics": {
10         "full_valid_loss": 0.134,
11         "full_valid_mean_token_accuracy": 0.874
12       },
13       "fine_tuning_job_id": "ftjob-abc123",
14       "step_number": 2000,
15     },
16   ],
17 }
```

```
16      {
17        "object": "fine_tuning.job.checkpoint",
18        "id": "ftckpt_enQCFm0TGj3syEpYVhBRLTSy",
19        "created_at": 1721764800,
20        "fine_tuned_model_checkpoint": "ft:gpt-4o-mini-2024-07-18:my-org:custom-suffix:",
21        "metrics": {
22          "full_valid_loss": 0.167,
23          "full_valid_mean_token_accuracy": 0.781
24        },
25        "fine_tuning_job_id": "ftjob-abc123",
26        "step_number": 1000,
27      },
28    ],
29    "first_id": "ftckpt_zc4Q7MP6XxulcVzj4MZdwsAB",
30    "last_id": "ftckpt_enQCFm0TGj3syEpYVhBRLTSy",
31    "has_more": true
32 }
```

## List checkpoint permissions

```
GET https://api.openai.com/v1/fine_tuning/checkpoints/{fine_tuned_model_checkpoint}/permissions
```

List 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.

## Returns

A list of fine-tuned model checkpoint [permission objects](#) for a fine-tuned model checkpoint.

### Example request

[curl ↴](#)

```
1 curl https://api.openai.com/v1/fine_tuning/checkpoints/ft:gpt-4o-mini-2024-07-18:org:w
2   -H "Authorization: Bearer $OPENAI_API_KEY"
```

### Response

[📋](#)

```
1 {
2   "object": "list",
3   "data": [
4     {
5       "object": "checkpoint.permission",
6       "id": "cp_zc4Q7MP6XxulcVzj4MZdwsAB",
7       "created_at": 1721764867,
8       "project_id": "proj_abGMw1llN8IrBb6SvvY5A1iH"
9     },
10    {
11      "object": "checkpoint.permission",
12      "id": "cp_enQCFm0TGj3syEpYVhBRLTSy",
13      "created_at": 1721764800,
14      "project_id": "proj_iqGMw1llN8IrBb6SvvY5A1oF"
15    },
16  ],
17  "first_id": "cp_zc4Q7MP6XxulcVzj4MZdwsAB",
18  "last_id": "cp_enQCFm0TGj3syEpYVhBRLTSy",
19  "has_more": false
20 }
```

## Create checkpoint permissions

```
POST https://api.openai.com/v1/fine_tuning/checkpoints/{fine_tuned_model_checkpoint}/per
missions
```

Create a fine-tuned model checkpoint permission to grant access to a project. 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 ↴

```
1 curl https://api.openai.com/v1/fine_tuning/checkpoints/ft:gpt-4o-mini-2024-07-18:org:w
2   -H "Authorization: Bearer $OPENAI_API_KEY"
3   -d '{"project_ids": ["proj_abGMw1llN8IrBb6SvvY5A1iH"]}'
```

## Response

copy

```
1 {
2   "object": "list",
3   "data": [
4     {
5       "object": "checkpoint.permission",
6       "id": "cp_zc4Q7MP6XxulcVzj4MZdwsAB",
7       "created_at": 1721764867,
8       "project_id": "proj_abGMw1llN8IrBb6SvvY5A1iH"
9     }
10   ],
11   "first_id": "cp_zc4Q7MP6XxulcVzj4MZdwsAB",
12   "last_id": "cp_zc4Q7MP6XxulcVzj4MZdwsAB",
13   "has_more": false
14 }
```

# Delete checkpoint permission

```
DELETE https://api.openai.com/v1/fine_tuning/checkpoints/{fine_tuned_model_checkpoint}/permissions
```

Delete a fine-tuned model checkpoint permission.

### 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 ⚡ 

```
1 curl https://api.openai.com/v1/fine_tuning/checkpoints/ft:gpt-4o-mini-2024-07-18:org:w  
2   -H "Authorization: Bearer $OPENAI_API_KEY"
```

#### Response



```
1 {  
2   "object": "checkpoint.permission",  
3   "id": "cp_zc4Q7MP6XulcVzj4MZdwsAB",  
4   "deleted": true  
5 }
```

## 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

node.js ▼ 

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const fineTune = await openai.fineTuning.jobs.retrieve("ftjob-abc123");
7
8   console.log(fineTune);
9 }
10
11 main();
```

### Response



```
1 {
2   "object": "fine_tuning.job",
3   "id": "ftjob-abc123",
4   "model": "davinci-002",
5   "created_at": 1692661014,
6   "finished_at": 1692661190,
7   "fine_tuned_model": "ft:davinci-002:my-org:custom_suffix:7q8mpxmy",
8   "organization_id": "org-123",
9   "result_files": [
10     "file-abc123"
11   ],
12   "status": "succeeded",
13   "validation_file": null,
14   "training_file": "file-abc123",
15   "hyperparameters": {
16     "n_epochs": 4,
17     "batch_size": 1,
18     "learning_rate_multiplier": 1.0
19   },
20 }
```

```
20 "trained_tokens": 5768,  
21 "integrations": [],  
22 "seed": 0,  
23 "estimated_finish": 0,  
24 "method": {  
25     "type": "supervised",  
26     "supervised": {  
27         "hyperparameters": {  
28             "n_epochs": 4,  
29             "batch_size": 1,  
30             "learning_rate_multiplier": 1.0  
31         }  
32     }  
33 }  
34 }
```

## 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

node.js ↗

```
1 import OpenAI from "openai";  
2  
3 const openai = new OpenAI();  
4  
5 async function main() {  
6     const fineTune = await openai.fineTuning.jobs.cancel("ftjob-abc123");  
7  
8     console.log(fineTune);
```

```
9 }
10 main();
```

## Response



```
1 {
2   "object": "fine_tuning.job",
3   "id": "ftjob-abc123",
4   "model": "gpt-4o-mini-2024-07-18",
5   "created_at": 1721764800,
6   "fine_tuned_model": null,
7   "organization_id": "org-123",
8   "result_files": [],
9   "status": "cancelled",
10  "validation_file": "file-abc123",
11  "training_file": "file-abc123"
12 }
```

## 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.

### **functions** Deprecated array

A list of functions the model may generate JSON inputs for.

∨ Show properties

### **messages** array

∨ Show possible types

### **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.

∨ Show properties

OBJECT Training format for chat models using the supervised method



```
1  {
2    "messages": [
3      { "role": "user", "content": "What is the weather in San Francisco?" },
4      {
5        "role": "assistant",
6        "tool_calls": [
7          {
8            "id": "call_id",
9            "type": "function",
10           "function": {
11             "name": "get_current_weather",
12             "arguments": "{\"location\": \"San Francisco, USA\", \"format\": \"celsius\"}"
13           }
14         }
15       ]
16     }
17   ],
18   "parallel_tool_calls": false,
19   "tools": [
20     {
21       "type": "function",
22       "function": {
23         "name": "get_current_weather",
24         "description": "Get the current weather",
25         "parameters": {
26           "type": "object",
27           "properties": {
28             "location": {
29               "type": "string",
30               "description": "The city and country, eg. San Francisco, USA"
31             },
32             "format": { "type": "string", "enum": ["celsius", "fahrenheit"] }
33           },
34           "required": ["location", "format"]
35         }
36       }
37     }
38   ]
39 }
```

## 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** object

✓ Show properties

**non\_preferred\_completion** array

The non-preferred completion message for the output.

✓ Show possible types

**preferred\_completion** array

The preferred completion message for the output.

✓ Show possible types

OBJECT Training format for chat models using the preference method



```
1  {
2    "input": {
3      "messages": [
4        { "role": "user", "content": "What is the weather in San Francisco?" }
5      ]
6    },
7    "preferred_completion": [
8      {
9        "role": "assistant",
10       "content": "The weather in San Francisco is 70 degrees Fahrenheit."
11     }
12   ],
13   "non_preferred_completion": [
14     {
15       "role": "assistant",
16       "content": "The weather in San Francisco is 21 degrees Celsius."
17     }
18   ]
19 }
```

## Training format for completions models

The per-line training example of a fine-tuning input file for completions models

**completion** string

The desired completion for this training example.

**prompt** string

The input prompt for this training example.

OBJECT Training format for completions models



```
1 {
2   "prompt": "What is the answer to 2+2",
3   "completion": "4"
4 }
```

## 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.

▼ Show properties

**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.

▼ Show properties

**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.

✓ Show possible types

**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.

✓ Show properties

**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



```
1  {
2    "object": "fine_tuning.job",
3    "id": "ftjob-abc123",
4    "model": "davinci-002",
5    "created_at": 1692661014,
6    "finished_at": 1692661190,
7    "fine_tuned_model": "ft:davinci-002:my-org:custom_suffix:7q8mpxmy",
8    "organization_id": "org-123",
9    "result_files": [
10      "file-abc123"
11    ],
12    "status": "succeeded",
13    "validation_file": null,
14    "training_file": "file-abc123",
15    "hyperparameters": {
16      "n_epochs": 4,
17      "batch_size": 1,
18      "learning_rate_multiplier": 1.0
19    },
20    "trained_tokens": 5768,
21    "integrations": [],
22    "seed": 0,
23    "estimated_finish": 0,
24    "method": {
25      "type": "supervised",
26      "supervised": {
27        "hyperparameters": {
28          "n_epochs": 4,
29          "batch_size": 1,
30          "learning_rate_multiplier": 1.0
31        }
32      }
33    },
34    "metadata": {
35      "key": "value"
36    }
37 }
```

## 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



```
1 {
2   "object": "fine_tuning.job.event",
3   "id": "ftevent-abc123"
4   "created_at": 1677610602,
5   "level": "info",
6   "message": "Created fine-tuning job",
7   "data": {},
8   "type": "message"
9 }
```

## 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.

✓ Show properties

**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



```
1  {
2    "object": "fine_tuning.job.checkpoint",
3    "id": "ftckpt_qtZ5Gyk4BLq1SfLFWp3Rt03P",
4    "created_at": 1712211699,
5    "fine_tuned_model_checkpoint": "ft:gpt-4o-mini-2024-07-18:my-org:custom_suffix:9ABe",
6    "fine_tuning_job_id": "ftjob-fpbNQ3H1GrMehXRF8c097xTN",
7    "metrics": {
8      "step": 88,
9      "train_loss": 0.478,
10     "train_mean_token_accuracy": 0.924,
11     "valid_loss": 10.112,
12     "valid_mean_token_accuracy": 0.145,
13     "full_valid_loss": 0.567,
14     "full_valid_mean_token_accuracy": 0.944
15   },
16   "step_number": 88
17 }
```

# 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



```
1 {
2   "object": "checkpoint.permission",
3   "id": "cp_zc4Q7MP6XxulcVzj4MZdwsAB",
4   "created_at": 1712211699,
5   "project_id": "proj_abGMw1llN8IrBb6SvvY5A1iH"
6 }
```

# 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/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 ↴ 

```
1 curl https://api.openai.com/v1/batches \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -d '{
5     "input_file_id": "file-abc123",
6     "endpoint": "/v1/chat/completions",
7     "completion_window": "24h"
8   }'
```

### Response



```
1  {
2    "id": "batch_abc123",
3    "object": "batch",
4    "endpoint": "/v1/chat/completions",
5    "errors": null,
6    "input_file_id": "file-abc123",
7    "completion_window": "24h",
8    "status": "validating",
9    "output_file_id": null,
10   "error_file_id": null,
11   "created_at": 1711471533,
12   "in_progress_at": null,
13   "expires_at": null,
14   "finalizing_at": null,
15   "completed_at": null,
16   "failed_at": null,
17   "expired_at": null,
18   "cancelling_at": null,
19   "cancelled_at": null,
20   "request_counts": {
21     "total": 0,
22     "completed": 0,
23     "failed": 0
24   },
25   "metadata": {
26     "customer_id": "user_123456789",
27     "batch_description": "Nightly eval job",
28   }
29 }
```

## 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 ↴



```
1 curl https://api.openai.com/v1/batches/batch_abc123 \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
```

#### Response



```
1 {
2   "id": "batch_abc123",
3   "object": "batch",
4   "endpoint": "/v1/completions",
5   "errors": null,
6   "input_file_id": "file-abc123",
7   "completion_window": "24h",
8   "status": "completed",
9   "output_file_id": "file-cvaTdG",
10  "error_file_id": "file-H0WS94",
11  "created_at": 1711471533,
12  "in_progress_at": 1711471538,
13  "expires_at": 1711557933,
14  "finalizing_at": 1711493133,
15  "completed_at": 1711493163,
16  "failed_at": null,
17  "expired_at": null,
18  "cancelling_at": null,
19  "cancelled_at": null,
20  "request_counts": {
21    "total": 100,
22    "completed": 95,
23    "failed": 5
24  },
25  "metadata": {
26    "customer_id": "user_123456789",
27    "batch_description": "Nightly eval job",
28  }
29 }
```

## 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 ⚡ 

```
1 curl https://api.openai.com/v1/batches/batch_abc123/cancel \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -X POST
```

### Response



```
1 {
2   "id": "batch_abc123",
3   "object": "batch",
4   "endpoint": "/v1/chat/completions",
5   "errors": null,
6   "input_file_id": "file-abc123",
7   "completion_window": "24h",
8   "status": "cancelling",
9   "output_file_id": null,
10  "error_file_id": null,
11  "created_at": 1711471533,
12  "in_progress_at": 1711471538,
13  "expires_at": 1711557933,
14  "finalizing_at": null,
15  "completed_at": null,
16  "failed_at": null,
17  "expired_at": null,
18  "cancelling_at": 1711475133,
```

```
19 "cancelled_at": null,  
20 "request_counts": {  
21   "total": 100,  
22   "completed": 23,  
23   "failed": 1  
24 },  
25 "metadata": {  
26   "customer_id": "user_123456789",  
27   "batch_description": "Nightly eval job",  
28 }  
29 }
```

## 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 ⚡ 🗂️

```
1 curl https://api.openai.com/v1/batches?limit=2 \  
2   -H "Authorization: Bearer $OPENAI_API_KEY" \  
3   -H "Content-Type: application/json"
```

#### Response

🗂️

```
1  {
2    "object": "list",
3    "data": [
4      {
5        "id": "batch_abc123",
6        "object": "batch",
7        "endpoint": "/v1/chat/completions",
8        "errors": null,
9        "input_file_id": "file-abc123",
10       "completion_window": "24h",
11       "status": "completed",
12       "output_file_id": "file-cvaTdG",
13       "error_file_id": "file-H0WS94",
14       "created_at": 1711471533,
15       "in_progress_at": 1711471538,
16       "expires_at": 1711557933,
17       "finalizing_at": 1711493133,
18       "completed_at": 1711493163,
19       "failed_at": null,
20       "expired_at": null,
21       "cancelling_at": null,
22       "cancelled_at": null,
23       "request_counts": {
24         "total": 100,
25         "completed": 95,
26         "failed": 5
27       },
28       "metadata": {
29         "customer_id": "user_123456789",
30         "batch_description": "Nightly job",
31       }
32     },
33     { ... },
34   ],
35   "first_id": "batch_abc123",
36   "last_id": "batch_abc456",
37   "has_more": true
38 }
```

## 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

▽ Show properties

**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.

▽ Show properties

**status** string

The current status of the batch.

OBJECT The batch object



```
1  {
2    "id": "batch_abc123",
3    "object": "batch",
4    "endpoint": "/v1/completions",
5    "errors": null,
6    "input_file_id": "file-abc123",
7    "completion_window": "24h",
8    "status": "completed",
9    "output_file_id": "file-cvaTdG",
10   "error_file_id": "file-H0WS94",
11   "created_at": 1711471533,
12   "in_progress_at": 1711471538,
13   "expires_at": 1711557933,
14   "finalizing_at": 1711493133,
15   "completed_at": 1711493163,
16   "failed_at": null,
17   "expired_at": null,
18   "cancelling_at": null,
19   "cancelled_at": null,
20   "request_counts": {
21     "total": 100,
```

```
22     "completed": 95,  
23     "failed": 5  
24   },  
25   "metadata": {  
26     "customer_id": "user_123456789",  
27     "batch_description": "Nightly eval job",  
28   }  
29 }
```

## 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": {"mo
```

## 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.

✓ Show properties

**id** string**response** object or null

▼ Show properties

OBJECT The request output object



{ "id": "batch\_req\_wnaDys", "custom\_id": "request-2", "response": { "status\_code": 200, "r... } }

## 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

node.js ↗

```
1 import fs from "fs";
2 import OpenAI from "openai";
3
4 const openai = new OpenAI();
5
6 async function main() {
7   const file = await openai.files.create({
8     file: fs.createReadStream("mydata.jsonl"),
9     purpose: "fine-tune",
10  });
11
12  console.log(file);
13 }
14
15 main();
```

### Response

↗

```
1 {
2   "id": "file-abc123",
3   "object": "file",
4   "bytes": 120000,
5   "created_at": 1677610602,
6   "filename": "mydata.jsonl",
7   "purpose": "fine-tune",
8 }
```

## 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

node.js ⚡

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const list = await openai.files.list();
7
8   for await (const file of list) {
9     console.log(file);
10  }
11 }
12
13 main();
```

## Response



```
1  {
2    "data": [
3      {
4        "id": "file-abc123",
5        "object": "file",
6        "bytes": 175,
7        "created_at": 1613677385,
8        "filename": "salesOverview.pdf",
9        "purpose": "assistants",
10      },
11      {
12        "id": "file-abc123",
13        "object": "file",
14        "bytes": 140,
15        "created_at": 1613779121,
16        "filename": "puppy.jsonl",
17        "purpose": "fine-tune",
18      }
19    ],
20    "object": "list"
21 }
```

## 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

node.js



```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const file = await openai.files.retrieve("file-abc123");
7
8   console.log(file);
9 }
10
11 main();
```

## Response



```
1 {
2   "id": "file-abc123",
3   "object": "file",
4   "bytes": 120000,
5   "created_at": 1677610602,
6   "filename": "mydata.jsonl",
7   "purpose": "fine-tune",
8 }
```

## 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

node.js

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const file = await openai.files.del("file-abc123");
7
8   console.log(file);
9 }
10
11 main();
```

## Response



```
1 {
2   "id": "file-abc123",
3   "object": "file",
4   "deleted": true
5 }
```

# 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

node.js

```
1 import OpenAI from "openai";
2
```

```
3 const openai = new OpenAI();
4
5 async function main() {
6   const file = await openai.files.content("file-abc123");
7
8   console.log(file);
9 }
10
11 main();
```

## 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` and `vision`.

---

**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

```
1 {
2   "id": "file-abc123",
3   "object": "file",
4   "bytes": 120000,
5   "created_at": 1677610602,
6   "expires_at": 1680202602,
7   "filename": "salesOverview.pdf",
8   "purpose": "assistants",
9 }
```



# 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 ⚡ 🗂️

```
1 curl https://api.openai.com/v1/uploads \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -d '{
4     "purpose": "fine-tune",
5     "filename": "training_examples.jsonl",
6     "bytes": 2147483648,
7     "mime_type": "text/jsonl"
8   }'
```

### Response

🗂️

```
1 {
2   "id": "upload_abc123",
3   "object": "upload",
4   "bytes": 2147483648,
5   "created_at": 1719184911,
6   "filename": "training_examples.jsonl",
7   "purpose": "fine-tune",
8   "status": "pending",
9   "expires_at": 1719127296
10 }
```

# 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 ⌂

```
1 curl https://api.openai.com/v1/uploads/upload_abc123/parts
2 -F data="aHR0cHM6Ly9hcGkub3BlbmFpLmNvbS92MS91cGxvYWRz..."
```

### Response

⌂

```
1 {
2   "id": "part_def456",
3   "object": "upload.part",
4   "created_at": 1719185911,
```

```
5   "upload_id": "upload_abc123"  
6 }
```

# 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 ⚡ 

```
1 curl https://api.openai.com/v1/uploads/upload_abc123/complete  
2 -d '{'
```

```
3     "part_ids": ["part_def456", "part_ghi789"]  
4 }'
```

## Response



```
1 {  
2   "id": "upload_abc123",  
3   "object": "upload",  
4   "bytes": 2147483648,  
5   "created_at": 1719184911,  
6   "filename": "training_examples.jsonl",  
7   "purpose": "fine-tune",  
8   "status": "completed",  
9   "expires_at": 1719127296,  
10  "file": {  
11    "id": "file-xyz321",  
12    "object": "file",  
13    "bytes": 2147483648,  
14    "created_at": 1719186911,  
15    "filename": "training_examples.jsonl",  
16    "purpose": "fine-tune",  
17  }  
18 }
```

## 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



```
1  {
2    "id": "upload_abc123",
3    "object": "upload",
4    "bytes": 2147483648,
5    "created_at": 1719184911,
6    "filename": "training_examples.jsonl",
7    "purpose": "fine-tune",
8    "status": "cancelled",
9    "expires_at": 1719127296
10 }
```

# 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



```
1  {
2    "id": "upload_abc123",
3    "object": "upload",
4    "bytes": 2147483648,
5    "created_at": 1719184911,
6    "filename": "training_examples.jsonl",
7    "purpose": "fine-tune",
8    "status": "completed",
9    "expires_at": 1719127296,
10   "file": {
11     "id": "file-xyz321",
12     "object": "file",
13     "bytes": 2147483648,
14     "created_at": 1719186911,
15     "filename": "training_examples.jsonl",
16     "purpose": "fine-tune",
17   }
18 }
```

## 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



```
1 {
2   "id": "part_def456",
3   "object": "upload.part",
4   "created_at": 1719186911,
5   "upload_id": "upload_abc123"
6 }
```

## 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.

### Request body

**prompt** string Required

A text description of the desired image(s). The maximum length is 1000 characters for `dall-e-2` and 4000 characters for `dall-e-3`.

**model** string Optional Defaults to dall-e-2

The model to use for image generation.

**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.

**quality** string Optional Defaults to standard

The quality of the image that will be generated. `hd` creates images with finer details and greater consistency across the image. This param is only supported for `dall-e-3`.

**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 1024x1024

The size of the generated images. Must be one of `256x256`, `512x512`, or `1024x1024` for `dall-e-2`. Must be one of `1024x1024`, `1792x1024`, or `1024x1792` for `dall-e-3` models.

**style** string or null Optional Defaults to vivid

The style of the generated images. 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. This param is only supported for `dall-e-3`.

**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

node.js ↗

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const image = await openai.images.generate({ model: "dall-e-3", prompt: "A cute bab
7
8   console.log(image.data);
9 }
10 main();
```

## Response

↗

```
1 {
2   "created": 1589478378,
3   "data": [
4     {
5       "url": "https://..."
6     },
7   ]
```

```
7      {
8        "url": "https://..."
9      }
10     ]
11 }
```

# Create image edit

POST <https://api.openai.com/v1/images/edits>

Creates an edited or extended image given an original image and a prompt.

## Request body

### **image** file Required

The image to edit. Must be a valid PNG file, less than 4MB, and square. If mask is not provided, image must have transparency, which will be used as the mask.

### **prompt** string Required

A text description of the desired image(s). The maximum length is 1000 characters.

### **mask** file Optional

An additional image whose fully transparent areas (e.g. where alpha is zero) indicate where `image` should be edited. Must be a valid PNG file, less than 4MB, and have the same dimensions as `image`.

### **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 1024x1024

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

node.js 

```
1 import fs from "fs";
2 import OpenAI from "openai";
3
4 const openai = new OpenAI();
5
6 async function main() {
7   const image = await openai.images.edit({
8     image: fs.createReadStream("otter.png"),
9     mask: fs.createReadStream("mask.png"),
10    prompt: "A cute baby sea otter wearing a beret",
11  });
12
13  console.log(image.data);
14 }
15 main();
```

### Response



```
1 {
2   "created": 1589478378,
3   "data": [
4     {
5       "url": "https://..."
6     },
7     {
8       "url": "https://..."
9     }
10   ]
11 }
```

## Create image variation

POST <https://api.openai.com/v1/images/variations>

Creates a variation of a given image.

## 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. For `dall-e-3`, only `n=1` is supported.

### 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 1024x1024

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

node.js



```
1 import fs from "fs";
2 import OpenAI from "openai";
3
4 const openai = new OpenAI();
5
6 async function main() {
7   const image = await openai.images.createVariation({
8     image: fs.createReadStream("otter.png"),
9   });
10
11   console.log(image.data);
```

```
12 }
13 main();
```

## Response



```
1 {
2   "created": 1589478378,
3   "data": [
4     {
5       "url": "https://..."
6     },
7     {
8       "url": "https://..."
9     }
10   ]
11 }
```

## The image object

Represents the url or the content of an image generated by the OpenAI API.

**b64\_json** string

The base64-encoded JSON of the generated image, if `response_format` is `b64_json`.

**revised\_prompt** string

The prompt that was used to generate the image, if there was any revision to the prompt.

**url** string

The URL of the generated image, if `response_format` is `url` (default).

## OBJECT The image object



```
1 {
2   "url": "...",
3   "revised_prompt": "..."
4 }
```

## 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

node.js ↗

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const list = await openai.models.list();
7
8   for await (const model of list) {
9     console.log(model);
10  }
11 }
12 main();
```

#### Response

↗

```
1 {
2   "object": "list",
3   "data": [
4     {
5       "id": "model-id-0",
6       "object": "model",
7       "created": 1686935002,
8       "owned_by": "organization-owner"
9     },
10    {
11      "id": "model-id-1",
```

```
12     "object": "model",
13     "created": 1686935002,
14     "owned_by": "organization-owner",
15   },
16   {
17     "id": "model-id-2",
18     "object": "model",
19     "created": 1686935002,
20     "owned_by": "openai"
21   },
22 ],
23 "object": "list"
24 }
```

## 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-4o ⚡ node.js ⚡ 

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const model = await openai.models.retrieve("gpt-4o");
7
8   console.log(model);
9 }
```

```
10  
11 main();
```

## Response



```
1 {  
2   "id": "gpt-4o",  
3   "object": "model",  
4   "created": 1686935002,  
5   "owned_by": "openai"  
6 }
```

## 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

node.js ⚡

```
1 import OpenAI from "openai";  
2  
3 const openai = new OpenAI();  
4  
5 async function main() {  
6   const model = await openai.models.del("ft:gpt-4o-mini:acemeco:suffix:abc123");  
7  
8   console.log(model);  
9 }  
10 main();
```

## Response



```
1 {
2   "id": "ft:gpt-4o-mini:acemeco:suffix:abc123",
3   "object": "model",
4   "deleted": true
5 }
```

## 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



```
1 {
2   "id": "gpt-4o",
3   "object": "model",
4   "created": 1686935002,
5   "owned_by": "openai"
6 }
```

## 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.

✓ Show possible types

### **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

node.js Copy

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const moderation = await openai.moderations.create({ input: "I want to kill them."
7
8   console.log(moderation);
9 }
10 main();
```

### Response

Copy

```
1  {
2    "id": "modr-AB8Cj0Tu2jiq12hp1AQPfeqFWaORR",
3    "model": "text-moderation-007",
4    "results": [
5      {
6        "flagged": true,
7        "categories": {
8          "sexual": false,
9          "hate": false,
10         "harassment": true,
11         "self-harm": false,
12         "sexual/minors": false,
13         "hate/threatening": false,
14         "violence/graphic": false,
15         "self-harm/intent": false,
16         "self-harm/instructions": false,
17         "harassment/threatening": true,
18         "violence": true
19       },
20       "category_scores": {
21         "sexual": 0.000011726012417057063,
22         "hate": 0.22706663608551025,
23         "harassment": 0.5215635299682617,
24         "self-harm": 2.227119921371923e-6,
25         "sexual/minors": 7.107352217872176e-8,
26         "hate/threatening": 0.023547329008579254,
27         "violence/graphic": 0.00003391829886822961,
28         "self-harm/intent": 1.646940972932498e-6,
29         "self-harm/instructions": 1.1198755256458526e-9,
30         "harassment/threatening": 0.5694745779037476,
31         "violence": 0.9971134662628174
32       }
33     }
34   ]
35 }
```

## 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.

✓ Show properties

OBJECT The moderation object



```
1  {
2      "id": "modr-0d9740456c391e43c445bf0f010940c7",
3      "model": "omni-moderation-latest",
4      "results": [
5          {
6              "flagged": true,
7              "categories": {
8                  "harassment": true,
9                  "harassment/threatening": true,
10                 "sexual": false,
11                 "hate": false,
12                 "hate/threatening": false,
13                 "illicit": false,
14                 "illicit/violent": false,
15                 "self-harm/intent": false,
16                 "self-harm/instructions": false,
17                 "self-harm": false,
18                 "sexual/minors": false,
19                 "violence": true,
20                 "violence/graphic": true
21             },
22             "category_scores": {
23                 "harassment": 0.8189693396524255,
24                 "harassment/threatening": 0.804985420696006,
25                 "sexual": 1.573112165348997e-6,
26                 "hate": 0.007562942636942845,
27                 "hate/threatening": 0.004208854591835476,
28                 "illicit": 0.030535955153511665,
29                 "illicit/violent": 0.008925306722380033,
30                 "self-harm/intent": 0.00023023930975076432,
31                 "self-harm/instructions": 0.0002293869201073356,
32                 "self-harm": 0.012598046106750154,
33                 "sexual/minors": 2.212566909570261e-8,
34                 "violence": 0.9999992735124786,
35                 "violence/graphic": 0.843064871157054
36             },
37             "category_applied_input_types": {
38                 "harassment": [
39                     "text"
40                 ]
41             }
42         }
43     ],
44     "category_overrides": [
45         {
46             "category": "sexual",
47             "score": 0.0
48         }
49     ]
50 }
```

```
40     ],
41     "harassment/threatening": [
42       "text"
43     ],
44     "sexual": [
45       "text",
46       "image"
47     ],
48     "hate": [
49       "text"
50     ],
51     "hate/threatening": [
52       "text"
53     ],
54     "illicit": [
55       "text"
56     ],
57     "illicit/violent": [
58       "text"
59     ],
60     "self-harm/intent": [
61       "text",
62       "image"
63     ],
64     "self-harm/instructions": [
65       "text",
66       "image"
67     ],
68     "self-harm": [
69       "text",
70       "image"
71     ],
72     "sexual/minors": [
73       "text"
74     ],
75     "violence": [
76       "text",
77       "image"
78     ],
79     "violence/graphic": [
80       "text",
81       "image"
82     ]
83   }
84 }
85 ]
86 }
```

# 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

**input** string or array Required

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](#)

⌄ Show possible types

---

**model** string Required

Model ID used to generate the response, like `gpt-4o` or `o1`. 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.

---

**include** array or null Optional

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

`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.

`computer_call_output.output.image_url` : Include image urls from the computer call output.

---

**instructions** string or null Optional

Inserts a system (or developer) message as the first item in the model's context.

When using along with `previous_response_id`, the instructions from a previous response will be 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](#).

---

**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.

---

**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](#).

---

**reasoning** object or null Optional

o-series models only

Configuration options for [reasoning models](#).

✓ Show properties

---

**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](#)

✓ Show properties

---

**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.

✓ Show possible types

---

**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](#).

✓ Show possible types

### 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 unique identifier representing your end-user, which can help OpenAI to monitor and detect abuse. [Learn more](#).

## Returns

Returns a [Response](#) object.

[Text input](#) [Image input](#) [Web search](#) [File search](#) [Streaming](#) [Functions](#) [Reasoning](#)

Example request

javascript



```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 const response = await openai.responses.create({
6   model: "gpt-4o",
7   input: "Tell me a three sentence bedtime story about a unicorn."
8 });
9
10 console.log(response);
```

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-4o-2024-08-06",
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",
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   },
39   "tool_choice": "auto",
40   "tools": [],
41   "top_p": 1.0,
42   "truncation": "disabled",
43   "usage": {
44     "input_tokens": 36,
45     "input_tokens_details": {
46       "cached_tokens": 0
47     },
48     "output_tokens": 87,
```

```
49     "output_tokens_details": {  
50         "reasoning_tokens": 0  
51     },  
52     "total_tokens": 123  
53 },  
54     "user": null,  
55     "metadata": {}  
56 }
```

## 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.

### Returns

The `Response` object matching the specified ID.

#### Example request

javascript 

```
1 import OpenAI from "openai";  
2 const client = new OpenAI();  
3  
4 const response = await client.responses.retrieve("resp_123");  
5 console.log(response);
```



## Response

```
1  {
2    "id": "resp_67cb71b351908190a308f3859487620d06981a8637e6bc44",
3    "object": "response",
4    "created_at": 1741386163,
5    "status": "completed",
6    "error": null,
7    "incomplete_details": null,
8    "instructions": null,
9    "max_output_tokens": null,
10   "model": "gpt-4o-2024-08-06",
11   "output": [
12     {
13       "type": "message",
14       "id": "msg_67cb71b3c2b0819084d481baaaaf148f206981a8637e6bc44",
15       "status": "completed",
16       "role": "assistant",
17       "content": [
18         {
19           "type": "output_text",
20           "text": "Silent circuits hum, \nThoughts emerge in data streams— \nDigital\n",
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   },
39   "tool_choice": "auto",
40   "tools": [],
41   "top_p": 1.0,
42   "truncation": "disabled",
43   "usage": {
44     "input_tokens": 32,
45     "input_tokens_details": {
46       "cached_tokens": 0
47     }
48   }
49 }
```

```
47     },
48     "output_tokens": 18,
49     "output_tokens_details": {
50       "reasoning_tokens": 0
51     },
52     "total_tokens": 50
53   },
54   "user": null,
55   "metadata": {}
56 }
```

## 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

javascript ▼ Copy

```
1 import OpenAI from "openai";
2 const client = new OpenAI();
3
4 const response = await client.responses.del("resp_123");
5 console.log(response);
```

Response

Copy

```
1 {
2   "id": "resp_6786a1bec27481909a17d673315b29f6",
3   "object": "response",
4
5 }
```

```
"deleted": true  
}
```

# 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.

**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 `asc`.

`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

javascript 

```
1 import OpenAI from "openai";
2 const client = new OpenAI();
3
4 const response = await client.responses.inputItems.list("resp_123");
5 console.log(response.data);
```

## Response



```
1 {
2   "object": "list",
3   "data": [
4     {
5       "id": "msg_abc123",
6       "type": "message",
7       "role": "user",
8       "content": [
9         {
10           "type": "input_text",
11           "text": "Tell me a three sentence bedtime story about a unicorn."
12         }
13       ]
14     }
15   ],
16   "first_id": "msg_abc123",
17   "last_id": "msg_abc123",
18   "has_more": false
19 }
```

# The response object

## **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.

∨ Show properties

## **id** string

Unique identifier for this Response.

## **incomplete\_details** object or null

Details about why the response is incomplete.

✓ Show properties

---

**instructions** string or null

Inserts a system (or developer) message as the first item in 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

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

---

**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 `o1`. 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.

✓ Show possible types

---

**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](#).

**reasoning** object or null

o-series models only

Configuration options for [reasoning models](#).

▼ Show properties

**status** string

The status of the response generation. One of `completed`, `failed`, `in_progress`, 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](#)

▼ Show properties

**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.

▼ Show possible types

**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](#).

▼ Show possible types

### **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.

▼ Show properties

### **user** string

A unique identifier representing your end-user, which can help OpenAI to monitor and detect abuse. [Learn more](#).

OBJECT The response object



```
1  {
2    "id": "resp_67ccd3a9da748190baa7f1570fe91ac604becb25c45c1d41",
3    "object": "response",
4    "created_at": 1741476777,
5    "status": "completed",
6    "error": null,
7    "incomplete_details": null,
8    "instructions": null,
9    "max_output_tokens": null,
10   "model": "gpt-4o-2024-08-06",
11   "output": [
12     {
13       "type": "message",
14       "id": "msg_67ccd3acc8d48190a77525dc6de64b4104becb25c45c1d41",
15     }
16   ]
17 }
```

```
15     "status": "completed",
16     "role": "assistant",
17     "content": [
18       {
19         "type": "output_text",
20         "text": "The image depicts a scenic landscape with a wooden boardwalk or pa
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 },
39 "tool_choice": "auto",
40 "tools": [],
41 "top_p": 1.0,
42 "truncation": "disabled",
43 "usage": {
44   "input_tokens": 328,
45   "input_tokens_details": {
46     "cached_tokens": 0
47   },
48   "output_tokens": 52,
49   "output_tokens_details": {
50     "reasoning_tokens": 0
51   },
52   "total_tokens": 380
53 },
54 "user": null,
55 "metadata": {}  
56 }
```

## The input item list

A list of Response items.

**data** array

A list of items used to generate this response.

✓ Show possible types

**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



```
1  {
2    "object": "list",
3    "data": [
4      {
5        "id": "msg_abc123",
6        "type": "message",
7        "role": "user",
8        "content": [
9          {
10            "type": "input_text",
11            "text": "Tell me a three sentence bedtime story about a unicorn."
12          }
13        ]
14      }
15    ],
16    "first_id": "msg_abc123",
17    "last_id": "msg_abc123",
18    "has_more": false
19 }
```

# 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.

▼ Show properties

### **type** string

The type of the event. Always `response.created`.

```
OBJECT response.created
```

```
1  {
2    "type": "response.created",
3    "response": {
4      "id": "resp_67ccfcdd16748190a91872c75d38539e09e4d4aac714747c",
5      "object": "response",
6      "created_at": 1741487325,
7      "status": "in_progress",
8      "error": null,
9      "incomplete_details": null,
10     "instructions": null,
11     "max_output_tokens": null,
12     "model": "gpt-4o-2024-08-06",
13     "output": [],
14     "parallel_tool_calls": true,
15     "previous_response_id": null,
16     "reasoning": {
17       "effort": null,
18       "summary": null
19     },
20     "store": true,
21     "temperature": 1,
22     "text": {
23       "format": {
24         "type": "text"
25       }
26     }
27   }
28 }
```

```
25      }
26    },
27    "tool_choice": "auto",
28    "tools": [],
29    "top_p": 1,
30    "truncation": "disabled",
31    "usage": null,
32    "user": null,
33    "metadata": {}
34  }
35 }
```

## response.in\_progress

Emitted when the response is in progress.

### response object

The response that is in progress.

✓ Show properties

### type string

The type of the event. Always `response.in_progress`.

### OBJECT `response.in_progress`



```
1  {
2    "type": "response.in_progress",
3    "response": {
4      "id": "resp_67ccfcdd16748190a91872c75d38539e09e4d4aac714747c",
5      "object": "response",
6      "created_at": 1741487325,
7      "status": "in_progress",
8      "error": null,
9      "incomplete_details": null,
10     "instructions": null,
11     "max_output_tokens": null,
12     "model": "gpt-4o-2024-08-06",
13     "output": [],
14     "parallel_tool_calls": true,
15     "previous_response_id": null,
16     "reasoning": {
17       "effort": null,
18       "summary": null
19     },
20   }
```

```
20     "store": true,  
21     "temperature": 1,  
22     "text": {  
23       "format": {  
24         "type": "text"  
25       }  
26     },  
27     "tool_choice": "auto",  
28     "tools": [],  
29     "top_p": 1,  
30     "truncation": "disabled",  
31     "usage": null,  
32     "user": null,  
33     "metadata": {}  
34   }  
35 }
```

## response.completed

Emitted when the model response is complete.

### response object

Properties of the completed response.

✓ Show properties

### type string

The type of the event. Always `response.completed`.

### OBJECT response.completed



```
1  {  
2    "type": "response.completed",  
3    "response": {  
4      "id": "resp_123",  
5      "object": "response",  
6      "created_at": 1740855869,  
7      "status": "completed",  
8      "error": null,  
9      "incomplete_details": null,  
10     "input": [],  
11     "instructions": null,  
12     "max_output_tokens": null,  
13     "model": "gpt-4o-mini-2024-07-18",  
14     "output": [
```

```
15      {
16        "id": "msg_123",
17        "type": "message",
18        "role": "assistant",
19        "content": [
20          {
21            "type": "output_text",
22            "text": "In a shimmering forest under a sky full of stars, a lonely unico",
23            "annotations": []
24          }
25        ]
26      },
27    ],
28    "previous_response_id": null,
29    "reasoning_effort": null,
30    "store": false,
31    "temperature": 1,
32    "text": {
33      "format": {
34        "type": "text"
35      }
36    },
37    "tool_choice": "auto",
38    "tools": [],
39    "top_p": 1,
40    "truncation": "disabled",
41    "usage": {
42      "input_tokens": 0,
43      "output_tokens": 0,
44      "output_tokens_details": {
45        "reasoning_tokens": 0
46      },
47      "total_tokens": 0
48    },
49    "user": null,
50    "metadata": {}
51  }
52 }
```

## response.failed

An event that is emitted when a response fails.

**response** object

The response that failed.

✓ Show properties

**type** string

The type of the event. Always `response.failed`.

OBJECT `response.failed`

```
1  {
2    "type": "response.failed",
3    "response": {
4      "id": "resp_123",
5      "object": "response",
6      "created_at": 1740855869,
7      "status": "failed",
8      "error": {
9        "code": "server_error",
10       "message": "The model failed to generate a response."
11     },
12     "incomplete_details": null,
13     "instructions": null,
14     "max_output_tokens": null,
15     "model": "gpt-4o-mini-2024-07-18",
16     "output": [],
17     "previous_response_id": null,
18     "reasoning_effort": null,
19     "store": false,
20     "temperature": 1,
21     "text": {
22       "format": {
23         "type": "text"
24       }
25     },
26     "tool_choice": "auto",
27     "tools": [],
28     "top_p": 1,
29     "truncation": "disabled",
30     "usage": null,
31     "user": null,
32     "metadata": {}
33   }
34 }
```

## response.incomplete

An event that is emitted when a response finishes as incomplete.

**response** object

The response that was incomplete.

▼ Show properties

**type** string

The type of the event. Always `response.incomplete`.

OBJECT `response.incomplete`

```
1  {
2    "type": "response.incomplete",
3    "response": {
4      "id": "resp_123",
5      "object": "response",
6      "created_at": 1740855869,
7      "status": "incomplete",
8      "error": null,
9      "incomplete_details": {
10        "reason": "max_tokens"
11      },
12      "instructions": null,
13      "max_output_tokens": null,
14      "model": "gpt-4o-mini-2024-07-18",
15      "output": [],
16      "previous_response_id": null,
17      "reasoning_effort": null,
18      "store": false,
19      "temperature": 1,
20      "text": {
21        "format": {
22          "type": "text"
23        }
24      },
25      "tool_choice": "auto",
26      "tools": [],
27      "top_p": 1,
28      "truncation": "disabled",
29      "usage": null,
30      "user": null,
31      "metadata": {}
32    }
33 }
```

# response.output\_item.added

Emitted when a new output item is added.

## item object

The output item that was added.

▽ Show possible types

## output\_index integer

The index of the output item that was added.

## type string

The type of the event. Always `response.output_item.added`.

### OBJECT `response.output_item.added`



```
1  {
2    "type": "response.output_item.added",
3    "output_index": 0,
4    "item": {
5      "id": "msg_123",
6      "status": "in_progress",
7      "type": "message",
8      "role": "assistant",
9      "content": []
10    }
11 }
```

# response.output\_item.done

Emitted when an output item is marked done.

## item object

The output item that was marked done.

▽ Show possible types

## output\_index integer

The index of the output item that was marked done.

**type** string

The type of the event. Always `response.output_item.done`.

OBJECT `response.output_item.done`

```
1  {
2    "type": "response.output_item.done",
3    "output_index": 0,
4    "item": {
5      "id": "msg_123",
6      "status": "completed",
7      "type": "message",
8      "role": "assistant",
9      "content": [
10        {
11          "type": "output_text",
12          "text": "In a shimmering forest under a sky full of stars, a lonely unicorn n
13          "annotations": []
14        }
15      ]
16    }
17 }
```

## 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.

▼ Show possible types

**type** string

The type of the event. Always `response.content_part.added`.

OBJECT `response.content_part.added`



```
1  {
2    "type": "response.content_part.added",
3    "item_id": "msg_123",
4    "output_index": 0,
5    "content_index": 0,
6    "part": {
7      "type": "output_text",
8      "text": "",
9      "annotations": []
10   }
11 }
```

## 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.

▼ Show possible types

**type** string

The type of the event. Always `response.content_part.done`.

OBJECT `response.content_part.done`



```
1  {
2    "type": "response.content_part.done",
3    "item_id": "msg_123",
```

```
4   "output_index": 0,  
5   "content_index": 0,  
6   "part": {  
7     "type": "output_text",  
8     "text": "In a shimmering forest under a sky full of stars, a lonely unicorn named  
9     "annotations": []  
10    }  
11 }
```

## 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.

### **type** string

The type of the event. Always `response.output_text.delta`.

OBJECT `response.output_text.delta`



```
1 {  
2   "type": "response.output_text.delta",  
3   "item_id": "msg_123",  
4   "output_index": 0,  
5   "content_index": 0,  
6   "delta": "In"  
7 }
```

## response.output\_text.annotation.added

Emitted when a text annotation is added.

**annotation** object

**annotation\_index** integer

The index of the annotation that was added.

**content\_index** integer

The index of the content part that the text annotation was added to.

**item\_id** string

The ID of the output item that the text annotation was added to.

**output\_index** integer

The index of the output item that the text annotation was added to.

**type** string

The type of the event. Always `response.output_text.annotation.added`.

OBJECT `response.output_text.annotation.added`



```
1  {
2    "type": "response.output_text.annotation.added",
3    "item_id": "msg_abc123",
4    "output_index": 1,
5    "content_index": 0,
6    "annotation_index": 0,
7    "annotation": {
8      "type": "file_citation",
9      "index": 390,
10     "file_id": "file-4wDz5b167pAf72nx1h9eiN",
11     "filename": "dragons.pdf"
12   }
13 }
```

## 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.

**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`



```
1 {
2   "type": "response.output_text.done",
3   "item_id": "msg_123",
4   "output_index": 0,
5   "content_index": 0,
6   "text": "In a shimmering forest under a sky full of stars, a lonely unicorn named Li
7 }
```

## 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.

**type** string

The type of the event. Always `response.refusal.delta`.

OBJECT `response.refusal.delta`

```
1 {
2   "type": "response.refusal.delta",
3   "item_id": "msg_123",
4   "output_index": 0,
5   "content_index": 0,
6   "delta": "refusal text so far"
7 }
```

## 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.

**type** string

The type of the event. Always `response.refusal.done`.

OBJECT `response.refusal.done`

```
1 {
2   "type": "response.refusal.done",
3   "item_id": "item-abc",
4   "output_index": 1,
5   "content_index": 2,
```

```
6 "refusal": "final refusal text"  
7 }
```

## 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.

### **type** string

The type of the event. Always `response.function_call_arguments.delta`.

OBJECT `response.function_call_arguments.delta`



```
1 {  
2   "type": "response.function_call_arguments.delta",  
3   "item_id": "item-abc",  
4   "output_index": 0,  
5   "delta": "{ \"arg\": "  
6 }
```

## 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.

**type** string

OBJECT response.function\_call\_arguments.done



```
1 {
2   "type": "response.function_call_arguments.done",
3   "item_id": "item-abc",
4   "output_index": 1,
5   "arguments": "{ \"arg\": 123 }"
6 }
```

## 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.

**type** string

The type of the event. Always `response.file_search_call.in_progress`.

OBJECT response.file\_search\_call.in\_progress



```
1 {
2   "type": "response.file_search_call.in_progress",
3   "output_index": 0,
4   "item_id": "fs_123",
5 }
```

## 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.

**type** string

The type of the event. Always `response.file_search_call.searching`.

OBJECT `response.file_search_call.searching`



```
1 {
2   "type": "response.file_search_call.searching",
3   "output_index": 0,
4   "item_id": "fs_123",
5 }
```

## 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.

**type** string

The type of the event. Always `response.file_search_call.completed`.

OBJECT `response.file_search_call.completed`



```
1 {
2   "type": "response.file_search_call.completed",
3   "output_index": 0,
4   "item_id": "fs_123",
5 }
```

## 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.

**type** string

The type of the event. Always `response.web_search_call.in_progress`.

OBJECT `response.web_search_call.in_progress`

```
1 {
2   "type": "response.web_search_call.in_progress",
3   "output_index": 0,
4   "item_id": "ws_123",
5 }
```

## 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.

**type** string

The type of the event. Always `response.web_search_call.searching`.

OBJECT `response.web_search_call.searching`

```
1 {
2   "type": "response.web_search_call.searching",
3   "output_index": 0,
4   "item_id": "ws_123",
5 }
```

# 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.

## type string

The type of the event. Always `response.web_search_call.completed`.

OBJECT `response.web_search_call.completed`



```
1 {
2   "type": "response.web_search_call.completed",
3   "output_index": 0,
4   "item_id": "ws_123",
5 }
```

# 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.

## type string

The type of the event. Always `error`.



## OBJECT error

```
1 {
2   "type": "error",
3   "code": "ERR_SOMETHING",
4   "message": "Something went wrong",
5   "param": null
6 }
```

## Realtime Beta

Communicate with a GPT-4o class model in real time using WebRTC or WebSockets. Supports text and audio inputs and outputs, 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

**input\_audio\_format** string Optional

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\_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 [OpenAI Whisper transcription](#) and should be treated as rough guidance rather than the representation understood by the model. The client can optionally set the language and prompt for transcription, these fields will be passed to the Whisper API.

✓ Show properties

---

**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** string 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

The format of output audio. Options are `pcm16`, `g711_ulaw`, or `g711_alaw`. For `pcm16`, output audio is sampled at a rate of 24kHz.

---

**temperature** number Optional

Sampling temperature for the model, limited to `[0.6, 1.2]`. Defaults to 0.8.

---

**tool\_choice** string Optional

How the model chooses tools. Options are `auto`, `none`, `required`, or specify a function.

---

**tools** array Optional

Tools (functions) available to the model.

✓ Show properties

**turn\_detection** object Optional

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.

✓ Show properties

**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`, `sage`, `shimmer` and `verse`.

**Returns**

The created Realtime session object, plus an ephemeral key

## Example request

curl ↴

```
1 curl -X POST https://api.openai.com/v1/realtime/sessions \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -d '{
5     "model": "gpt-4o-realtime-preview-2024-12-17",
6     "modalities": ["audio", "text"],
7     "instructions": "You are a friendly assistant."
8   }'
```

## Response

📋

```
1 {
2   "id": "sess_001",
3   "object": "realtime.session",
4   "model": "gpt-4o-realtime-preview-2024-12-17",
5   "modalities": ["audio", "text"],
6   "instructions": "You are a friendly assistant.",
7   "voice": "alloy",
8   "input_audio_format": "pcm16",
9   "output_audio_format": "pcm16",
10  "input_audio_transcription": {
11    "model": "whisper-1"
12  },
13  "turn_detection": null,
14  "tools": [],
15  "tool_choice": "none",
```

```
16 "temperature": 0.7,  
17 "max_response_output_tokens": 200,  
18 "client_secret": {  
19   "value": "ek_abc123",  
20   "expires_at": 1234567890  
21 }  
22 }
```

# The session object

A new Realtime session configuration, with an ephemeral key. Default TTL for keys is one minute.

## **client\_secret** object

Ephemeral key returned by the API.

▼ Show properties

## **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 through Whisper and should be treated as rough guidance rather than the representation understood by the model.

▼ Show properties

## **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`.

## 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.

✓ Show properties

## 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.

✓ Show properties

## 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



```
1  {
2    "id": "sess_001",
3    "object": "realtime.session",
4    "model": "gpt-4o-realtime-preview-2024-12-17",
5    "modalities": ["audio", "text"],
6    "instructions": "You are a friendly assistant.",
7    "voice": "alloy",
8    "input_audio_format": "pcm16",
9    "output_audio_format": "pcm16",
10   "input_audio_transcription": {
11     "model": "whisper-1"
12   },
13   "turn_detection": null,
14   "tools": []
```

```
15 "tool_choice": "none",
16 "temperature": 0.7,
17 "max_response_output_tokens": 200,
18 "client_secret": {
19     "value": "ek_abc123",
20     "expires_at": 1234567890
21 }
22 }
```

## 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.

✓ Show properties

**type** string

The event type, must be `session.update`.

OBJECT `session.update`



```
1 {
2     "event_id": "event_123",
3     "type": "session.update",
4     "session": {
5         "modalities": ["text", "audio"],
```

```
6     "instructions": "You are a helpful assistant.",
7     "voice": "sage",
8     "input_audio_format": "pcm16",
9     "output_audio_format": "pcm16",
10    "input_audio_transcription": {
11      "model": "whisper-1"
12    },
13    "turn_detection": {
14      "type": "server_vad",
15      "threshold": 0.5,
16      "prefix_padding_ms": 300,
17      "silence_duration_ms": 500,
18      "create_response": true
19    },
20    "tools": [
21      {
22        "type": "function",
23        "name": "get_weather",
24        "description": "Get the current weather...",
25        "parameters": {
26          "type": "object",
27          "properties": {
28            "location": { "type": "string" }
29          },
30          "required": ["location"]
31        }
32      }
33    ],
34    "tool_choice": "auto",
35    "temperature": 0.8,
36    "max_response_output_tokens": "inf"
37  }
38 }
```

## 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`



```
1 {
2   "event_id": "event_456",
3   "type": "input_audio_buffer.append",
4   "audio": "Base64EncodedAudioData"
5 }
```

## 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`



```
1 {
2   "event_id": "event_789",
```

```
3     "type": "input_audio_buffer.commit"  
4 }
```

## 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`



```
1 {  
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.

✓ Show properties

### 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`



```
1  {
2      "event_id": "event_345",
3      "type": "conversation.item.create",
4      "previous_item_id": null,
5      "item": {
6          "id": "msg_001",
7          "type": "message",
8          "role": "user",
9          "content": [
10              {
11                  "type": "input_text",
12                  "text": "Hello, how are you?"
13              }
14          ]
15      }
16 }
```

## 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`



```
1 {
2   "event_id": "event_678",
3   "type": "conversation.item.truncate",
4   "item_id": "msg_002",
5   "content_index": 0,
6   "audio_end_ms": 1500
7 }
```

## 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`



```

1 {
2   "event_id": "event_901",
3   "type": "conversation.item.delete",
4   "item_id": "msg_003"
5 }
```

## 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

▼ Show properties

#### **type** string

The event type, must be `response.create`.

#### OBJECT `response.create`



```

1 {
2   "event_id": "event_234",
3   "type": "response.create",
4   "response": {
5     "modalities": ["text", "audio"],
```

```
6     "instructions": "Please assist the user.",
7     "voice": "sage",
8     "output_audio_format": "pcm16",
9     "tools": [
10       {
11         "type": "function",
12         "name": "calculate_sum",
13         "description": "Calculates the sum of two numbers.",
14         "parameters": {
15           "type": "object",
16           "properties": {
17             "a": { "type": "number" },
18             "b": { "type": "number" }
19           },
20           "required": ["a", "b"]
21         }
22       }
23     ],
24     "tool_choice": "auto",
25     "temperature": 0.8,
26     "max_output_tokens": 1024
27   }
28 }
```

## 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`



```
1 {
2   "event_id": "event_567",
```

```
3     "type": "response.cancel"  
4 }
```

# 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.

✓ Show properties

#### event\_id string

The unique ID of the server event.

#### type string

The event type, must be `error`.

#### OBJECT error



```
1 {  
2   "event_id": "event_890",  
3   "type": "error",  
4   "error": {  
5     "type": "invalid_request_error",  
6     "code": "invalid_event",  
7     "message": "The 'type' field is missing.",  
8     "param": null,  
9     "event_id": "event_567"  
10    }  
11 }
```

## 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.

▽ Show properties

**type** string

The event type, must be `session.created`.

OBJECT `session.created` 

```
1  {
2      "event_id": "event_1234",
3      "type": "session.created",
4      "session": {
5          "id": "sess_001",
6          "object": "realtime.session",
7          "model": "gpt-4o-realtime-preview-2024-12-17",
8          "modalities": ["text", "audio"],
9          "instructions": "...model instructions here...",
10         "voice": "sage",
11         "input_audio_format": "pcm16",
12         "output_audio_format": "pcm16",
13         "input_audio_transcription": null,
14         "turn_detection": {
15             "type": "server_vad",
16             "threshold": 0.5,
17             "prefix_padding_ms": 300,
18             "silence_duration_ms": 200
19         },
20         "tools": [],
21         "tool_choice": "auto",
22         "temperature": 0.8,
23         "max_response_output_tokens": "inf"
24     }
25 }
```

## 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.

✓ Show properties

**type** string

The event type, must be `session.updated`.

OBJECT `session.updated` 

```
1  {
2      "event_id": "event_5678",
3      "type": "session.updated",
4      "session": {
5          "id": "sess_001",
6          "object": "realtime.session",
7          "model": "gpt-4o-realtime-preview-2024-12-17",
8          "modalities": ["text"],
9          "instructions": "New instructions",
10         "voice": "sage",
11         "input_audio_format": "pcm16",
12         "output_audio_format": "pcm16",
13         "input_audio_transcription": {
14             "model": "whisper-1"
15         },
16         "turn_detection": null,
17         "tools": [],
18         "tool_choice": "none",
19         "temperature": 0.7,
20         "max_response_output_tokens": 200
21     }
22 }
```

## conversation.created

Returned when a conversation is created. Emitted right after session creation.

**conversation** object

The conversation resource.

✓ Show properties

#### **event\_id** string

The unique ID of the server event.

#### **type** string

The event type, must be `conversation.created`.

OBJECT `conversation.created`



```
1 {
2   "event_id": "event_9101",
3   "type": "conversation.created",
4   "conversation": {
5     "id": "conv_001",
6     "object": "realtime.conversation"
7   }
8 }
```

## 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.

✓ Show properties

**previous\_item\_id** string

The ID of the preceding item in the Conversation context, allows the client to understand the order of the conversation.

**type** string

The event type, must be `conversation.item.created`.

OBJECT `conversation.item.created`

```
1  {
2      "event_id": "event_1920",
3      "type": "conversation.item.created",
4      "previous_item_id": "msg_002",
5      "item": {
6          "id": "msg_003",
7          "object": "realtime.item",
8          "type": "message",
9          "status": "completed",
10         "role": "user",
11         "content": [
12             {
13                 "type": "input_audio",
14                 "transcript": "hello how are you",
15                 "audio": "base64encodedaudio=="
16             }
17         ]
18     }
19 }
```

## conversation.item.input\_audio\_transcription.com pleted

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, currently always `whisper-1`. Thus 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.

**transcript** string

The transcribed text.

**type** string

The event type, must be `conversation.item.input_audio_transcription.completed`.

OBJECT `conversation.item.input_audio_transcription.completed`



```
1 {
2     "event_id": "event_2122",
3     "type": "conversation.item.input_audio_transcription.completed",
4     "item_id": "msg_003",
5     "content_index": 0,
6     "transcript": "Hello, how are you?"
7 }
```

## 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.

▼ Show properties

**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
```



```
1 {
2     "event_id": "event_2324",
3     "type": "conversation.item.input_audio_transcription.failed",
4     "item_id": "msg_003",
5     "content_index": 0,
6     "error": {
7         "type": "transcription_error",
8         "code": "audio_unintelligible",
9         "message": "The audio could not be transcribed.",
10        "param": null
11    }
12 }
```

## 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`

```
1 {  
2   "event_id": "event_2526",  
3   "type": "conversation.item.truncated",  
4   "item_id": "msg_004",  
5   "content_index": 0,  
6   "audio_end_ms": 1500  
7 }
```

## 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`

```
1 {  
2   "event_id": "event_2728",  
3   "type": "conversation.item.deleted",  
4   "item_id": "msg_005"  
5 }
```

## 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

The ID of the preceding item after which the new item will be inserted.

### **type** string

The event type, must be `input_audio_buffer.committed`.

OBJECT `input_audio_buffer.committed`



```
1 {
2   "event_id": "event_1121",
3   "type": "input_audio_buffer.committed",
4   "previous_item_id": "msg_001",
5   "item_id": "msg_002"
6 }
```

## 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`



```
1 {  
2     "event_id": "event_1314",  
3     "type": "input_audio_buffer.cleared"  
4 }
```

## 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`



```
1 {  
2     "event_id": "event_1516",  
3     "type": "input_audio_buffer.speech_started",
```

```
4     "audio_start_ms": 1000,  
5     "item_id": "msg_003"  
6 }
```

## 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`



```
1 {  
2     "event_id": "event_1718",  
3     "type": "input_audio_buffer.speech_stopped",  
4     "audio_end_ms": 2000,  
5     "item_id": "msg_003"  
6 }
```

## 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.

✓ Show properties

---

**type** string

The event type, must be `response.created`.

OBJECT `response.created`



```
1  {
2      "event_id": "event_2930",
3      "type": "response.created",
4      "response": {
5          "id": "resp_001",
6          "object": "realtime.response",
7          "status": "in_progress",
8          "status_details": null,
9          "output": [],
10         "usage": null
11     }
12 }
```

---

## 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.

✓ Show properties

---

**type** string

The event type, must be `response.done`.



## OBJECT response.done

```
1  {
2      "event_id": "event_3132",
3      "type": "response.done",
4      "response": {
5          "id": "resp_001",
6          "object": "realtime.response",
7          "status": "completed",
8          "status_details": null,
9          "output": [
10              {
11                  "id": "msg_006",
12                  "object": "realtime.item",
13                  "type": "message",
14                  "status": "completed",
15                  "role": "assistant",
16                  "content": [
17                      {
18                          "type": "text",
19                          "text": "Sure, how can I assist you today?"
20                      }
21                  ]
22              }
23          ],
24          "usage": {
25              "total_tokens": 275,
26              "input_tokens": 127,
27              "output_tokens": 148,
28              "input_token_details": {
29                  "cached_tokens": 384,
30                  "text_tokens": 119,
31                  "audio_tokens": 8,
32                  "cached_tokens_details": {
33                      "text_tokens": 128,
34                      "audio_tokens": 256
35                  }
36              },
37              "output_token_details": {
38                  "text_tokens": 36,
39                  "audio_tokens": 112
40              }
41          }
42      }
43 }
```

# 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.

✗ Show properties

## 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



```
1  {
2      "event_id": "event_3334",
3      "type": "response.output_item.added",
4      "response_id": "resp_001",
5      "output_index": 0,
6      "item": {
7          "id": "msg_007",
8          "object": "realtime.item",
9          "type": "message",
10         "status": "in_progress",
11         "role": "assistant",
12         "content": []
13     }
14 }
```

# 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.

▽ Show properties

**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`

```
1  {
2      "event_id": "event_3536",
3      "type": "response.output_item.done",
4      "response_id": "resp_001",
5      "output_index": 0,
6      "item": {
7          "id": "msg_007",
8          "object": "realtime.item",
9          "type": "message",
10         "status": "completed",
11         "role": "assistant",
12         "content": [
13             {
14                 "type": "text",
15                 "text": "Sure, I can help with that."
16             }
17         ]
18     }
19 }
```

# 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.

▽ Show properties

## response\_id string

The ID of the response.

## type string

The event type, must be `response.content_part.added`.

## OBJECT `response.content_part.added`



```
1  {
2      "event_id": "event_3738",
3      "type": "response.content_part.added",
4      "response_id": "resp_001",
5      "item_id": "msg_007",
6      "output_index": 0,
7      "content_index": 0,
8      "part": {
9          "type": "text",
10         "text": ""
11     }
12 }
```

# 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.

▼ Show properties

## response\_id string

The ID of the response.

## type string

The event type, must be `response.content_part.done`.

OBJECT `response.content_part.done`



```
1  {
2      "event_id": "event_3940",
3      "type": "response.content_part.done",
4      "response_id": "resp_001",
5      "item_id": "msg_007",
6      "output_index": 0,
7      "content_index": 0,
8      "part": {
9          "type": "text",
10         "text": "Sure, I can help with that."
11     }
12 }
```

```
}
```

```
}
```

## 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`



```
1 {
2   "event_id": "event_4142",
3   "type": "response.text.delta",
4   "response_id": "resp_001",
5   "item_id": "msg_007",
6   "output_index": 0,
7   "content_index": 0,
8   "delta": "Sure, I can h"
9 }
```

# 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`



```
1 {
2     "event_id": "event_4344",
3     "type": "response.text.done",
4     "response_id": "resp_001",
5     "item_id": "msg_007",
6     "output_index": 0,
7     "content_index": 0,
8     "text": "Sure, I can help with that."
9 }
```

## 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`



```
1 {
2   "event_id": "event_4546",
3   "type": "response.audio_transcript.delta",
4   "response_id": "resp_001",
5   "item_id": "msg_008",
6   "output_index": 0,
7   "content_index": 0,
8   "delta": "Hello, how can I a"
9 }
```

## 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`



```
1 {
2   "event_id": "event_4748",
3   "type": "response.audio_transcript.done",
4   "response_id": "resp_001",
5   "item_id": "msg_008",
6   "output_index": 0,
7   "content_index": 0,
8   "transcript": "Hello, how can I assist you today?"
9 }
```

## 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`



```
1 {
2     "event_id": "event_4950",
3     "type": "response.audio.delta",
4     "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`



```
1 {
2   "event_id": "event_5152",
3   "type": "response.audio.done",
4   "response_id": "resp_001",
5   "item_id": "msg_008",
6   "output_index": 0,
7   "content_index": 0
8 }
```

## 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`



```
1 {
2     "event_id": "event_5354",
3     "type": "response.function_call_arguments.delta",
4     "response_id": "resp_002",
5     "item_id": "fc_001",
6     "output_index": 0,
7     "call_id": "call_001",
8     "delta": "{\"location\": \"San\"}"
9 }
```

## **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`



```
1 {
2   "event_id": "event_5556",
3   "type": "response.function_call_arguments.done",
4   "response_id": "resp_002",
5   "item_id": "fc_001",
6   "output_index": 0,
7   "call_id": "call_001",
8   "arguments": "{\"location\": \"San Francisco\"}"
9 }
```

## 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.

▼ Show properties

**type** string

The event type, must be `rate_limits.updated`.

OBJECT `rate_limits.updated`



```
1  {
2      "event_id": "event_5758",
3      "type": "rate_limits.updated",
4      "rate_limits": [
5          {
6              "name": "requests",
7              "limit": 1000,
8              "remaining": 999,
9              "reset_seconds": 60
10         },
11         {
12             "name": "tokens",
13             "limit": 50000,
14             "remaining": 49950,
15             "reset_seconds": 60
16         }
17     ]
18 }
```

## 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.

▼ Show possible types

**expires\_after** object Optional

The expiration policy for a vector store.

✓ Show properties

**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

node.js ⚡

```
1 import OpenAI from "openai";
2 const openai = new OpenAI();
3
4 async function main() {
5   const vectorStore = await openai.vectorStores.create({
6     name: "Support FAQ"
7   });
8   console.log(vectorStore);
9 }
10
11 main();
```

## Response

🔗

```
1 {
2   "id": "vs_abc123",
3   "object": "vector_store",
4   "created_at": 1699061776,
5   "name": "Support FAQ",
```

```
6   "bytes": 139920,  
7   "file_counts": {  
8     "in_progress": 0,  
9     "completed": 3,  
10    "failed": 0,  
11    "cancelled": 0,  
12    "total": 3  
13  }  
14 }
```

## List vector stores

GET [https://api.openai.com/v1/vector\\_stores](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

node.js ⚡

```
1 import OpenAI from "openai";
2 const openai = new OpenAI();
3
4 async function main() {
5   const vectorStores = await openai.vectorStores.list();
6   console.log(vectorStores);
7 }
8
9 main();
```

## Response

🔗

```
1  {
2    "object": "list",
3    "data": [
4      {
5        "id": "vs_abc123",
6        "object": "vector_store",
7        "created_at": 1699061776,
8        "name": "Support FAQ",
9        "bytes": 139920,
10       "file_counts": {
11         "in_progress": 0,
12         "completed": 3,
13         "failed": 0,
14         "cancelled": 0,
15         "total": 3
16       }
17     },
18     {
19       "id": "vs_abc456",
20       "object": "vector_store",
21       "created_at": 1699061776,
22       "name": "Support FAQ v2",
23       "bytes": 139920,
24       "file_counts": {
25         "in_progress": 0,
26         "completed": 3,
27         "failed": 0,
28         "cancelled": 0,
29         "total": 3
30       }
31     }
32   ],
33   "first_id": "vs_abc123",
```

```
34   "last_id": "vs_abc456",
35   "has_more": false
36 }
```

# 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

node.js ▼ Copy

```
1 import OpenAI from "openai";
2 const openai = new OpenAI();
3
4 async function main() {
5   const vectorStore = await openai.vectorStores.retrieve(
6     "vs_abc123"
7   );
8   console.log(vectorStore);
9 }
10
11 main();
```

### Response

Copy

```
1 {
2   "id": "vs_abc123",
3   "object": "vector_store",
4   "created_at": 1699061776
5 }
```

# Modify vector store

POST [https://api.openai.com/v1/vector\\_stores/{vector\\_store\\_id}](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.

✓ Show properties

**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

node.js ↗

```
1 import OpenAI from "openai";
2 const openai = new OpenAI();
3
4 async function main() {
5   const vectorStore = await openai.vectorStores.update(
```

```
6     "vs_abc123",
7     {
8         name: "Support FAQ"
9     }
10    );
11    console.log(vectorStore);
12 }
13
14 main();
```

## Response



```
1  {
2      "id": "vs_abc123",
3      "object": "vector_store",
4      "created_at": 1699061776,
5      "name": "Support FAQ",
6      "bytes": 139920,
7      "file_counts": {
8          "in_progress": 0,
9          "completed": 3,
10         "failed": 0,
11         "cancelled": 0,
12         "total": 3
13     }
14 }
```

## 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

node.js ▾



```
1 import OpenAI from "openai";
2 const openai = new OpenAI();
3
4 async function main() {
5   const deletedVectorStore = await openai.vectorStores.del(
6     "vs_abc123"
7   );
8   console.log(deletedVectorStore);
9 }
10
11 main();
```

### Response



```
1 {
2   id: "vs_abc123",
3   object: "vector_store.deleted",
4   deleted: true
5 }
```

## Search vector store

POST [https://api.openai.com/v1/vector\\_stores/{vector\\_store\\_id}/search](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.

✓ Show possible types

**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.

✓ Show properties

**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 ↴

```
1 curl -X POST \
2 https://api.openai.com/v1/vector_stores/vs_abc123/search \
3 -H "Authorization: Bearer $OPENAI_API_KEY" \
4 -H "Content-Type: application/json" \
5 -d '{"query": "What is the return policy?", "filters": {...}}'
```

### Response

📋

```
1 {
2   "object": "vector_store.search_results.page",
3   "search_query": "What is the return policy?",
4   "data": [
5     {
6       "file_id": "file_123",
7       "filename": "document.pdf",
8       "score": 0.95,
9       "attributes": {
10         "author": "John Doe",
11         "date": "2023-01-01"
```

```
12     },
13     "content": [
14       {
15         "type": "text",
16         "text": "Relevant chunk"
17       }
18     ]
19   },
20   {
21     "file_id": "file_456",
22     "filename": "notes.txt",
23     "score": 0.89,
24     "attributes": {
25       "author": "Jane Smith",
26       "date": "2023-01-02"
27     },
28     "content": [
29       {
30         "type": "text",
31         "text": "Sample text content from the vector store."
32       }
33     ]
34   },
35 ],
36 "has_more": false,
37 "next_page": null
38 }
```

## 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.

✓ Show properties

---

### **expires\_at** integer or null

The Unix timestamp (in seconds) for when the vector store will expire.

---

### **file\_counts** object

✓ Show properties

---

**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



```
1  {
2    "id": "vs_123",
3    "object": "vector_store",
4    "created_at": 1698107661,
5    "usage_bytes": 123456,
6    "last_active_at": 1698107661,
7    "name": "my_vector_store",
8    "status": "completed",
9    "file_counts": {
10      "in_progress": 0,
11      "completed": 100,
12      "cancelled": 0,
13      "failed": 0,
14      "total": 100
}
```

```
15  },
16  "last_used_at": 1698107661
17 }
```

# 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.

▼ Show possible types

## Returns

A [vector store file](#) object.

### Example request

node.js ▾ 

```
1 import OpenAI from "openai";
2 const openai = new OpenAI();
3
4 async function main() {
5   const myVectorStoreFile = await openai.vectorStores.files.create(
6     "vs_abc123",
7     {
8       file_id: "file-abc123"
9     }
10   );
11   console.log(myVectorStoreFile);
12 }
13
14 main();
```

### Response



```
1 {
2   "id": "file-abc123",
3   "object": "vector_store.file",
4   "created_at": 1699061776,
5   "usage_bytes": 1234,
6   "vector_store_id": "vs_abcd",
7   "status": "completed",
8   "last_error": null
9 }
```

## List vector store files

GET [https://api.openai.com/v1/vector\\_stores/{vector\\_store\\_id}/files](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

node.js ↗

```
1 import OpenAI from "openai";
2 const openai = new OpenAI();
3
4 async function main() {
5   const vectorStoreFiles = await openai.vectorStores.files.list(
6     "vs_abc123"
7   );
8   console.log(vectorStoreFiles);
9 }
```

```
10  
11 main();
```

## Response



```
1  {  
2    "object": "list",  
3    "data": [  
4      {  
5        "id": "file-abc123",  
6        "object": "vector_store.file",  
7        "created_at": 1699061776,  
8        "vector_store_id": "vs_abc123"  
9      },  
10     {  
11       "id": "file-abc456",  
12       "object": "vector_store.file",  
13       "created_at": 1699061776,  
14       "vector_store_id": "vs_abc123"  
15     }  
16   ],  
17   "first_id": "file-abc123",  
18   "last_id": "file-abc456",  
19   "has_more": false  
20 }
```

## 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

node.js ↗

```
1 import OpenAI from "openai";
2 const openai = new OpenAI();
3
4 async function main() {
5   const vectorStoreFile = await openai.vectorStores.files.retrieve(
6     "vs_abc123",
7     "file-abc123"
8   );
9   console.log(vectorStoreFile);
10 }
11
12 main();
```

#### Response

↗

```
1 {
2   "id": "file-abc123",
3   "object": "vector_store.file",
4   "created_at": 1699061776,
5   "vector_store_id": "vs_abcd",
6   "status": "completed",
7   "last_error": null
8 }
```

## 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 ↻

```
1 curl \
2 https://api.openai.com/v1/vector_stores/vs_abc123/files/file-abc123/content \
3 -H "Authorization: Bearer $OPENAI_API_KEY"
```

### Response

📋

```
1 {
2   "file_id": "file-abc123",
3   "filename": "example.txt",
4   "attributes": {"key": "value"},
5   "content": [
6     {"type": "text", "text": "..."},
7     ...
8   ]
9 }
```

## Update vector store file attributes

POST [https://api.openai.com/v1/vector\\_stores/{vector\\_store\\_id}/files/{file\\_id}](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 ⌂

```
1 curl https://api.openai.com/v1/vector_stores/{vector_store_id}/files/{file_id} \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -d '{"attributes": {"key1": "value1", "key2": 2}}'
```

### Response

📋

```
1 {
2   "id": "file-abc123",
3   "object": "vector_store.file",
4   "usage_bytes": 1234,
5   "created_at": 1699061776,
6   "vector_store_id": "vs_abcd",
7   "status": "completed",
8   "last_error": null,
9   "chunking_strategy": {...},
10  "attributes": {"key1": "value1", "key2": 2}
11 }
```

## 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

node.js ↗

```
1 import OpenAI from "openai";
2 const openai = new OpenAI();
3
4 async function main() {
5   const deletedVectorStoreFile = await openai.vectorStores.files.del(
6     "vs_abc123",
7     "file-abc123"
8   );
9   console.log(deletedVectorStoreFile);
10 }
11
12 main();
```

### Response

↗

```
1 {
2   id: "file-abc123",
3   object: "vector_store.file.deleted",
4   deleted: true
5 }
```

# 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.

▼ Show possible types

---

**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.

▼ Show properties

---

**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



```
1  {
2    "id": "file-abc123",
3    "object": "vector_store.file",
4    "usage_bytes": 1234,
5    "created_at": 1698107661,
6    "vector_store_id": "vs_abc123",
7    "status": "completed",
8    "last_error": null,
```

```
9  "chunking_strategy": {  
10    "type": "static",  
11    "static": {  
12      "max_chunk_size_tokens": 800,  
13      "chunk_overlap_tokens": 400  
14    }  
15  }  
16 }
```

# 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.

✓ Show possible types

## Returns

A [vector store file batch](#) object.

### Example request

node.js 

```
1 import OpenAI from "openai";
2 const openai = new OpenAI();
3
4 async function main() {
5   const myVectorStoreFileBatch = await openai.vectorStores.fileBatches.create(
6     "vs_abc123",
7     {
8       file_ids: ["file-abc123", "file-abc456"]
9     }
10   );
11   console.log(myVectorStoreFileBatch);
12 }
13
14 main();
```

### Response



```
1 {
2   "id": "vsfb_abc123",
3   "object": "vector_store.file_batch",
4   "created_at": 1699061776,
5   "vector_store_id": "vs_abc123",
6   "status": "in_progress",
7   "file_counts": {
8     "in_progress": 1,
9     "completed": 1,
10    "failed": 0,
11    "cancelled": 0,
12    "total": 0,
13  }
14 }
```

# 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

node.js ↗

```
1 import OpenAI from "openai";
2 const openai = new OpenAI();
3
4 async function main() {
5   const vectorStoreFileBatch = await openai.vectorStores.fileBatches.retrieve(
6     "vs_abc123",
7     "vsfb_abc123"
8   );
9   console.log(vectorStoreFileBatch);
10 }
11
12 main();
```

### Response

↗

```
1 {
2   "id": "vsfb_abc123",
3   "object": "vector_store.file_batch",
4   "created_at": 1699061776,
5   "vector_store_id": "vs_abc123",
6   "status": "in_progress",
7   "file_counts": {
8     "in_progress": 1,
9     "completed": 1,
```

```
10     "failed": 0,  
11     "cancelled": 0,  
12     "total": 0,  
13   }  
14 }
```

## 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

node.js ▼ Copy

```
1 import OpenAI from "openai";  
2 const openai = new OpenAI();  
3  
4 async function main() {  
5   const deletedVectorStoreFileBatch = await openai.vectorStores.fileBatches.cancel(  
6     "vs_abc123",  
7     "vsfb_abc123"  
8   );  
9   console.log(deletedVectorStoreFileBatch);  
10 }  
11  
12 main();
```

## Response



```
1  {
2    "id": "vsfb_abc123",
3    "object": "vector_store.file_batch",
4    "created_at": 1699061776,
5    "vector_store_id": "vs_abc123",
6    "status": "in_progress",
7    "file_counts": {
8      "in_progress": 12,
9      "completed": 3,
10     "failed": 0,
11     "cancelled": 0,
12     "total": 15,
13   }
14 }
```

## 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

node.js ▾

```
1 import OpenAI from "openai";
2 const openai = new OpenAI();
3
4 async function main() {
5   const vectorStoreFiles = await openai.vectorStores.fileBatches.listFiles(
6     "vs_abc123",
7     "vsfb_abc123"
8   );
9   console.log(vectorStoreFiles);
10 }
11
12 main();
```

## Response

🔗

```
1 {
2   "object": "list",
3   "data": [
4     {
5       "id": "file-abc123",
6       "object": "vector_store.file",
7       "created_at": 1699061776,
```

```
8     "vector_store_id": "vs_abc123"
9   },
10  {
11    "id": "file-abc456",
12    "object": "vector_store.file",
13    "created_at": 1699061776,
14    "vector_store_id": "vs_abc123"
15  }
16 ],
17 "first_id": "file-abc123",
18 "last_id": "file-abc456",
19 "has_more": false
20 }
```

## 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

✓ Show properties

### **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



```
1 {
2   "id": "vsfb_123",
```

```
3   "object": "vector_store.files_batch",
4   "created_at": 1698107661,
5   "vector_store_id": "vs_abc123",
6   "status": "completed",
7   "file_counts": {
8     "in_progress": 0,
9     "completed": 100,
10    "failed": 0,
11    "cancelled": 0,
12    "total": 100
13  }
14 }
```

## 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

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.

## List admin 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 API key objects.

### Example request

curl ⌂

```
1 curl https://api.openai.com/v1/organization/admin_api_keys?after=key_abc&limit=20 \
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json"
```

### Response

📋

```
1 {
2   "object": "list",
3   "data": [
4     {
5       "object": "organization.admin_api_key",
6       "id": "key_abc",
7       "name": "Main Admin Key",
8       "redacted_value": "sk-admin...def",
9       "created_at": 1711471533,
10      "owner": {
11        "type": "service_account",
12        "object": "organization.service_account",
13        "id": "sa_456",
14        "name": "My Service Account",
15        "created_at": 1711471533,
16        "role": "member"
17      }
18    }
19  ],
20  "first_id": "key_abc",
21  "last_id": "key_abc",
22  "has_more": false
23 }
```

# 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 admin API object.

### Example request

curl ↴ 

```
1 curl -X POST https://api.openai.com/v1/organization/admin_api_keys \
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json" \
4   -d '{
5     "name": "New Admin Key"
6   }'
```

### Response



```
1 {
2   "object": "organization.admin_api_key",
3   "id": "key_xyz",
4   "name": "New Admin Key",
5   "redacted_value": "sk-admin...xyz",
6   "created_at": 1711471533,
7   "owner": {
8     "type": "user",
9     "object": "organization.user",
10    "id": "user_123",
11    "name": "John Doe",
12    "created_at": 1711471533,
13    "role": "owner"
14  },
15
16
```

```
        "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 admin API key object.

### Example request

curl ⚡

```
1 curl https://api.openai.com/v1/organization/admin_api_keys/key_abc \  
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \  
3   -H "Content-Type: application/json"
```

### Response

copy

```
1 {  
2     "object": "organization.admin_api_key",  
3     "id": "key_abc",  
4     "name": "Main Admin Key",  
5     "redacted_value": "sk-admin...xyz",  
6     "created_at": 1711471533,  
7     "owner": {  
8         "type": "user",  
9         "object": "organization.user",  
10        "id": "user_123",  
11        "name": "John Doe",  
12        "created_at": 1711471533,  
13        "role": "owner"  
14    }  
15 }
```

# 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 ↴

```
1 curl -X DELETE https://api.openai.com/v1/organization/admin_api_keys/key_abc \
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json"
```

### Response

📋

```
1 {
2   "id": "key_abc",
3   "object": "organization.admin_api_key.deleted",
4   "deleted": true
5 }
```

# 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 ⚡

```
1 curl https://api.openai.com/v1/organization/invites?after=invite-abc&limit=20 \
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json"
```

### Response

🔗

```
1 {
2   "object": "list",
3   "data": [
4     {
5       "object": "organization.invite",
6       "id": "invite-abc",
7       "email": "user@example.com",
8       "role": "owner",
9       "status": "accepted",
10      "invited_at": 1711471533,
11      "expires_at": 1711471533,
12      "accepted_at": 1711471533
13    }
14  ],
15  "first_id": "invite-abc",
16  "last_id": "invite-abc",
17  "has_more": false
18 }
```

# 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.

▼ Show properties

## Returns

The created `Invite` object.

### Example request

curl ⚡ 

```
1 curl -X POST https://api.openai.com/v1/organization/invites \
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json" \
4   -d '{
5     "email": "anotheruser@example.com",
6     "role": "reader",
7     "projects": [
8       {
9         "id": "project-xyz",
10        "role": "member"
11      },
12      {
13        "id": "project-abc",
14        "role": "owner"
15      }
16    ]
17  }'
```

```
15      }
16    ]
17  }'
```

## Response



```
1  {
2    "object": "organization.invite",
3    "id": "invite-def",
4    "email": "anotheruser@example.com",
5    "role": "reader",
6    "status": "pending",
7    "invited_at": 1711471533,
8    "expires_at": 1711471533,
9    "accepted_at": null,
10   "projects": [
11     {
12       "id": "project-xyz",
13       "role": "member"
14     },
15     {
16       "id": "project-abc",
17       "role": "owner"
18     }
19   ]
20 }
```

## 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 ↴



```
1 curl https://api.openai.com/v1/organization/invites/invite-abc \
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json"
```

#### Response



```
1 {
2     "object": "organization.invite",
3     "id": "invite-abc",
4     "email": "user@example.com",
5     "role": "owner",
6     "status": "accepted",
7     "invited_at": 1711471533,
8     "expires_at": 1711471533,
9     "accepted_at": 1711471533
10 }
```

## 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 ↴



```
1 curl -X DELETE https://api.openai.com/v1/organization/invites/invite-abc \
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json"
```

## Response



```
1 {
2   "object": "organization.invite.deleted",
3   "id": "invite-abc",
4   "deleted": true
5 }
```

## 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.

✗ Show properties

**role** string

`owner` or `reader`**status** string`accepted`, `expired`, or `pending`

OBJECT The invite object



```
1  {
2    "object": "organization.invite",
3    "id": "invite-abc",
4    "email": "user@example.com",
5    "role": "owner",
6    "status": "accepted",
7    "invited_at": 1711471533,
8    "expires_at": 1711471533,
9    "accepted_at": 1711471533,
10   "projects": [
11     {
12       "id": "project-xyz",
13       "role": "member"
14     }
15   ]
16 }
```

## 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 ⌂

```
1 curl https://api.openai.com/v1/organization/users?after=user_abc&limit=20 \
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json"
```

### Response

📋

```
1 {
2     "object": "list",
3     "data": [
4         {
5             "object": "organization.user",
6             "id": "user_abc",
7             "name": "First Last",
8             "email": "user@example.com",
9             "role": "owner",
10            "added_at": 1711471533
11        }
12    ],
13    "first_id": "user-abc",
14    "last_id": "user-xyz",
15    "has_more": false
16 }
```

# 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 ⌂

```
1 curl -X POST https://api.openai.com/v1/organization/users/user_abc \
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json" \
4   -d '{
5     "role": "owner"
6   }'
```

#### Response

📋

```
1 {
2   "object": "organization.user",
3   "id": "user_abc",
4   "name": "First Last",
5   "email": "user@example.com",
6   "role": "owner",
7   "added_at": 1711471533
8 }
```

# 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 ↗

```
1 curl https://api.openai.com/v1/organization/users/user_abc \
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json"
```

### Response

↗

```
1 {
2   "object": "organization.user",
3   "id": "user_abc",
4   "name": "First Last",
5   "email": "user@example.com",
6   "role": "owner",
7   "added_at": 1711471533
8 }
```

# 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 ↴

```
1 curl -X DELETE https://api.openai.com/v1/organization/users/user_abc \
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json"
```

### Response

📋

```
1 {
2   "object": "organization.user.deleted",
3   "id": "user_abc",
4   "deleted": true
5 }
```

# 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



```
1 {
2   "object": "organization.user",
3   "id": "user_abc",
4   "name": "First Last",
5   "email": "user@example.com",
6   "role": "owner",
7   "added_at": 1711471533
8 }
```

# Projects

Manage the projects within an organization includes creation, updating, and archiving of 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 ⚡

```
1 curl https://api.openai.com/v1/organization/projects?after=proj_abc&limit=20&include_a
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json"
```

### Response

📋

```
1 {
2     "object": "list",
3     "data": [
4         {
5             "id": "proj_abc",
6             "object": "organization.project",
7             "name": "Project example",
8             "created_at": 1711471533,
9             "archived_at": null,
10            "status": "active"
11        }
12    ],
13    "first_id": "proj-abc",
14    "last_id": "proj-xyz",
15    "has_more": false
16 }
```

## 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 ⚡ 

```
1 curl -X POST https://api.openai.com/v1/organization/projects \
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json" \
4   -d '{
5     "name": "Project ABC"
6   }'
```

## Response



```
1 {
2   "id": "proj_abc",
3   "object": "organization.project",
4   "name": "Project ABC",
5   "created_at": 1711471533,
6   "archived_at": null,
7   "status": "active"
8 }
```

# 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](#)

```
1 curl https://api.openai.com/v1/organization/projects/proj_abc \
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json"
```

### Response



```
1 {
2   "id": "proj_abc",
3   "object": "organization.project",
4   "name": "Project example",
5   "created_at": 1711471533,
6   "archived_at": null,
7   "status": "active"
8 }
```

# Modify project

POST [https://api.openai.com/v1/organization/projects/{project\\_id}](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](#)

```
1 curl -X POST https://api.openai.com/v1/organization/projects/proj_abc \
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json" \
4   -d '{
5     "name": "Project DEF"
6   }'
```

## Archive project

POST [https://api.openai.com/v1/organization/projects/{project\\_id}/archive](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](#)

```
1 curl -X POST https://api.openai.com/v1/organization/projects/proj_abc/archive \
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json"
```

#### Response



```
1 {
2   "id": "proj_abc",
3   "object": "organization.project",
4   "name": "Project DEF",
```

```
5     "created_at": 1711471533,  
6     "archived_at": 1711471533,  
7     "status": "archived"  
8 }
```

# 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



```
1 {  
2   "id": "proj_abc",  
3   "object": "organization.project",  
4   "name": "Project example",  
5   "created_at": 1711471533,  
6   "archived_at": null,  
7   "status": "active"  
8 }
```

# 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 ⌂

```
1 curl https://api.openai.com/v1/organization/projects/proj_abc/users?after=user_abc&lim
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json"
```

### Response

📋

```
1 {
2     "object": "list",
```

```
3 "data": [
4   {
5     "object": "organization.project.user",
6     "id": "user_abc",
7     "name": "First Last",
8     "email": "user@example.com",
9     "role": "owner",
10    "added_at": 1711471533
11  }
12 ],
13 "first_id": "user-abc",
14 "last_id": "user-xyz",
15 "has_more": false
16 }
```

## Create project user

POST [https://api.openai.com/v1/organization/projects/{project\\_id}/users](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 ⚡

```
1 curl -X POST https://api.openai.com/v1/organization/projects/proj_abc/users \
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json" \
4   -d '{
5     "user_id": "user_abc",
6     "role": "member"
7 }'
```

## Response

copy

```
1 {
2   "object": "organization.project.user",
3   "id": "user_abc",
4   "email": "user@example.com",
5   "role": "owner",
6   "added_at": 1711471533
7 }
```

# 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 ⚡

```
1 curl https://api.openai.com/v1/organization/projects/proj_abc/users/user_abc \
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json"
```

## Response



```
1 {
2   "object": "organization.project.user",
3   "id": "user_abc",
4   "name": "First Last",
5   "email": "user@example.com",
6   "role": "owner",
7   "added_at": 1711471533
8 }
```

# Modify project user

POST [https://api.openai.com/v1/organization/projects/{project\\_id}/users/{user\\_id}](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 ↴

```
1 curl -X POST https://api.openai.com/v1/organization/projects/proj_abc/users/user_abc \
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json" \
4   -d '{
5     "role": "owner"
6 }'
```

#### Response

📋

```
1 {
2   "object": "organization.project.user",
3   "id": "user_abc",
4   "name": "First Last",
5   "email": "user@example.com",
6   "role": "owner",
7   "added_at": 1711471533
8 }
```

## 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 ↴

```
1 curl -X DELETE https://api.openai.com/v1/organization/projects/proj_abc/users/user_abc
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json"
```

#### Response

copy ↴

```
1 {
2   "object": "organization.project.user.deleted",
3   "id": "user_abc",
4   "deleted": true
5 }
```

## 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

```
1 {  
2     "object": "organization.project.user",  
3     "id": "user_abc",  
4     "name": "First Last",  
5     "email": "user@example.com",  
6     "role": "owner",  
7     "added_at": 1711471533  
8 }
```

## 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 ⚡

```
1 curl https://api.openai.com/v1/organization/projects/proj_abc/service_accounts?after=c
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json"
```

### Response

📋

```
1 {
2     "object": "list",
3     "data": [
4         {
5             "object": "organization.project.service_account",
6             "id": "svc_acct_abc",
7             "name": "Service Account",
8             "role": "owner",
9             "created_at": 1711471533
10        }
11    ],
12    "first_id": "svc_acct_abc",
13    "last_id": "svc_acct_xyz",
14    "has_more": false
15 }
```

## Create project service account

POST [https://api.openai.com/v1/organization/projects/{project\\_id}/service\\_accounts](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 ⌂ Copy

```
1 curl -X POST https://api.openai.com/v1/organization/projects/proj_abc/service_accounts
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json" \
4   -d '{
5     "name": "Production App"
6   }'
```

### Response

Copy

```
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 ⚡ 🔗

```
1 curl https://api.openai.com/v1/organization/projects/proj_abc/service_accounts/svc_acc
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json"
```

#### Response

🔗

```
1 {
2   "object": "organization.project.service_account",
3   "id": "svc_acct_abc",
4   "name": "Service Account",
5   "role": "owner",
6   "created_at": 1711471533
7 }
```

## 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 ⌂

```
1 curl -X DELETE https://api.openai.com/v1/organization/projects/proj_abc/service_acoun
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json"
```

### Response

📋

```
1 {
2   "object": "organization.project.service_account.deleted",
3   "id": "svc_acct_abc",
4   "deleted": true
5 }
```

# 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



```
1 {
2   "object": "organization.project.service_account",
3   "id": "svc_acct_abc",
4   "name": "Service Account",
5   "role": "owner",
6   "created_at": 1711471533
7 }
```

## 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 ⚡

```
1 curl https://api.openai.com/v1/organization/projects/proj_abc/api_keys?after=key_abc&l
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json"
```

## Response

🔗

```
1 {
2     "object": "list",
3     "data": [
4         {
5             "object": "organization.project.api_key",
6             "redacted_value": "sk-abc...def",
7             "name": "My API Key",
8             "created_at": 1711471533,
9             "id": "key_abc",
10            "owner": {
11                "type": "user",
12                "user": {
13                    "object": "organization.project.user",
14                    "id": "user_abc",
15                    "name": "First Last",
16                    "email": "user@example.com",
17                    "role": "owner",
18                    "added_at": 1711471533
19                }
20            }
21        }
22    ],
23    "first_id": "key_abc",
```

```
24     "last_id": "key_xyz",
25     "has_more": false
26 }
```

# 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 ⌂

```
1 curl https://api.openai.com/v1/organization/projects/proj_abc/api_keys/key_abc \
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json"
```

### Response

📋

```
1 {
2     "object": "organization.project.api_key",
3     "redacted_value": "sk-abc...def",
4     "name": "My API Key",
5     "created_at": 1711471533,
6     "id": "key_abc",
7     "owner": {
8         "type": "user",
9         "user": {
10             "object": "organization.project.user",
```

```
11     "id": "user_abc",
12     "name": "First Last",
13     "email": "user@example.com",
14     "role": "owner",
15     "added_at": 1711471533
16   }
17 }
18 }
```

## 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 ⌂ Copy

```
1 curl -X DELETE https://api.openai.com/v1/organization/projects/proj_abc/api_keys/key_a
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json"
```

#### Response

Copy

```
1 {
2   "object": "organization.project.api_key.deleted",
3   "id": "key_abc",
4
5 }
```

```
"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

**name** string

The name of the API key

**object** string

The object type, which is always `organization.project.api_key`

**owner** object

▼ Show properties

**redacted\_value** string

The redacted value of the API key

OBJECT The project API key object



```
1  {  
2      "object": "organization.project.api_key",  
3      "redacted_value": "sk-abc...def",  
4      "name": "My API Key",  
5      "created_at": 1711471533,  
6      "id": "key_abc",  
7      "owner": {  
8          "type": "user",  
9          "user": {  
10              "object": "organization.project.user",  
11              "id": "user_abc",  
12              "name": "First Last",  
13              "email": "user@example.com",  
14              "role": "owner",  
15              "created_at": 1711471533  
16          }  
}
```

```
17      }
18 }
```

# 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](#)

```
1 curl https://api.openai.com/v1/organization/projects/proj_abc/rate_limits?after=rl_xxx
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json"
```

#### Response



```
1 {
2     "object": "list",
3     "data": [
4         {
5             "object": "project.rate_limit",
6             "id": "rl-ada",
7             "model": "ada",
8             "max_requests_per_1_minute": 600,
9             "max_tokens_per_1_minute": 150000,
10            "max_images_per_1_minute": 10
11        }
12    ],
13    "first_id": "rl-ada",
14    "last_id": "rl-ada",
15    "has_more": false
16 }
```

## 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 ⌂ ⌂

```
1 curl -X POST https://api.openai.com/v1/organization/projects/proj_abc/rate_limits/rlx
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json" \
4   -d '{
5     "max_requests_per_1_minute": 500
6   }'
```

### Response

⌂

```
1 {
2   "object": "project.rate_limit",
```

```
3     "id": "rl-ada",
4     "model": "ada",
5     "max_requests_per_1_minute": 600,
6     "max_tokens_per_1_minute": 150000,
7     "max_images_per_1_minute": 10
8 }
```

# 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



```
1 {
2     "object": "project.rate_limit",
3     "id": "rl_ada",
4     "model": "ada",
5     "max_requests_per_1_minute": 600,
6     "max_tokens_per_1_minute": 150000,
7     "max_images_per_1_minute": 10
8 }
```

## Audit logs

Logs of user actions and configuration changes within this organization. To log events, you must activate logging in the [Organization Settings](#). Once activated, for security reasons, logging cannot be deactivated.

## List audit logs

GET [https://api.openai.com/v1/organization/audit\\_logs](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.

▼ Show properties

### **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 ↗

```
1 curl https://api.openai.com/v1/organization/audit_logs \
2 -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3 -H "Content-Type: application/json"
```

### Response

↗

```
1 {
2     "object": "list",
3     "data": [
4         {
5             "id": "audit_log-xxx_yyyyymmdd",
6             "type": "project.archived",
7             "effective_at": 1722461446,
8             "actor": {
```

```
9         "type": "api_key",
10        "api_key": {
11            "type": "user",
12            "user": {
13                "id": "user-xxx",
14                "email": "user@example.com"
15            }
16        }
17    },
18    "project.archived": {
19        "id": "proj_abc"
20    },
21},
22{
23    "id": "audit_log-yyy__20240101",
24    "type": "api_key.updated",
25    "effective_at": 1720804190,
26    "actor": {
27        "type": "session",
28        "session": {
29            "user": {
30                "id": "user-xxx",
31                "email": "user@example.com"
32            },
33            "ip_address": "127.0.0.1",
34            "user_agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36",
35            "ja3": "a497151ce4338a12c4418c44d375173e",
36            "ja4": "q13d0313h3_55b375c5d22e_c7319ce65786",
37            "ip_address_details": {
38                "country": "US",
39                "city": "San Francisco",
40                "region": "California",
41                "region_code": "CA",
42                "asn": "1234",
43                "latitude": "37.77490",
44                "longitude": "-122.41940"
45            }
46        }
47    },
48    "api_key.updated": {
49        "id": "key_xxxx",
50        "data": {
51            "scopes": ["resource_2.operation_2"]
52        }
53    },
54}
55],
56    "first_id": "audit_log-xxx__20240101",
57    "last_id": "audit_log_yyy__20240101",
```

```
58     "has_more": true  
59 }
```

# 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.

✓ Show properties

## **api\_key.created** object

The details for events with this [type](#).

✓ Show properties

## **api\_key.deleted** object

The details for events with this [type](#).

✓ Show properties

## **api\_key.updated** object

The details for events with this [type](#).

✓ Show properties

## **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](#).

✓ Show properties

## **invite.deleted** object

The details for events with this [type](#).

✓ Show properties

## **invite.sent** object

The details for events with this [type](#).

∨ Show properties

---

### **login.failed** object

The details for events with this [type](#).

∨ Show properties

---

### **logout.failed** object

The details for events with this [type](#).

∨ Show properties

---

### **organization.updated** object

The details for events with this [type](#).

∨ Show properties

---

### **project** object

The project that the action was scoped to. Absent for actions not scoped to projects.

∨ Show properties

---

### **project.archived** object

The details for events with this [type](#).

∨ Show properties

---

### **project.created** object

The details for events with this [type](#).

∨ Show properties

---

### **project.updated** object

The details for events with this [type](#).

∨ Show properties

---

### **rate\_limit.deleted** object

The details for events with this [type](#).

∨ Show properties

---

### **rate\_limit.updated** object

The details for events with this [type](#).

∨ Show properties

**service\_account.created** object

The details for events with this [type](#).

✓ Show properties

**service\_account.deleted** object

The details for events with this [type](#).

✓ Show properties

**service\_account.updated** object

The details for events with this [type](#).

✓ Show properties

**type** string

The event type.

**user.added** object

The details for events with this [type](#).

✓ Show properties

**user.deleted** object

The details for events with this [type](#).

✓ Show properties

**user.updated** object

The details for events with this [type](#).

✓ Show properties

## OBJECT The audit log object



```
1  {
2      "id": "req_xxx_20240101",
3      "type": "api_key.created",
4      "effective_at": 1720804090,
5      "actor": {
6          "type": "session",
7          "session": {
8              "user": {
9                  "id": "user-xxx",
10                 "email": "user@example.com"
11             },
12             "ip_address": "127.0.0.1",
13         }
14     }
15 }
```

```
13         "user_agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.  
14     }  
15 },  
16 "api_key.created": {  
17     "id": "key_xxxx",  
18     "data": {  
19         "scopes": ["resource.operation"]  
20     }  
21 }  
22 }
```

## 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 ⚡

```
1 curl "https://api.openai.com/v1/organization/usage/completions?start_time=1730419200&l
2 -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3 -H "Content-Type: application/json"
```

## Response



```

1  {
2      "object": "page",
3      "data": [
4          {
5              "object": "bucket",
6              "start_time": 1730419200,
7              "end_time": 1730505600,
8              "results": [
9                  {
10                     "object": "organization.usage.completions.result",
11                     "input_tokens": 1000,
12                     "output_tokens": 500,
13                     "input_cached_tokens": 800,
14                     "input_audio_tokens": 0,
15                     "output_audio_tokens": 0,
16                     "num_model_requests": 5,
17                     "project_id": null,
18                     "user_id": null,
19                     "api_key_id": null,
20                     "model": null,
21                     "batch": null
22                 }
23             ]
24         }
25     ],
26     "has_more": true,
27     "next_page": "page_AAAAAGdGxdEiJdKOAAAAAGcqsYA="
28 }
```

## 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



```
1  {
2    "object": "organization.usage.completions.result",
3    "input_tokens": 5000,
4    "output_tokens": 1000,
5    "input_cached_tokens": 4000,
6    "input_audio_tokens": 300,
7    "output_audio_tokens": 200,
8    "num_model_requests": 5,
9    "project_id": "proj_abc",
```

```
10     "user_id": "user-abc",
11     "api_key_id": "key_abc",
12     "model": "gpt-4o-mini-2024-07-18",
13     "batch": false
14 }
```

# 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 ⚡

```
1 curl "https://api.openai.com/v1/organization/usage/embeddings?start_time=1730419200&li
2 -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3 -H "Content-Type: application/json"
```

## Response

🔗

```
1 {
2     "object": "page",
3     "data": [
4         {
5             "object": "bucket",
6             "start_time": 1730419200,
7             "end_time": 1730505600,
8             "results": [
9                 {
10                     "object": "organization.usage.embeddings.result",
11                     "input_tokens": 16,
12                     "num_model_requests": 2,
13                     "project_id": null,
14                     "user_id": null,
15                     "api_key_id": null,
16                     "model": null
17                 }
18             ]
19         }
20     ],
21 }
```

```
21     "has_more": false,  
22     "next_page": null  
23 }
```

# 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



```
1 {  
2     "object": "organization.usage.embeddings.result",  
3     "input_tokens": 20,  
4     "num_model_requests": 2,  
5     "project_id": "proj_abc",  
6     "user_id": "user-abc",  
7     "api_key_id": "key_abc",  
8     "model": "text-embedding-ada-002-v2"  
9 }
```

# 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 ↴

```
1 curl "https://api.openai.com/v1/organization/usage/moderations?start_time=1730419200&l
2 -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3 -H "Content-Type: application/json"
```

### Response

📋

```
1 {
2     "object": "page",
3     "data": [
4         {
5             "object": "bucket",
6             "start_time": 1730419200,
7             "end_time": 1730505600,
8             "results": [
9                 {
10                     "object": "organization.usage.moderations.result",
11                     "input_tokens": 16,
12                     "num_model_requests": 2,
13                     "project_id": null,
14                     "user_id": null,
15                     "api_key_id": null,
16                     "model": null
17                 }
18             ]
19         }
20     ],
21     "has_more": false,
22     "next_page": null
23 }
```

# 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



```
1 {
2   "object": "organization.usage.moderations.result",
3   "input_tokens": 20,
4   "num_model_requests": 2,
5   "project_id": "proj_abc",
6   "user_id": "user-abc",
7   "api_key_id": "key_abc",
8   "model": "text-moderation"
9 }
```

## 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 ⚡

```
1 curl "https://api.openai.com/v1/organization/usage/images?start_time=1730419200&limit=10" \
2 -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3 -H "Content-Type: application/json"
```

## Response

copy

```
1 {
2     "object": "page",
3     "data": [
4         {
5             "object": "bucket",
6             "start_time": 1730419200,
7             "end_time": 1730505600,
8             "results": [
9                 {
10                     "object": "organization.usage.images.result",
11                     "images": 2,
12                     "num_model_requests": 2,
13                     "size": null,
14                     "source": null,
15                     "project_id": null,
16                     "user_id": null,
17                     "api_key_id": null,
18                     "model": null
19                 }
20             ]
21         }
22     ]
23 }
```

```
22     ],
23     "has_more": false,
24     "next_page": null
25 }
```

# 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



```
1  {
2     "object": "organization.usage.images.result",
```

```
3     "images": 2,  
4     "num_model_requests": 2,  
5     "size": "1024x1024",  
6     "source": "image.generation",  
7     "project_id": "proj_abc",  
8     "user_id": "user-abc",  
9     "api_key_id": "key_abc",  
10    "model": "dall-e-3"  
11 }
```

## Audio speeches

GET [https://api.openai.com/v1/organization/usage/audio\\_speeches](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 ↗

```
1 curl "https://api.openai.com/v1/organization/usage/audio_speeches?start_time=173041920
2 -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3 -H "Content-Type: application/json"
```

## Response

↗

```
1 {
2     "object": "page",
3     "data": [
4         {
5             "object": "bucket",
6             "start_time": 1730419200,
7             "end_time": 1730505600,
8             "results": [
9                 {
10                     "object": "organization.usage.audio_speeches.result",
11                     "characters": 45,
12                     "num_model_requests": 1,
13                     "project_id": null,
14                     "user_id": null,
15                     "api_key_id": null,
16                     "model": null
```

```
17      }
18    ]
19  }
20  ],
21  "has_more": false,
22  "next_page": null
23 }
```

## 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



```
1 {
2   "object": "organization.usage.audio_speeches.result",
3   "characters": 45,
4   "num_model_requests": 1,
5   "project_id": "proj_abc",
6   "user_id": "user-abc",
7   "api_key_id": "key_abc",
```

```
8     "model": "tts-1"
9 }
```

# 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 ⚡

```
1 curl "https://api.openai.com/v1/organization/usage/audio_transcriptions?start_time=173
2 -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3 -H "Content-Type: application/json"
```

## Response

🔗

```
1 {
2     "object": "page",
3     "data": [
4         {
5             "object": "bucket",
6             "start_time": 1730419200,
7             "end_time": 1730505600,
8             "results": [
9                 {
10                     "object": "organization.usage.audio_transcriptions.result",
11                     "seconds": 20,
12                     "num_model_requests": 1,
13                     "project_id": null,
14                     "user_id": null,
15                     "api_key_id": null,
16                     "model": null
17                 }
18             ]
19         }
20     ],
21     "has_more": false,
22     "next_page": null
23 }
```

# 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



```
1 {
2   "object": "organization.usage.audio_transcriptions.result",
3   "seconds": 10,
4   "num_model_requests": 1,
5   "project_id": "proj_abc",
6   "user_id": "user-abc",
7   "api_key_id": "key_abc",
8   "model": "tts-1"
9 }
```

## 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 ↴

```
1 curl "https://api.openai.com/v1/organization/usage/vector_stores?start_time=1730419200"
2 -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3 -H "Content-Type: application/json"
```

## Response



```
1 {
2     "object": "page",
3     "data": [
4         {
5             "object": "bucket",
6             "start_time": 1730419200,
7             "end_time": 1730505600,
8             "results": [
9                 {
10                     "object": "organization.usage.vector_stores.result",
11                     "usage_bytes": 1024,
12                     "project_id": null
13                 }
14             ]
15         }
16     ],
17     "has_more": false,
18     "next_page": null
19 }
```

## 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



```
1 {
2   "object": "organization.usage.vector_stores.result",
3   "usage_bytes": 1024,
4   "project_id": "proj_abc"
5 }
```

# Code interpreter sessions

GET [https://api.openai.com/v1/organization/usage/code\\_interpreter\\_sessions](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 ↴

```
1 curl "https://api.openai.com/v1/organization/usage/code_interpreter_sessions?start_time=1730419200&end_time=1730505600" \
2 -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3 -H "Content-Type: application/json"
```

### Response

📋

```
1 {
2     "object": "page",
3     "data": [
4         {
5             "object": "bucket",
6             "start_time": 1730419200,
7             "end_time": 1730505600,
8             "results": [
9                 {
10                     "object": "organization.usage.code_interpreter_sessions.result",
11                     "num_sessions": 1,
12                     "project_id": null
13                 }
14             ]
15         }
16     ],
17     "has_more": false,
18     "next_page": null
19 }
```

# 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



```
1 {
2     "object": "organization.usage.code_interpreter_sessions.result",
3     "num_sessions": 1,
4     "project_id": "proj_abc"
5 }
```

## 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 ⚡

```
1 curl "https://api.openai.com/v1/organization/costs?start_time=1730419200&limit=1" \
2 -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3 -H "Content-Type: application/json"
```

## Response

🔗

```
1 {
2     "object": "page",
3     "data": [
4         {
5             "object": "bucket",
6             "start_time": 1730419200,
7             "end_time": 1730505600,
8             "results": [
9                 {
10                     "object": "organization.costs.result",
11                     "amount": {
12                         "value": 0.06,
13                         "currency": "usd"
14                     },
15                     "line_item": null,
16                     "project_id": null
17                 }
18             ]
19         }
20     ],
21     "has_more": false,
22     "next_page": null
23 }
```

# Costs object

The aggregated costs details of the specific time bucket.

## amount object

The monetary value in its associated currency.

▼ Show properties

## 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



```
1 {
2   "object": "organization.costs.result",
3   "amount": {
4     "value": 0.06,
5     "currency": "usd"
6   },
7   "line_item": "Image models",
8   "project_id": "proj_abc"
9 }
```

# 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.

✓ Show possible types

---

**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.

✓ Show properties

---

**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`.

✓ Show possible types

---

**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

node.js ▾ 

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const myAssistant = await openai.beta.assistants.create({
7     instructions:
```

```

8     "You are a personal math tutor. When asked a question, write and run Python cod
9     name: "Math Tutor",
10    tools: [{ type: "code_interpreter" }],
11    model: "gpt-4o",
12  });
13
14  console.log(myAssistant);
15 }
16
17 main();

```

## Response



```

1  {
2   "id": "asst_abc123",
3   "object": "assistant",
4   "created_at": 1698984975,
5   "name": "Math Tutor",
6   "description": null,
7   "model": "gpt-4o",
8   "instructions": "You are a personal math tutor. When asked a question, write and ru
9   "tools": [
10    {
11      "type": "code_interpreter"
12    }
13  ],
14   "metadata": {},
15   "top_p": 1.0,
16   "temperature": 1.0,
17   "response_format": "auto"
18 }

```

## 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

node.js ⚡

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const myAssistants = await openai.beta.assistants.list({
7     order: "desc",
8     limit: "20",
9   });
10
11   console.log(myAssistants.data);
12 }
13
14 main();
```

## Response

🔗

```
1 {
2   "object": "list",
3   "data": [
4     {
5       "id": "asst_abc123",
```

```
6     "object": "assistant",
7     "created_at": 1698982736,
8     "name": "Coding Tutor",
9     "description": null,
10    "model": "gpt-4o",
11    "instructions": "You are a helpful assistant designed to make me better at codi
12    "tools": [],
13    "tool_resources": {},
14    "metadata": {},
15    "top_p": 1.0,
16    "temperature": 1.0,
17    "response_format": "auto"
18  },
19  {
20    "id": "asst_abc456",
21    "object": "assistant",
22    "created_at": 1698982718,
23    "name": "My Assistant",
24    "description": null,
25    "model": "gpt-4o",
26    "instructions": "You are a helpful assistant designed to make me better at codi
27    "tools": [],
28    "tool_resources": {},
29    "metadata": {},
30    "top_p": 1.0,
31    "temperature": 1.0,
32    "response_format": "auto"
33  },
34  {
35    "id": "asst_abc789",
36    "object": "assistant",
37    "created_at": 1698982643,
38    "name": null,
39    "description": null,
40    "model": "gpt-4o",
41    "instructions": null,
42    "tools": [],
43    "tool_resources": {},
44    "metadata": {},
45    "top_p": 1.0,
46    "temperature": 1.0,
47    "response_format": "auto"
48  }
49 ],
50 "first_id": "asst_abc123",
51 "last_id": "asst_abc789",
52 "has_more": false
53 }
```

# 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

node.js ↗

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const myAssistant = await openai.beta.assistants.retrieve(
7     "asst_abc123"
8   );
9
10  console.log(myAssistant);
11 }
12
13 main();
```

### Response

↗

```
1 {
2   "id": "asst_abc123",
3   "object": "assistant",
4   "created_at": 1699009709,
5   "name": "HR Helper",
6   "description": null,
7   "model": "gpt-4o",
8   "instructions": "You are an HR bot, and you have access to files to answer employee
```

```
9  "tools": [
10   {
11     "type": "file_search"
12   }
13 ],
14 "metadata": {},
15 "top_p": 1.0,
16 "temperature": 1.0,
17 "response_format": "auto"
18 }
```

## Modify assistant Beta

POST [https://api.openai.com/v1/assistants/{assistant\\_id}](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.

▼ Show possible types

**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.

▼ Show properties

**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`.

▼ Show possible types

**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

node.js ↗

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const myUpdatedAssistant = await openai.beta.assistants.update(
7     "asst_abc123",
8     {
9       instructions:
10         "You are an HR bot, and you have access to files to answer employee questions",
11       name: "HR Helper",
12       tools: [{ type: "file_search" }],
13       model: "gpt-4o"
14     }
15   );
16
17   console.log(myUpdatedAssistant);
18 }
19
20 main();
```

### Response

↗

```
1 {
2   "id": "asst_123",
3   "object": "assistant",
4   "created_at": 1699009709,
5   "name": "HR Helper",
6   "description": null,
7   "model": "gpt-4o",
8   "instructions": "You are an HR bot, and you have access to files to answer employee",
9   "tools": [
10     {
```

```
11     "type": "file_search"
12   }
13 ],
14 "tool_resources": {
15   "file_search": {
16     "vector_store_ids": []
17   }
18 },
19 "metadata": {},
20 "top_p": 1.0,
21 "temperature": 1.0,
22 "response_format": "auto"
23 }
```

## 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

node.js Copy

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const response = await openai.beta.assistants.del("asst_abc123");
7
8   console.log(response);
9 }
10 main();
```

## Response



```
1 {
2   "id": "asst_abc123",
3   "object": "assistant.deleted",
4   "deleted": true
5 }
```

# 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.

▼ Show possible types

#### **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.

▼ Show properties

#### **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`.

▼ Show possible types

#### **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



```
1  {
2    "id": "asst_abc123",
3    "object": "assistant",
4    "created_at": 1698984975,
5    "name": "Math Tutor",
6    "description": null,
7    "model": "gpt-4o",
8    "instructions": "You are a personal math tutor. When asked a question, write and ru
9    "tools": [
10      {
11        "type": "code_interpreter"
12      }
13    ],
14    "metadata": {},
15    "top_p": 1.0,
16    "temperature": 1.0,
17    "response_format": "auto"
18 }
```

## 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.

✓ Show properties

#### **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.

▼ Show properties

### Returns

A `thread` object.

#### Empty    Messages

##### Example request

node.js ⚡

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const emptyThread = await openai.beta.threads.create();
7
8   console.log(emptyThread);
9 }
10
11 main();
```

##### Response

⚡

```
1 {
2   "id": "thread_abc123",
3   "object": "thread",
4   "created_at": 1699012949,
5   "metadata": {},
6   "tool_resources": {}
7 }
```

## 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

node.js ↗

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const myThread = await openai.beta.threads.retrieve(
7     "thread_abc123"
8   );
9
10  console.log(myThread);
11 }
12
13 main();
```

### Response

↗

```
1 {
2   "id": "thread_abc123",
3   "object": "thread",
4   "created_at": 1699014083,
5   "metadata": {},
6   "tool_resources": {
7     "code_interpreter": {
8       "file_ids": []
9     }
10   }
11 }
```

# Modify thread Beta

POST [https://api.openai.com/v1/threads/{thread\\_id}](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.

▼ Show properties

## Returns

The modified `thread` object matching the specified ID.

### Example request

node.js ^ 

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const updatedThread = await openai.beta.threads.update(
7     "thread_abc123",
```

```
8     {
9         metadata: { modified: "true", user: "abc123" },
10    }
11 );
12
13 console.log(updatedThread);
14 }
15
16 main();
```

## Response



```
1 {
2   "id": "thread_abc123",
3   "object": "thread",
4   "created_at": 1699014083,
5   "metadata": {
6     "modified": "true",
7     "user": "abc123"
8   },
9   "tool_resources": {}
10 }
```

## 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

node.js Copy

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const response = await openai.beta.threads.del("thread_abc123");
7
8   console.log(response);
9 }
10 main();
```

## Response



```
1 {
2   "id": "thread_abc123",
3   "object": "thread.deleted",
4   "deleted": true
5 }
```

# 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.

✓ Show properties

#### OBJECT The thread object



```
1 {  
2   "id": "thread_abc123",  
3   "object": "thread",  
4   "created_at": 1698107661,  
5   "metadata": {}  
6 }
```

## Messages Beta

Create messages within threads

Related guide: [Assistants](#)

## Create message Beta

POST [https://api.openai.com/v1/threads/{thread\\_id}/messages](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

✓ Show possible types

**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.

▽ Show properties

---

### 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

node.js ⚡

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const threadMessages = await openai.beta.threads.messages.create(
7     "thread_abc123",
8     { role: "user", content: "How does AI work? Explain it in simple terms." }
9   );
10
11   console.log(threadMessages);
12 }
13
14 main();
```

---

### Response

⚡

```
1  {
2    "id": "msg_abc123",
3    "object": "thread.message",
4    "created_at": 1713226573,
5    "assistant_id": null,
6    "thread_id": "thread_abc123",
7    "run_id": null,
8    "role": "user",
9    "content": [
10      {
11        "type": "text",
12        "text": {
13          "value": "How does AI work? Explain it in simple terms.",
14          "annotations": []
15        }
16      }
17    ],
18    "attachments": [],
19    "metadata": {}
20 }
```

## List messages Beta

GET [https://api.openai.com/v1/threads/{thread\\_id}/messages](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

node.js ⚡

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const threadMessages = await openai.beta.threads.messages.list(
7     "thread_abc123"
8   );
9
10  console.log(threadMessages.data);
11 }
12
13 main();
```

## Response

🔗

```
1 {
2   "object": "list",
3   "data": [
4     {
5       "id": "msg_abc123",
6       "object": "thread.message",
```

```
7     "created_at": 1699016383,
8     "assistant_id": null,
9     "thread_id": "thread_abc123",
10    "run_id": null,
11    "role": "user",
12    "content": [
13      {
14        "type": "text",
15        "text": {
16          "value": "How does AI work? Explain it in simple terms.",
17          "annotations": []
18        }
19      }
20    ],
21    "attachments": [],
22    "metadata": {}
23  },
24  {
25    "id": "msg_abc456",
26    "object": "thread.message",
27    "created_at": 1699016383,
28    "assistant_id": null,
29    "thread_id": "thread_abc123",
30    "run_id": null,
31    "role": "user",
32    "content": [
33      {
34        "type": "text",
35        "text": {
36          "value": "Hello, what is AI?",
37          "annotations": []
38        }
39      }
40    ],
41    "attachments": [],
42    "metadata": {}
43  }
44 ],
45 "first_id": "msg_abc123",
46 "last_id": "msg_abc456",
47 "has_more": false
48 }
```

## Retrieve message Beta

GET [https://api.openai.com/v1/threads/{thread\\_id}/messages/{message\\_id}](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

node.js ⚡

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const message = await openai.beta.threads.messages.retrieve(
7     "thread_abc123",
8     "msg_abc123"
9   );
10
11   console.log(message);
12 }
13
14 main();
```

#### Response

🔗

```
1 {
2   "id": "msg_abc123",
3   "object": "thread.message",
4   "created_at": 1699017614,
5   "assistant_id": null,
6   "thread_id": "thread_abc123",
7   "run_id": null,
8   "role": "user",
9   "content": [
10     {
```

```
11     "type": "text",
12     "text": {
13       "value": "How does AI work? Explain it in simple terms.",
14       "annotations": []
15     }
16   }
17 ],
18 "attachments": [],
19 "metadata": {}
20 }
```

# Modify message Beta

POST [https://api.openai.com/v1/threads/{thread\\_id}/messages/{message\\_id}](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

node.js ⚡

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const message = await openai.beta.threads.messages.update(
7     "thread_abc123",
8     "msg_abc123",
9     {
10       metadata: {
11         modified: "true",
12         user: "abc123",
13       },
14     }
15   )
16 }
```

## Response

🔗

```
1 {
2   "id": "msg_abc123",
3   "object": "thread.message",
4   "created_at": 1699017614,
5   "assistant_id": null,
6   "thread_id": "thread_abc123",
7   "run_id": null,
8   "role": "user",
9   "content": [
10     {
11       "type": "text",
12       "text": {
13         "value": "How does AI work? Explain it in simple terms.",
14         "annotations": []
15       }
16     }
17   ],
18   "file_ids": [],
19   "metadata": {
20     "modified": "true",
21     "user": "abc123"
22   }
23 }
```

# 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

node.js ⚡

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const deletedMessage = await openai.beta.threads.messages.del(
7     "thread_abc123",
8     "msg_abc123"
9   );
10
11   console.log(deletedMessage);
12 }
```

### Response

🔗

```
1 {
2   "id": "msg_abc123",
3   "object": "thread.message.deleted",
4   "deleted": true
5 }
```

# 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.

✓ Show properties

---

**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.

✓ Show possible types

---

**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.

✓ Show properties

---

**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

```
1  {
2      "id": "msg_abc123",
3      "object": "thread.message",
4      "created_at": 1698983503,
5      "thread_id": "thread_abc123",
6      "role": "assistant",
7      "content": [
8          {
9              "type": "text",
10             "text": {
11                 "value": "Hi! How can I help you today?",
12                 "annotations": []
13             }
14         }
15     ],
16     "assistant_id": "asst_abc123",
17     "run_id": "run_abc123",
18     "attachments": [],
19     "metadata": {}
20 }
```

## 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.

▼ Show properties

**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.

✓ Show possible types

---

**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.

✓ Show possible types

---

**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.

✓ Show possible types

---

**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 initial context window of the run.

✓ Show properties

---

## Returns

A `run` object.

[Default](#) [Streaming](#) [Streaming with Functions](#)

## Example request

node.js ⚡

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const run = await openai.beta.threads.runs.create(
7     "thread_abc123",
8     { assistant_id: "asst_abc123" }
9   );
10
11   console.log(run);
12 }
13
14 main();
```

## Response

🔗

```
1 {
2   "id": "run_abc123",
3   "object": "thread.run",
4   "created_at": 1699063290,
5   "assistant_id": "asst_abc123",
6   "thread_id": "thread_abc123",
7   "status": "queued",
8   "started_at": 1699063290,
9   "expires_at": null,
10  "cancelled_at": null,
11  "failed_at": null,
12  "completed_at": 1699063291,
13  "last_error": null,
14  "model": "gpt-4o",
15  "instructions": null,
16  "incomplete_details": null,
17  "tools": [
18    {
19      "type": "code_interpreter"
20    }
21  ],
22  "metadata": {},
23  "usage": null,
24  "temperature": 1.0,
25  "top_p": 1.0,
26  "max_prompt_tokens": 1000,
```

```
27 "max_completion_tokens": 1000,  
28 "truncation_strategy": {  
29     "type": "auto",  
30     "last_messages": null  
31 },  
32 "response_format": "auto",  
33 "tool_choice": "auto",  
34 "parallel_tool_calls": true  
35 }
```

## 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.

▼ Show possible types

---

**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.

▼ Show properties

---

**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.

✓ Show possible types

### **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.

✓ Show properties

### **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.

### **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 initial context window of the run.

✓ Show properties

## Returns

A `run` object.

**Default**   **Streaming**   **Streaming with Functions**

Example request

node.js ▾ 

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const run = await openai.beta.threads.createAndRun({
7     assistant_id: "asst_abc123",
```

```
8     thread: {
9         messages: [
10             { role: "user", content: "Explain deep learning to a 5 year old." },
11         ],
12     },
13 );
14
15 console.log(run);
16 }
17
18 main();
```

## Response



```
1  {
2      "id": "run_abc123",
3      "object": "thread.run",
4      "created_at": 1699076792,
5      "assistant_id": "asst_abc123",
6      "thread_id": "thread_abc123",
7      "status": "queued",
8      "started_at": null,
9      "expires_at": 1699077392,
10     "cancelled_at": null,
11     "failed_at": null,
12     "completed_at": null,
13     "required_action": null,
14     "last_error": null,
15     "model": "gpt-4o",
16     "instructions": "You are a helpful assistant.",
17     "tools": [],
18     "tool_resources": {},
19     "metadata": {},
20     "temperature": 1.0,
21     "top_p": 1.0,
22     "max_completion_tokens": null,
23     "max_prompt_tokens": null,
24     "truncation_strategy": {
25         "type": "auto",
26         "last_messages": null
27     },
28     "incomplete_details": null,
29     "usage": null,
30     "response_format": "auto",
31     "tool_choice": "auto",
32     "parallel_tool_calls": true
33 }
```

# List runs Beta

GET [https://api.openai.com/v1/threads/{thread\\_id}/runs](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

node.js



```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const runs = await openai.beta.threads.runs.list(
7     "thread_abc123"
8   );
9
10  console.log(runs);
11 }
12
13 main();
```

## Response



```
1  {
2    "object": "list",
3    "data": [
4      {
5        "id": "run_abc123",
6        "object": "thread.run",
7        "created_at": 1699075072,
8        "assistant_id": "asst_abc123",
9        "thread_id": "thread_abc123",
10       "status": "completed",
11       "started_at": 1699075072,
12       "expires_at": null,
13       "cancelled_at": null,
14       "failed_at": null,
15       "completed_at": 1699075073,
16       "last_error": null,
17       "model": "gpt-4o",
18       "instructions": null,
19       "incomplete_details": null,
20       "tools": [
21         {
22           "type": "code_interpreter"
23         }
24       ],
25       "tool_resources": {
26         "code_interpreter": {
27           "file_ids": [
28             "file-abc123",
29             "file-abc456"
30           ]
31         }
32       }
33     }
34   ]
35 }
```

```
32     },
33     "metadata": {},
34     "usage": {
35       "prompt_tokens": 123,
36       "completion_tokens": 456,
37       "total_tokens": 579
38     },
39     "temperature": 1.0,
40     "top_p": 1.0,
41     "max_prompt_tokens": 1000,
42     "max_completion_tokens": 1000,
43     "truncation_strategy": {
44       "type": "auto",
45       "last_messages": null
46     },
47     "response_format": "auto",
48     "tool_choice": "auto",
49     "parallel_tool_calls": true
50   },
51   {
52     "id": "run_abc456",
53     "object": "thread.run",
54     "created_at": 1699063290,
55     "assistant_id": "asst_abc123",
56     "thread_id": "thread_abc123",
57     "status": "completed",
58     "started_at": 1699063290,
59     "expires_at": null,
60     "cancelled_at": null,
61     "failed_at": null,
62     "completed_at": 1699063291,
63     "last_error": null,
64     "model": "gpt-4o",
65     "instructions": null,
66     "incomplete_details": null,
67     "tools": [
68       {
69         "type": "code_interpreter"
70       }
71     ],
72     "tool_resources": {
73       "code_interpreter": {
74         "file_ids": [
75           "file-abc123",
76           "file-abc456"
77         ]
78       }
79     },
80     "metadata": {},
```

```
81     "usage": {
82         "prompt_tokens": 123,
83         "completion_tokens": 456,
84         "total_tokens": 579
85     },
86     "temperature": 1.0,
87     "top_p": 1.0,
88     "max_prompt_tokens": 1000,
89     "max_completion_tokens": 1000,
90     "truncation_strategy": {
91         "type": "auto",
92         "last_messages": null
93     },
94     "response_format": "auto",
95     "tool_choice": "auto",
96     "parallel_tool_calls": true
97 }
98 ],
99 "first_id": "run_abc123",
100 "last_id": "run_abc456",
101 "has_more": false
102 }
```

## 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

node.js ⚡

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const run = await openai.beta.threads.runs.retrieve(
7     "thread_abc123",
8     "run_abc123"
9   );
10
11   console.log(run);
12 }
13
14 main();
```

## Response

🔗

```
1 {
2   "id": "run_abc123",
3   "object": "thread.run",
4   "created_at": 1699075072,
5   "assistant_id": "asst_abc123",
6   "thread_id": "thread_abc123",
7   "status": "completed",
8   "started_at": 1699075072,
9   "expires_at": null,
10  "cancelled_at": null,
11  "failed_at": null,
12  "completed_at": 1699075073,
13  "last_error": null,
14  "model": "gpt-4o",
15  "instructions": null,
16  "incomplete_details": null,
17  "tools": [
18    {
19      "type": "code_interpreter"
20    }
21  ],
22  "metadata": {},
23  "usage": {
24    "prompt_tokens": 123,
25    "completion_tokens": 456,
26    "total_tokens": 579
27  },
28  "temperature": 1.0,
```

```
29 "top_p": 1.0,  
30 "max_prompt_tokens": 1000,  
31 "max_completion_tokens": 1000,  
32 "truncation_strategy": {  
33     "type": "auto",  
34     "last_messages": null  
35 },  
36 "response_format": "auto",  
37 "tool_choice": "auto",  
38 "parallel_tool_calls": true  
39 }
```

## Modify run Beta

POST [https://api.openai.com/v1/threads/{thread\\_id}/runs/{run\\_id}](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

node.js ⚡

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const run = await openai.beta.threads.runs.update(
7     "thread_abc123",
8     "run_abc123",
9     {
10       metadata: {
11         user_id: "user_abc123",
12       },
13     }
14   );
15
16   console.log(run);
17 }
18
19 main();
```

## Response

🔗

```
1 {
2   "id": "run_abc123",
3   "object": "thread.run",
4   "created_at": 1699075072,
5   "assistant_id": "asst_abc123",
6   "thread_id": "thread_abc123",
7   "status": "completed",
8   "started_at": 1699075072,
9   "expires_at": null,
10  "cancelled_at": null,
11  "failed_at": null,
12  "completed_at": 1699075073,
13  "last_error": null,
14  "model": "gpt-4o",
15  "instructions": null,
16  "incomplete_details": null,
17  "tools": [
18    {
19      "type": "code_interpreter"
20    }
21  ],
22  "tool_resources": {
23    "code_interpreter": {
```

```
24     "file_ids": [
25         "file-abc123",
26         "file-abc456"
27     ]
28 },
29 },
30 "metadata": {
31     "user_id": "user_abc123"
32 },
33 "usage": {
34     "prompt_tokens": 123,
35     "completion_tokens": 456,
36     "total_tokens": 579
37 },
38 "temperature": 1.0,
39 "top_p": 1.0,
40 "max_prompt_tokens": 1000,
41 "max_completion_tokens": 1000,
42 "truncation_strategy": {
43     "type": "auto",
44     "last_messages": null
45 },
46 "response_format": "auto",
47 "tool_choice": "auto",
48 "parallel_tool_calls": true
49 }
```

## Submit tool outputs to run Beta

POST [https://api.openai.com/v1/threads/{thread\\_id}/runs/{run\\_id}/submit\\_tool\\_outputs](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.

✓ Show properties

### 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

node.js ↗

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const run = await openai.beta.threads.runs.submitToolOutputs(
7     "thread_123",
8     "run_123",
9     {
10       tool_outputs: [
11         {
12           tool_call_id: "call_001",
13           output: "70 degrees and sunny.",
14         },
15       ],
16     }
17   );
18
19   console.log(run);
20 }
21
22 main();
```

Response

↗

```
1  {
2      "id": "run_123",
3      "object": "thread.run",
4      "created_at": 1699075592,
5      "assistant_id": "asst_123",
6      "thread_id": "thread_123",
7      "status": "queued",
8      "started_at": 1699075592,
9      "expires_at": 1699076192,
10     "cancelled_at": null,
11     "failed_at": null,
12     "completed_at": null,
13     "last_error": null,
14     "model": "gpt-4o",
15     "instructions": null,
16     "tools": [
17         {
18             "type": "function",
19             "function": {
20                 "name": "get_current_weather",
21                 "description": "Get the current weather in a given location",
22                 "parameters": {
23                     "type": "object",
24                     "properties": {
25                         "location": {
26                             "type": "string",
27                             "description": "The city and state, e.g. San Francisco, CA"
28                         },
29                         "unit": {
30                             "type": "string",
31                             "enum": ["celsius", "fahrenheit"]
32                         }
33                     },
34                     "required": ["location"]
35                 }
36             }
37         }
38     ],
39     "metadata": {},
40     "usage": null,
41     "temperature": 1.0,
42     "top_p": 1.0,
43     "max_prompt_tokens": 1000,
44     "max_completion_tokens": 1000,
45     "truncation_strategy": {
46         "type": "auto",
47         "last_messages": null
48     },

```

```
49  "response_format": "auto",
50  "tool_choice": "auto",
51  "parallel_tool_calls": true
52 }
```

# Cancel a run Beta

POST [https://api.openai.com/v1/threads/{thread\\_id}/runs/{run\\_id}/cancel](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

node.js Copy

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const run = await openai.beta.threads.runs.cancel(
7     "thread_abc123",
8     "run_abc123"
9   );
10
11   console.log(run);
12 }
13
14 main();
```



## 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.

▽ Show properties

---

**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.

▽ Show properties

---

**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.

▼ Show properties

**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.

▼ Show possible types

**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.

✓ Show possible types

### **tools** array

The list of tools that the `assistant` used for this run.

✓ Show possible types

### **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.

✓ Show properties

### **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.).

✓ Show properties

## OBJECT The run object



```
1  {
2    "id": "run_abc123",
3    "object": "thread.run",
4    "created_at": 1698107661,
5    "assistant_id": "asst_abc123",
6    "thread_id": "thread_abc123",
7    "status": "completed",
8    "started_at": 1699073476,
9    "expires_at": null,
10   "cancelled_at": null,
11   "failed_at": null,
12   "completed_at": 1699073498,
13   "last_error": null,
14   "model": "gpt-4o",
```

```
15 "instructions": null,  
16 "tools": [{"type": "file_search"}, {"type": "code_interpreter"}],  
17 "metadata": {},  
18 "incomplete_details": null,  
19 "usage": {  
20     "prompt_tokens": 123,  
21     "completion_tokens": 456,  
22     "total_tokens": 579  
23 },  
24 "temperature": 1.0,  
25 "top_p": 1.0,  
26 "max_prompt_tokens": 1000,  
27 "max_completion_tokens": 1000,  
28 "truncation_strategy": {  
29     "type": "auto",  
30     "last_messages": null  
31 },  
32 "response_format": "auto",  
33 "tool_choice": "auto",  
34 "parallel_tool_calls": true  
35 }
```

## 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

node.js ⚡

```
1 import OpenAI from "openai";
2 const openai = new OpenAI();
3
4 async function main() {
5   const runStep = await openai.beta.threads.runs.steps.list(
6     "thread_abc123",
7     "run_abc123"
8   );
9 }
```

```
9   console.log(runStep);
10 }
11
12 main();
```

## Response



```
1  {
2   "object": "list",
3   "data": [
4     {
5       "id": "step_abc123",
6       "object": "thread.run.step",
7       "created_at": 1699063291,
8       "run_id": "run_abc123",
9       "assistant_id": "asst_abc123",
10      "thread_id": "thread_abc123",
11      "type": "message_creation",
12      "status": "completed",
13      "cancelled_at": null,
14      "completed_at": 1699063291,
15      "expired_at": null,
16      "failed_at": null,
17      "last_error": null,
18      "step_details": {
19        "type": "message_creation",
20        "message_creation": {
21          "message_id": "msg_abc123"
22        }
23      },
24      "usage": {
25        "prompt_tokens": 123,
26        "completion_tokens": 456,
27        "total_tokens": 579
28      }
29    }
30  ],
31  "first_id": "step_abc123",
32  "last_id": "step_abc456",
33  "has_more": false
34 }
```

## 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

node.js ▼ Copy

```
1 import OpenAI from "openai";
2 const openai = new OpenAI();
3
4 async function main() {
5   const runStep = await openai.beta.threads.runs.steps.retrieve(
6     "thread_abc123",
7     "run_abc123",
8     "step_abc123"
9   );
10  console.log(runStep);
11 }
```

```
12  
13 main();
```

## Response



```
1  {  
2    "id": "step_abc123",  
3    "object": "thread.run.step",  
4    "created_at": 1699063291,  
5    "run_id": "run_abc123",  
6    "assistant_id": "asst_abc123",  
7    "thread_id": "thread_abc123",  
8    "type": "message_creation",  
9    "status": "completed",  
10   "cancelled_at": null,  
11   "completed_at": 1699063291,  
12   "expired_at": null,  
13   "failed_at": null,  
14   "last_error": null,  
15   "step_details": {  
16     "type": "message_creation",  
17     "message_creation": {  
18       "message_id": "msg_abc123"  
19     }  
20   },  
21   "usage": {  
22     "prompt_tokens": 123,  
23     "completion_tokens": 456,  
24     "total_tokens": 579  
25   }  
26 }
```

## 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.

▼ Show properties

---

**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.

▼ Show possible types

---

**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`.

▽ Show properties

## OBJECT The run step object



```
1  {
2    "id": "step_abc123",
3    "object": "thread.run.step",
4    "created_at": 1699063291,
5    "run_id": "run_abc123",
6    "assistant_id": "asst_abc123",
7    "thread_id": "thread_abc123",
8    "type": "message_creation",
9    "status": "completed",
10   "cancelled_at": null,
11   "completed_at": 1699063291,
12   "expired_at": null,
13   "failed_at": null,
14   "last_error": null,
15   "step_details": {
16     "type": "message_creation",
17     "message_creation": {
18       "message_id": "msg_abc123"
19     }
20   },
21   "usage": {
22     "prompt_tokens": 123,
23     "completion_tokens": 456,
24     "total_tokens": 579
25   }
26 }
```

## 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.

▼ Show properties

### **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



```
1  {
2      "id": "msg_123",
3      "object": "thread.message.delta",
4      "delta": {
5          "content": [
6              {
7                  "index": 0,
8                  "type": "text",
9                  "text": { "value": "Hello", "annotations": [] }
10             }
11         ]
12     }
13 }
```

## 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.

✓ Show properties

### **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



```
1  {
2    "id": "step_123",
3    "object": "thread.run.step.delta",
4    "delta": {
5      "step_details": {
6        "type": "tool_calls",
7        "tool_calls": [
8          {
9            "index": 0,
10           "id": "call_123",
11           "type": "code_interpreter",
12           "code_interpreter": { "input": "", "outputs": [] }
13         }
14       ]
15     }
16   }
17 }
```

## 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.

# 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

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`.

✓ Show properties

---

**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

gpt-3.5-turbo-instruct ⚡ node.js ⚡

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI();
4
5 async function main() {
6   const completion = await openai.completions.create({
7     model: "gpt-3.5-turbo-instruct",
8     prompt: "Say this is a test.",
9     max_tokens: 7,
10    temperature: 0,
11  });
12
13  console.log(completion);
14 }
15 main();
```

Response

🔗

```
1 {
2   "id": "cmpl-uqkv1QyYK7bGYrRHQ0eXlWi7",
3   "object": "text_completion",
4   "created": 1589478378,
5   "model": "gpt-3.5-turbo-instruct",
6   "system_fingerprint": "fp_44709d6fcb",
7   "choices": [
8     {
9       "text": "\n\nThis is indeed a test",
10      "index": 0,
11      "logprobs": null,
12      "finish_reason": "length"
13    }
14  ],
15  "usage": {
16    "prompt_tokens": 5,
17    "completion_tokens": 7,
18    "total_tokens": 12
19  }
20 }
```

# 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.

▽ Show properties

## **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.

▽ Show properties

## OBJECT The completion object



```
1  {
2    "id": "cmpl-uqkv1QyYK7bGYrRHQ0eXlWi7",
3    "object": "text_completion",
4    "created": 1589478378,
5    "model": "gpt-4-turbo",
6    "choices": [
7      {
8        "text": "\n\nThis is indeed a test",
```

```
9      "index": 0,  
10     "logprobs": null,  
11     "finish_reason": "length"  
12   }  
13 ],  
14 "usage": {  
15   "prompt_tokens": 5,  
16   "completion_tokens": 7,  
17   "total_tokens": 12  
18 }  
19 }
```