



Introduction

You can interact with the API through HTTP requests from any language, via our official Python bindings, our official Node.js library, or a [community-maintained library](#).

To install the official Python bindings, run the following command:

```
pip install openai
```



To install the official Node.js library, run the following command in your Node.js project directory:

```
npm install openai
```



Authentication

API keys

The OpenAI API uses API keys for authentication. You can create API keys at a user or service account level. Service accounts are tied to a "bot" individual and should be used to provision access for production systems. Each API key can be scoped to one of the following,

- 1 **Project keys** - Provides access to a single project ([preferred option](#)); access [Project API keys](#) by selecting the specific project you wish to generate keys against.
- 2 **User keys** - Our legacy keys. Provides access to all organizations and all projects that user has been added to; access [API Keys](#) to view your available keys. We highly advise transitioning to project keys for best security practices, although access via this method is currently still supported.

Remember that your API key is a secret! Do not share it with others or expose it in any client-side code (browsers, apps). Production requests must be routed through your own backend server where your API key can be securely loaded from an environment variable or key management service.

All API requests should include your API key in an `Authorization` HTTP header as follows:

`Authorization: Bearer OPENAI_API_KEY`



Organizations and projects (optional)

For users who belong to multiple organizations or are accessing their projects through their legacy user API key, you can pass a header to specify which organization and project is used for an API request. Usage from these API requests will count as usage for the specified organization and project.

To access the `Default project` in an organization, leave out the `OpenAI-Project` header

Example curl command:

```
1 curl https://api.openai.com/v1/models \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "OpenAI-Organization: org-s35FdYxQ2Jzq7TACZwd2kXxg" \
4   -H "OpenAI-Project: $PROJECT_ID"
```



Example with the `openai` Python package:

```
1 from openai import OpenAI
2
3 client = OpenAI(
4     organization='org-s35FdYxQ2Jzq7TACZwd2kXxg',
5     project='$PROJECT_ID',
6 )
```



Example with the `openai` Node.js package:

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI({
4     organization: "org-s35FdYxQ2Jzq7TACZwd2kXxg",
5     project: "$PROJECT_ID",
```



```
6 });
```

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.

Making requests

You can paste the command below into your terminal to run your first API request. Make sure to replace `$OPENAI_API_KEY` with your secret API key. If you are using a legacy user key and you have multiple projects, you will also need to [specify the Project Id](#). For improved security, we recommend transitioning to project based keys instead.

```
1 curl https://api.openai.com/v1/chat/completions \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -d '{
5     "model": "gpt-4o-mini",
6     "messages": [{"role": "user", "content": "Say this is a test!"}],
7     "temperature": 0.7
8   }'
```



This request queries the `gpt-4o-mini` model (which under the hood points to a [gpt-4o-mini model variant](#)) to complete the text starting with a prompt of "Say this is a test". You should get a response back that resembles the following:

```
1 {
2   "id": "chatcmpl-abc123",
3   "object": "chat.completion",
4   "created": 1677858242,
5   "model": "gpt-4o-mini",
6   "usage": {
7     "prompt_tokens": 13,
8     "completion_tokens": 7,
9     "total_tokens": 20,
10    "completion_tokens_details": {
11      "reasoning_tokens": 0
12    }
13  }
14}
```



```
12         }
13     },
14     "choices": [
15       {
16         "message": {
17           "role": "assistant",
18           "content": "\n\nThis is a test!"
19         },
20         "logprobs": null,
21         "finish_reason": "stop",
22         "index": 0
23       }
24     ]
25 }
```

Now that you've generated your first chat completion, let's break down the **response object**. We can see the `finish_reason` is `stop` which means the API returned the full chat completion generated by the model without running into any limits. In the choices list, we only generated a single message but you can set the `n` parameter to generate multiple messages choices.

Streaming

The OpenAI API provides the ability to stream responses back to a client in order to allow partial results for certain requests. To achieve this, we follow the **Server-sent events** standard. Our official **Node** and **Python** libraries include helpers to make parsing these events simpler.

Streaming is supported for both the **Chat Completions API** and the **Assistants API**. This section focuses on how streaming works for Chat Completions. Learn more about how streaming works in the Assistants API [here](#).

In Python, a streaming request looks like:

```
1 from openai import OpenAI
2
3 client = OpenAI()
4
5 stream = client.chat.completions.create(
6   model="gpt-4o-mini",
```



```
7     messages=[{"role": "user", "content": "Say this is a test"}],  
8     stream=True,  
9 )  
10 for chunk in stream:  
11     if chunk.choices[0].delta.content is not None:  
12         print(chunk.choices[0].delta.content, end="")
```

In Node / Typescript, a streaming request looks like:

```
1 import OpenAI from "openai";  
2  
3 const openai = new OpenAI();  
4  
5 async function main() {  
6     const stream = await openai.chat.completions.create({  
7         model: "gpt-4o-mini",  
8         messages: [{ role: "user", content: "Say this is a test" }],  
9         stream: true,  
10    });  
11    for await (const chunk of stream) {  
12        process.stdout.write(chunk.choices[0]?.delta?.content || "");  
13    }  
14 }  
15  
16 main();
```



Parsing Server-sent events

Parsing Server-sent events is non-trivial and should be done with caution. Simple strategies like splitting by a new line may result in parsing errors. We recommend using [existing client libraries](#) when possible.

Debugging requests

In addition to [error codes](#) returned from API responses, it may sometimes be necessary to inspect HTTP response headers as well. Of particular interest will be the headers which contain the unique ID of a particular API request, and 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, which will allow 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.

Request ID in Python

```
1 from openai import OpenAI
2 client = OpenAI()
3
4 response = client.chat.completions.create(
5     messages=[{
6         "role": "user",
7         "content": "Say this is a test",
8     }],
9     model="gpt-4o-mini",
10 )
11
12 print(response._request_id)
```



Request ID in JavaScript

```
1 import OpenAI from 'openai';
2 const client = new OpenAI();
3
4 const response = await client.chat.completions.create({
5   messages: [{ role: 'user', content: 'Say this is a test' }],
6   model: 'gpt-4o-mini'
7 });
8
9 console.log(response._request_id);
```



Access raw response objects in SDKs

If you are using a lower-level HTTP client (like `fetch` or `HttpClient` in C#), you should already have access to response headers as a part of the HTTP interface.

If you are using one of OpenAI's **official SDKs** (which largely abstract the HTTP request/response cycle), you will need to access raw HTTP responses in a slightly different way.

Below is an example of accessing the raw response object (and the `x-ratelimit-limit-tokens` header) using our **Python SDK**.

```
1 from openai import OpenAI
2 client = OpenAI()
3
4 response = client.chat.completions.with_raw_response.create(
5   messages=[{
6     "role": "user",
7     "content": "Say this is a test",
8   }],
9   model="gpt-4o-mini",
10 )
11 print(response.headers.get('x-ratelimit-limit-tokens'))
12
13 # get the object that `chat.completions.create()` would have returned
14 completion = response.parse()
15 print(completion)
```



Here is how you'd access a raw response (and the `x-ratelimit-limit-tokens` header) using

our [JavaScript SDK](#).



```
1 import OpenAI from 'openai';
2 const client = new OpenAI();
3
4 const response = await client.chat.completions.create({
5   messages: [{ role: 'user', content: 'Say this is a test' }],
6   model: 'gpt-4o-mini'
7 }).asResponse();
8
9 // access the underlying Response object
10 console.log(response.headers.get('x-ratelimit-limit-tokens'));
```

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

model string Required

One of the available **TTS models**: [tts-1](#) or [tts-1-hd](#)

input string Required

The text to generate audio for. The maximum length is 4096 characters.

voice string Required

The voice to use when generating the audio. Supported voices are [alloy](#), [echo](#), [fable](#), [onyx](#), [nova](#), 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 ▾



```
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 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** 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  
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) translate, in one of these formats: flac, mp3, mp4, mpeg, mpg, m4a, ogg, wav, or webm.

model string 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 headi
3 }
```

The transcription object (JSON)

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

text string

The transcribed text.

The transcription object (JSON)



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

The transcription object (Verbose JSON)

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

language string

The language of the input audio.

duration string

The duration of the input audio.

text string

The transcribed text.

words array

Extracted words and their corresponding timestamps.

✓ Show properties

segments array

Segments of the transcribed text and their corresponding details.

✓ Show properties

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",
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, 50
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 }
```

Chat

Given a list of messages comprising a conversation, the model will return a response.

Related guide: [Chat Completions](#)

Create chat completion

POST <https://api.openai.com/v1/chat/completions>

Creates a model response for the given chat conversation.

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

ID of the model to use. See the [model endpoint compatibility](#) table for details on which models work with the Chat API.

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.

metadata object or null Optional

Developer-defined tags and values used for filtering completions in the [dashboard](#).

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 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`.

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.

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`.

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`.

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.

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.](#)

response_format object Optional

An object specifying the format that the model must output. Compatible with **GPT-4o**, **GPT-4o mini**, **GPT-4 Turbo** and all GPT-3.5 Turbo models newer than `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

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 null

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.

stream boolean or null Optional Defaults to false

If set, partial message deltas will be sent, like in ChatGPT. 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

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.

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

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

parallel_tool_calls boolean Optional Defaults to true

Whether to enable **parallel function calling** during tool use.

user string Optional

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

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

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-3.5-turbo-0125 ▾ curl ▾



```
1 curl https://api.openai.com/v1/chat/completions \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -d '{
5     "model": "gpt-3.5-turbo-0125",
6     "messages": [
7       {
8         "role": "system",
9         "content": "You are a helpful assistant."
10      },
11      {
12        "role": "user",
13        "content": "Hello!"
14      }
15    ]
16  }'
```

Response



```
1 {
2   "id": "chatcmpl-123",
3   "object": "chat.completion",
4   "created": 1677652288,
5   "model": "gpt-4o-mini",
6   "system_fingerprint": "fp_44709d6fcb",
7   "choices": [
8     {
9       "index": 0,
10      "message": {
11        "role": "assistant",
12        "content": "\n\nHello there, how may I assist you today?",
13      },
14      "logprobs": null,
15      "finish_reason": "stop"
16    }],
17   "usage": {
18     "prompt_tokens": 9,
19     "completion_tokens": 12,
20     "total_tokens": 21,
21     "completion_tokens_details": {
```

```
21     "reasoning_tokens": 0
22   }
23 }
24 }
```

The chat completion object

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

id string

A unique identifier for the chat completion.

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.

model string

The model used for the chat completion.

service_tier string or null

The service tier used for processing the request. This field is only included if the `service_tier` parameter is specified in 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.

object string

The object type, which is always `chat.completion`.

usage object

Usage statistics for the completion request.

▼ Show properties

The chat completion object



```
1  {
2      "id": "chatcmpl-123",
3      "object": "chat.completion",
4      "created": 1677652288,
5      "model": "gpt-4o-mini",
6      "system_fingerprint": "fp_44709d6fcb",
7      "choices": [
8          {
9              "index": 0,
10             "message": {
11                 "role": "assistant",
12                 "content": "\n\nHello there, how may I assist you today?",
13             },
14             "logprobs": null,
15             "finish_reason": "stop"
16         },
17         "usage": {
18             "prompt_tokens": 9,
19             "completion_tokens": 12,
20             "total_tokens": 21,
21             "completion_tokens_details": {
22                 "reasoning_tokens": 0
23             }
24     }
```

The chat completion chunk object

Represents a streamed chunk of a chat completion response returned by model, based on the provided input.

id string

A unique identifier for the chat completion. Each chunk has the same ID.

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.

model string

The model to generate the completion.

service_tier string or null

The service tier used for processing the request. This field is only included if the `service_tier` parameter is specified in 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.

object string

The object type, which is always `chat.completion.chunk`.

usage object

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

The chat completion chunk object



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

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.

▽ 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.

encoding_format string Optional Defaults to float

The format to return the embeddings in. Can be either `float` or `base64`.

dimensions integer Optional

The number of dimensions the resulting output embeddings should have. Only supported in `text-embedding-3` and later models.

user string Optional

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

Returns

A list of [embedding](#) objects.

Example request

curl ▾



```
1 curl https://api.openai.com/v1/embeddings \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
```

```
3   -H "Content-Type: application/json" \
4   -d '{
5     "input": "The food was delicious and the waiter...",
6     "model": "text-embedding-ada-002",
7     "encoding_format": "float"
8   }'
```

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.

index integer

The index of the embedding in the list of embeddings.

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](#).

object string

The object type, which is always "embedding".

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
```

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](#) or [completions](#) format.

See the [fine-tuning guide](#) for more details.

hyperparameters object Optional

The hyperparameters used for the fine-tuning job.

✓ Show properties

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:7p4lUReI`.

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.

integrations array or null Optional

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

✓ 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.

Returns

A [fine-tuning.job](#) object.

[Default](#) [Epochs](#) [Validation file](#) [W&B Integration](#)

Example request

curl ▾



```
1 curl https://api.openai.com/v1/fine_tuning/jobs \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -d '{
5     "training_file": "file-BK7bzQj3FfZFXr7DbL6xJwfo",
6     "model": "gpt-4o-mini"
7   }'
```

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 }
```

List fine-tuning jobs

```
GET https://api.openai.com/v1/fine_tuning/jobs
```

List your organization's fine-tuning jobs

Query parameters

after string Optional

Identifier for the last job from the previous pagination request.

limit integer Optional Defaults to 20

Number of fine-tuning jobs to retrieve.

Returns

A list of paginated **fine-tuning job** objects.

Example request

curl ↴



```
1 curl https://api.openai.com/v1/fine_tuning/jobs?limit=2 \
2     -H "Authorization: Bearer $OPENAI_API_KEY"
```

Response



```
1  {
2      "object": "list",
3      "data": [
4          {
5              "object": "fine_tuning.job.event",
6              "id": "ft-event-TjX01Mf0niCZX64t9PUQT5hn",
7              "created_at": 1689813489,
8              "level": "warn",
9              "message": "Fine tuning process stopping due to job cancellation",
10             "data": null,
11             "type": "message"
12         },
13         { ... },
14         { ... }
15     ], "has_more": true
16 }
```

List fine-tuning events

```
GET https://api.openai.com/v1/fine_tuning/jobs/{fine_tuning_job_id}/events
```

Get status updates for a fine-tuning job.

Path parameters

fine_tuning_job_id string Required

The ID of the fine-tuning job to get events for.

Query parameters

after string Optional

Identifier for the last event from the previous pagination request.

limit integer Optional Defaults to 20

Number of events to retrieve.

Returns

A list of fine-tuning event objects.

Example request

curl ▾



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

Response



```
1  {
2    "object": "list",
3    "data": [
4      {
5        "object": "fine_tuning.job.event",
6        "id": "ft-event-ddTJfwuMVpfLXseOOAmOGqjm",
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
```

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-suf",
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     "object": "fine_tuning.job.checkpoint",
18     "id": "ftckpt_enQCFmOTGj3syEpYVhBRLTSy",
19     "created_at": 1721764800,
20     "fine_tuned_model_checkpoint": "ft:gpt-4o-mini-2024-07-18:my-org:custom-suf",
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_enQCFmOTGj3syEpYVhBRLTSy",
31 "has_more": true
32 }
```

Retrieve fine-tuning job

```
GET https://api.openai.com/v1/fine_tuning/jobs/{fine_tuning_job_id}
```

Get info about a fine-tuning job.

[Learn more about fine-tuning](#)

Path parameters

fine_tuning_job_id string Required

The ID of the fine-tuning job.

Returns

The **fine-tuning** object with the given ID.

Example request

curl ▾



```
1 curl https://api.openai.com/v1/fine_tuning/jobs/ft-AF1WoRqd3aJAHsqc9NY7iL8F \
2   -H "Authorization: Bearer $OPENAI_API_KEY"
```

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   "trained_tokens": 5768,
21   "integrations": [],
22   "seed": 0,
23   "estimated_finish": 0
24 }
```

Cancel fine-tuning

```
POST https://api.openai.com/v1/fine_tuning/jobs/{fine_tuning_job_id}/cancel
```

Immediately cancel a fine-tune job.

Path parameters

fine_tuning_job_id string Required

The ID of the fine-tuning job to cancel.

Returns

The cancelled **fine-tuning** object.

Example request

curl ▾



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

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   "hyperparameters": {
10     "n_epochs": "auto"
11   },
12   "status": "cancelled",
13   "validation_file": "file-abc123",
14   "training_file": "file-abc123"
15 }
```

Training format for chat models

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

messages array

>Show possible types

tools array

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

∨ Show properties

parallel_tool_calls boolean

Whether to enable **parallel function calling** during tool use.

functions Deprecated array

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

∨ Show properties

Training format for chat models



```
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\": \"ce"
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         }
33       }
34     }
35   ]
36 }
```

```
32         "format": { "type": "string", "enum": ["celsius", "fahrenheit"] }
33     },
34     "required": ["location", "format"]
35   }
36 }
37 ]
38 ]
39 }
```

Training format for completions models

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

prompt string

The input prompt for this training example.

completion string

The desired completion for this training example.

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.

id string

The object identifier, which can be referenced in the API endpoints.

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

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. See the [fine-tuning guide](#) for more details.

✗ 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](#).

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](#).

integrations array or null

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

✓ Show possible types

seed integer

The seed used for the fine-tuning job.

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.

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 }
```

The fine-tuning job event object

Fine-tuning job event object

id string

created_at integer

level string

message string

object string

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 }
```

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.

id string

The checkpoint identifier, which can be referenced in the API endpoints.

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.

step_number integer

The step number that the checkpoint was created at.

metrics object

Metrics at the step number during the fine-tuning job.

✓ Show properties

fine_tuning_job_id string

The name of the fine-tuning job that this checkpoint was created from.

object string

The object type, which is always "fine_tuning.job.checkpoint".

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:",
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 }
```

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

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 100 MB in size.

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.

completion_window string Required

The time frame within which the batch should be processed. Currently only `24h` is supported.

metadata object or null Optional

Optional custom metadata for the batch.

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-HOWS94",
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-HOWS94",
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

id string

object string

The object type, which is always `batch`.

endpoint string

The OpenAI API endpoint used by the batch.

errors object

✓ Show properties

input_file_id string

The ID of the input file for the batch.

completion_window string

The time frame within which the batch should be processed.

status string

The current status of the batch.

output_file_id string

The ID of the file containing the outputs of successfully executed requests.

error_file_id string

The ID of the file containing the outputs of requests with errors.

created_at integer

The Unix timestamp (in seconds) for when the batch was created.

in_progress_at integer

The Unix timestamp (in seconds) for when the batch started processing.

expires_at integer

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

finalizing_at integer

The Unix timestamp (in seconds) for when the batch started finalizing.

completed_at integer

The Unix timestamp (in seconds) for when the batch was completed.

failed_at integer

The Unix timestamp (in seconds) for when the batch failed.

expired_at integer

The Unix timestamp (in seconds) for when the batch expired.

cancelling_at integer

The Unix timestamp (in seconds) for when the batch started cancelling.

cancelled_at integer

The Unix timestamp (in seconds) for when the batch was cancelled.

request_counts object

The request counts for different statuses within the batch.

▼ 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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

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-HOWS94",
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.

The request input object



```
{"custom_id": "request-1", "method": "POST", "url": "/v1/chat/completions", "body":
```

The request output object

The per-line object of the batch output and error files

id string

custom_id string

A developer-provided per-request id that will be used to match outputs to inputs.

response object or null

✓ Show properties

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

The request output object



```
{"id": "batch_req_wnaDys", "custom_id": "request-2", "response": {"status_code": 200}}
```

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 100 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.

Use "assistants" for **Assistants** and **Message** files, "vision" for Assistants image file inputs, "batch" for **Batch API**, and "fine-tune" for **Fine-tuning**.

Returns

The uploaded **File** object.

Example request

[curl](#)

```
1 curl https://api.openai.com/v1/files \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -F purpose="fine-tune" \
4   -F file="@mydata.jsonl"
```

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 that belong to the user's organization.

Query parameters

purpose string Optional

Only return files with the given purpose.

Returns

A list of **File** objects.

Example request

curl ▾



```
1 curl https://api.openai.com/v1/files \
2     -H "Authorization: Bearer $OPENAI_API_KEY"
```

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

curl ▾



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

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

[curl ▾](#)

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

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

[curl ▾](#)

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

The file object

The `File` object represents a document that has been uploaded to OpenAI.

id string

The file identifier, which can be referenced in the API endpoints.

bytes integer

The size of the file, in bytes.

created_at integer

The Unix timestamp (in seconds) for when the file was created.

filename string

The name of the file.

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`.

The file object



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

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`s, the correct `mime_type` must be specified. Please refer to documentation for the supported MIME types for your use case:

- **Assistants**

For guidance on the proper filename extensions for each purpose, please follow the documentation on [creating a File](#).

Request body

filename string Required

The name of the file to upload.

purpose string Required

The intended purpose of the uploaded file.

See the [documentation on File purposes](#).

bytes integer Required

The number of bytes in the file you are uploading.

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.

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/part
2 -F data="aHR0cHM6Ly9hcGkub3B1bmFpLmNvbS92MS91cGxvYWRz..."
```

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.

id string

The Upload unique identifier, which can be referenced in API endpoints.

created_at integer

The Unix timestamp (in seconds) for when the Upload was created.

filename string

The name of the file to be uploaded.

bytes integer

The intended number of bytes to be uploaded.

purpose string

The intended purpose of the file. [Please refer here](#) for acceptable values.

status string

The status of the Upload.

expires_at integer

The Unix timestamp (in seconds) for when the Upload was created.

object string

The object type, which is always "upload".

file

The `File` object represents a document that has been uploaded to OpenAI.

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.

id string

The upload Part unique identifier, which can be referenced in API endpoints.

created_at integer

The Unix timestamp (in seconds) for when the Part was created.

upload_id string

The ID of the Upload object that this Part was added to.

object string

The object type, which is always `upload.part`.

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 1024×1024

The size of the generated images. Must be one of `256×256`, `512×512`, or `1024×1024` for `dall-e-2`. Must be one of `1024×1024`, `1792×1024`, or `1024×1792` 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

curl ▾



```
1 curl https://api.openai.com/v1/images/generations \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -d '{
5     "model": "dall-e-3",
6     "prompt": "A cute baby sea otter",
7     "n": 1,
8     "size": "1024x1024"
9   }'
```

Response



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

size string or null Optional Defaults to 1024×1024

The size of the generated images. Must be one of `256×256`, `512×512`, or `1024×1024`.

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.

user string Optional

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

Returns

Returns a list of `image` objects.

Example request

curl ▾



```
1 curl https://api.openai.com/v1/images/edits \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -F image="@otter.png" \
4   -F mask="@mask.png" \
5   -F prompt="A cute baby sea otter wearing a beret" \
6   -F n=2 \
7   -F size="1024×1024"
```

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 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 1024×1024

The size of the generated images. Must be one of `256×256`, `512×512`, or `1024×1024`.

user string Optional

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

Returns

Returns a list of **image** objects.

Example request

curl ▾



```
1 curl https://api.openai.com/v1/images/variations \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -F image="@otter.png" \
4   -F n=2 \
5   -F size="1024x1024"
```

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`.

url string

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

revised_prompt string

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

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

curl ↴



```
1 curl https://api.openai.com/v1/models \
2     -H "Authorization: Bearer $OPENAI_API_KEY"
```

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-3.5-turbo-instruct ▾ curl ▾



```
1 curl https://api.openai.com/v1/models/gpt-3.5-turbo-instruct \
2   -H "Authorization: Bearer $OPENAI_API_KEY"
```

Response

gpt-3.5-turbo-instruct ▾



```
1 {
2   "id": "gpt-3.5-turbo-instruct",
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

curl ▾



```
1 curl https://api.openai.com/v1/models/ft:gpt-4o-mini:acemeco:suffix:abc123 \
2     -X DELETE \
3     -H "Authorization: Bearer $OPENAI_API_KEY"
```

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.

id string

The model identifier, which can be referenced in the API endpoints.

created integer

The Unix timestamp (in seconds) when the model was created.

object string

The object type, which is always "model".

owned_by string

The organization that owns the model.

The model object



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

curl ▾



```
1 curl https://api.openai.com/v1/moderations \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -d '{
5     "input": "I want to kill them."
6   }'
```

Response



```
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

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     "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 }
```

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.

name string or null Optional

The name of the assistant. The maximum length is 256 characters.

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.

tools array Optional Defaults to []

A list of tools 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

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

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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

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.

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.

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

Returns

An **assistant** object.

Code Interpreter

Files

Example request

curl ▾



```
1 curl "https://api.openai.com/v1/assistants" \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -H "OpenAI-Beta: assistants=v2" \
5   -d '{
6     "instructions": "You are a personal math tutor. When asked a question, write
7       \"name\": \"Math Tutor\",
8       \"tools\": [\"type\": \"code_interpreter\"],
9       \"model\": \"gpt-4o"
10    }'
```

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 an
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

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.

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, ending with obj_foo, your subsequent call can include `before=obj_foo` in order to fetch the previous page of the list.

Returns

A list of **assistant** objects.

Example request

curl ▾



```
1 curl "https://api.openai.com/v1/assistants?order=desc&limit=20" \
```

```
2 -H "Content-Type: application/json" \
3 -H "Authorization: Bearer $OPENAI_API_KEY" \
4 -H "OpenAI-Beta: assistants=v2"
```

Response



```
1 {
2   "object": "list",
3   "data": [
4     {
5       "id": "asst_abci23",
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",
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",
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

curl ▾



```
1 curl https://api.openai.com/v1/assistants/asst_abc123 \  
2   -H "Content-Type: application/json" \  
3   -H "Authorization: Bearer $OPENAI_API_KEY" \  
4   -H "OpenAI-Beta: assistants=v2"
```

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 empl
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}
```

Modifies an assistant.

Path parameters

assistant_id string Required

The ID of the assistant to modify.

Request body

model 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.

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.

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

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

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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

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.

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.

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

Returns

The modified `assistant` object.

Example request

curl ▾



```
1 curl https://api.openai.com/v1/assistants/asst_abc123 \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -H "OpenAI-Beta: assistants=v2" \
5   -d '{
6     "instructions": "You are an HR bot, and you have access to files to answer e
7     "tools": [{"type": "file_search"}],
8     "model": "gpt-4o"
9   }'
```

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 empl
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

[curl](#)

```
1 curl https://api.openai.com/v1/assistants/asst_abc123 \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -H "OpenAI-Beta: assistants=v2" \
5   -X DELETE
```

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.

id string

The identifier, which can be referenced in API endpoints.

object string

The object type, which is always `assistant`.

created_at integer

The Unix timestamp (in seconds) for when the assistant was created.

name string or null

The name of the assistant. The maximum length is 256 characters.

description string or null

The description of the assistant. The maximum length is 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.

instructions string or null

The system instructions that the assistant uses. The maximum length is 256,000 characters.

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

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

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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

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.

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.

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

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 an
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

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

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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

Returns

A **thread** object.

[Empty](#) [Messages](#)

Example request

[curl](#) 

```
1 curl https://api.openai.com/v1/threads \  
2   -H "Content-Type: application/json" \  
3   -H "Authorization: Bearer $OPENAI_API_KEY" \  
4   -H "OpenAI-Beta: assistants=v2" \  
5   -d ''
```

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

curl ▾



```
1 curl https://api.openai.com/v1/threads/thread_abc123 \  
2   -H "Content-Type: application/json" \  
3   -H "Authorization: Bearer $OPENAI_API_KEY" \  
4   -H "OpenAI-Beta: assistants=v2"
```

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}

Modifies a thread.

Path parameters

thread_id string Required

The ID of the thread to modify. Only the `metadata` can be modified.

Request body

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

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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

Returns

The modified **thread** object matching the specified ID.

Example request

curl ▾



```
1 curl https://api.openai.com/v1/threads/thread_abc123 \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -H "OpenAI-Beta: assistants=v2" \
5   -d '{
6     "metadata": {
7       "modified": "true",
8       "user": "abc123"
9     }
10   }'
```

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

[curl ▾](#)

```
1 curl https://api.openai.com/v1/threads/thread_abc123 \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -H "OpenAI-Beta: assistants=v2" \
5   -X DELETE
```

Response



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

The thread object Beta

Represents a thread that contains [messages](#).

id string

The identifier, which can be referenced in API endpoints.

object string

The object type, which is always `thread`.

created_at integer

The Unix timestamp (in seconds) for when the thread was created.

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

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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

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
```

Create a message.

Path parameters

thread_id string Required

The ID of the [thread](#) to create a message for.

Request body

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.

content string or array Required

>Show possible types

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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

Returns

A **message** object.

Example request

curl ▾



```
1 curl https://api.openai.com/v1/threads/thread_abc123/messages \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -H "OpenAI-Beta: assistants=v2" \
5   -d '{
6     "role": "user",
7     "content": "How does AI work? Explain it in simple terms."
8   }'
```

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

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

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.

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, ending with obj_foo, your subsequent call can include before=obj_foo in order to fetch the previous page of the list.

`run_id` string Optional

Filter messages by the run ID that generated them.

Returns

A list of `message` objects.

Example request

[curl](#)

```
1 curl https://api.openai.com/v1/threads/thread_abc123/messages \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -H "OpenAI-Beta: assistants=v2"
```

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}
```

Retrieve a message.

Path parameters

thread_id string Required

The ID of the **thread** to which this message belongs.

message_id string Required

The ID of the message to retrieve.

Returns

The **message** object matching the specified ID.

Example request

[curl ▾](#)

```
1 curl https://api.openai.com/v1/threads/thread_abc123/messages/msg_abc123 \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -H "OpenAI-Beta: assistants=v2"
```

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}
```

Modifies a message.

Path parameters

thread_id string Required

The ID of the thread to which this message belongs.

message_id string Required

The ID of the message to modify.

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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

Returns

The modified **message** object.

Example request

curl ▾



```
1 curl https://api.openai.com/v1/threads/thread_abc123/messages/msg_abc123 \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -H "OpenAI-Beta: assistants=v2" \
5   -d '{
6     "metadata": {
7       "modified": "true",
8       "user": "abc123"
9     }
10   }'
```

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

thread_id string Required

The ID of the thread to which this message belongs.

message_id string Required

The ID of the message to delete.

Returns

Deletion status

Example request

curl ▾



```
1 curl -X DELETE https://api.openai.com/v1/threads/thread_abc123/messages/msg_abc123
2   -H "Content-Type: application/json" \
```

```
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -H "OpenAI-Beta: assistants=v2"
```

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**.

id string

The identifier, which can be referenced in API endpoints.

object string

The object type, which is always `thread.message`.

created_at integer

The Unix timestamp (in seconds) for when the message was created.

thread_id string

The **thread** ID that this message belongs to.

status string

The status of the message, which can be either `in_progress`, `incomplete`, or `completed`.

incomplete_details object or null

On an incomplete message, details about why the message is incomplete.

✓ Show properties

completed_at integer or null

The Unix timestamp (in seconds) for when the message was completed.

incomplete_at integer or null

The Unix timestamp (in seconds) for when the message was marked as incomplete.

role string

The entity that produced the message. One of `user` or `assistant`.

content array

The content of the message in array of text and/or images.

▼ Show possible types

assistant_id string or null

If applicable, the ID of the `assistant` that authored this message.

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.

attachments array or null

A list of files attached to the message, and the tools they were added to.

▼ 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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

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 }
```

```
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.

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.

instructions string or null Optional

Overrides the **instructions** of the assistant. This is useful for modifying the behavior on a per-run basis.

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

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

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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

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.

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.

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.

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.

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.

truncation_strategy object 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

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

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

Returns

A **run** object.

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

Example request

curl ▾



```
1 curl https://api.openai.com/v1/threads/thread_abc123/runs \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -H "OpenAI-Beta: assistants=v2" \
5   -d '{
6     "assistant_id": "asst_abc123"
7   }'
```

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.

thread object Optional

✗ Show properties

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.

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.

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.

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

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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

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.

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.

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.

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.

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.

truncation_strategy object 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

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

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

Returns

A `run` object.

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

Example request

curl ↴



```
1 curl https://api.openai.com/v1/threads/runs \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -H "OpenAI-Beta: assistants=v2" \
5   -d '{
6     "assistant_id": "asst_abc123",
7     "thread": {
8       "messages": [
9         {"role": "user", "content": "Explain deep learning to a 5 year old."}
10      ]
11    }
12  }'
```

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
```

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

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.

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, ending with obj_foo, your subsequent call can include `before=obj_foo` in order to fetch the previous page of the list.

Returns

A list of `run` objects.

Example request

[curl](#)

```
1 curl https://api.openai.com/v1/threads/thread_abc123/runs \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -H "OpenAI-Beta: assistants=v2"
```

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 "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

thread_id string Required

The ID of the **thread** that was run.

run_id string Required

The ID of the run to retrieve.

Returns

The `run` object matching the specified ID.

Example request

[curl ↻](#)

```
1 curl https://api.openai.com/v1/threads/thread_abc123/runs/run_abc123 \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "OpenAI-Beta: assistants=v2"
```

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}
```

Modifies a run.

Path parameters

thread_id string Required

The ID of the **thread** that was run.

run_id string Required

The ID of the run to modify.

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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

Returns

The modified **run** object matching the specified ID.

Example request

curl ▾



```
1 curl https://api.openai.com/v1/threads/thread_abc123/runs/run_abc123 \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
```

```
4   -H "OpenAI-Beta: assistants=v2" \
5   -d '{
6     "metadata": {
7       "user_id": "user_abc123"
8     }
9   }'
```

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_output  
s
```

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

thread_id string Required

The ID of the `thread` to which this run belongs.

run_id string Required

The ID of the run that requires the tool output submission.

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

curl ▾



```
1 curl https://api.openai.com/v1/threads/thread_123/runs/run_123/submit_tool_output
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -H "OpenAI-Beta: assistants=v2" \
5   -d '{
6     "tool_outputs": [
7       {
8         "tool_call_id": "call_001",
9         "output": "70 degrees and sunny."
10      }
11    ]
12  }'
```

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
```

Cancels a run that is `in_progress`.

Path parameters

thread_id string Required

The ID of the thread to which this run belongs.

run_id string Required

The ID of the run to cancel.

Returns

The modified **run** object matching the specified ID.

Example request

curl ▾



```
1 curl https://api.openai.com/v1/threads/thread_abc123/runs/run_abc123/cancel \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "OpenAI-Beta: assistants=v2" \
4   -X POST
```

Response



```
1 {
2   "id": "run_abc123",
3   "object": "thread.run",
4   "created_at": 1699076126,
5   "assistant_id": "asst_abc123",
6   "thread_id": "thread_abc123",
7   "status": "cancelling",
8   "started_at": 1699076126,
9   "expires_at": 1699076726,
10  "cancelled_at": null,
11  "failed_at": null,
12  "completed_at": null,
13  "last_error": null,
14  "model": "gpt-4o",
15  "instructions": "You summarize books.",
16  "tools": [
17    {
18      "type": "file_search"
19    }
20  ],
21  "tool_resources": {
```

```
22     "file_search": {  
23         "vector_store_ids": ["vs_123"]  
24     }  
25 },  
26 "metadata": {},  
27 "usage": null,  
28 "temperature": 1.0,  
29 "top_p": 1.0,  
30 "response_format": "auto",  
31 "tool_choice": "auto",  
32 "parallel_tool_calls": true  
33 }
```

The run object Beta

Represents an execution run on a [thread](#).

id string

The identifier, which can be referenced in API endpoints.

object string

The object type, which is always `thread.run`.

created_at integer

The Unix timestamp (in seconds) for when the run was created.

thread_id string

The ID of the [thread](#) that was executed on as a part of this run.

assistant_id string

The ID of the [assistant](#) used for execution of this run.

status string

The status of the run, which can be either `queued`, `in_progress`, `requires_action`, `cancelling`, `cancelled`, `failed`, `completed`, `incomplete`, or `expired`.

required_action object or null

Details on the action required to continue the run. Will be `null` if no action is required.

✗ Show properties

last_error object or null

The last error associated with this run. Will be `null` if there are no errors.

✓ Show properties

expires_at integer or null

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

started_at integer or null

The Unix timestamp (in seconds) for when the run was started.

cancelled_at integer or null

The Unix timestamp (in seconds) for when the run was cancelled.

failed_at integer or null

The Unix timestamp (in seconds) for when the run failed.

completed_at integer or null

The Unix timestamp (in seconds) for when the run was completed.

incomplete_details object or null

Details on why the run is incomplete. Will be `null` if the run is not incomplete.

✓ Show properties

model string

The model that the **assistant** used for this run.

instructions string

The instructions that the **assistant** used for this run.

tools array

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

✓ 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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

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

temperature number or null

The sampling temperature used for this run. If not set, defaults to 1.

top_p number or null

The nucleus sampling value used for this run. If not set, defaults to 1.

max_prompt_tokens integer or null

The maximum number of prompt tokens specified to have been used over the course of the run.

max_completion_tokens integer or null

The maximum number of completion tokens specified to have been used over the course of the run.

truncation_strategy object

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

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

parallel_tool_calls boolean

Whether to enable **parallel function calling** during tool use.

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

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

thread_id string Required

The ID of the thread the run and run steps belong to.

run_id string Required

The ID of the run the run steps belong to.

Query parameters

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.

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, ending 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.

Returns

A list of [run step](#) objects.

Example request

[curl](#) ▾ [Copy](#)

```
1 curl https://api.openai.com/v1/threads/thread_abc123/runs/run_abc123/steps \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -H "OpenAI-Beta: assistants=v2"
```

Response

[Copy](#)

```
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

thread_id string Required

The ID of the thread to which the run and run step belongs.

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.

Query parameters

include[] array Optional

A list of additional fields to include in the response. Currently the only supported value is

`step_details.tool_calls[*].file_search.results[*].content` to fetch the file search result

content.

See the [file search tool documentation](#) for more information.

Returns

The **run step** object matching the specified ID.

Example request

curl ▾



```
1 curl https://api.openai.com/v1/threads/thread_abc123/runs/run_abc123/steps/step_ab
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -H "OpenAI-Beta: assistants=v2"
```

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.

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`.

created_at integer

The Unix timestamp (in seconds) for when the run step was created.

assistant_id string

The ID of the [assistant](#) associated with the run step.

thread_id string

The ID of the [thread](#) that was run.

run_id string

The ID of the [run](#) that this run step is a part of.

type string

The type of run step, which can be either `message_creation` or `tool_calls`.

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

last_error object or null

The last error associated with this run step. Will be `null` if there are no errors.

▼ Show properties

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.

cancelled_at integer or null

The Unix timestamp (in seconds) for when the run step was cancelled.

failed_at integer or null

The Unix timestamp (in seconds) for when the run step failed.

completed_at integer or null

The Unix timestamp (in seconds) for when the run step completed.

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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

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

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 }
```

Vector stores Beta

Vector stores are used to store files for use by the `file_search` tool.

Related guide: [File Search](#)

Create vector store Beta

POST https://api.openai.com/v1/vector_stores

Create a vector store.

Request body

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.

name string Optional

The name of the vector store.

expires_after object Optional

The expiration policy for a vector store.

▼ Show properties

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

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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

Returns

A **vector store** object.

Example request

[curl](#)

```
1 curl https://api.openai.com/v1/vector_stores \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -H "OpenAI-Beta: assistants=v2"
5   -d '{
6     "name": "Support FAQ"
7   }'
```

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 Beta

```
GET https://api.openai.com/v1/vector_stores
```

Returns a list of vector stores.

Query parameters

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.

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, ending with obj_foo, your subsequent call can include `before=obj_foo` in order to fetch the previous page of the list.

Returns

A list of **vector store** objects.

Example request

curl ▾



```
1 curl https://api.openai.com/v1/vector_stores \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -H "OpenAI-Beta: assistants=v2"
```

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 Beta

```
GET https://api.openai.com/v1/vector_stores/{vector_store_id}
```

Retrieves a vector store.

Path parameters

vector_store_id string Required

The ID of the vector store to retrieve.

Returns

The **vector store** object matching the specified ID.

Example request

curl ▾



```
1 curl https://api.openai.com/v1/vector_stores/vs_abc123 \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -H "OpenAI-Beta: assistants=v2"
```

Response



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

Modify vector store Beta

```
POST https://api.openai.com/v1/vector_stores/{vector_store_id}
```

Modifies a vector store.

Path parameters

vector_store_id string Required

The ID of the vector store to modify.

Request body

name string or null Optional

The name of the vector store.

expires_after object 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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

Returns

The modified **vector store** object.

Example request

[curl](#)

```
1 curl https://api.openai.com/v1/vector_stores/vs_abc123 \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -H "OpenAI-Beta: assistants=v2"
5   -d '{
6     "name": "Support FAQ"
7   }'
```

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 Beta

```
DELETE https://api.openai.com/v1/vector_stores/{vector_store_id}
```

Delete a vector store.

Path parameters

vector_store_id string Required

The ID of the vector store to delete.

Returns

Deletion status

Example request

curl ▾



```
1 curl https://api.openai.com/v1/vector_stores/vs_abc123 \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -H "OpenAI-Beta: assistants=v2" \
5   -X DELETE
```

Response



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

The vector store object Beta

A vector store is a collection of processed files can be used by the `file_search` tool.

id string

The identifier, which can be referenced in API endpoints.

object string

The object type, which is always `vector_store`.

created_at integer

The Unix timestamp (in seconds) for when the vector store was created.

name string

The name of the vector store.

usage_bytes integer

The total number of bytes used by the files in the vector store.

file_counts object

>Show properties

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.

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.

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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

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   "metadata": {},  
17   "last_used_at": 1698107661  
18 }
```

Vector store files Beta

Vector store files represent files inside a vector store.

Related guide: [File Search](#)

Create vector store file Beta

```
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.

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

curl ▾



```
1 curl https://api.openai.com/v1/vector_stores/vs_abc123/files \
2     -H "Authorization: Bearer $OPENAI_API_KEY" \
3     -H "Content-Type: application/json" \
4     -H "OpenAI-Beta: assistants=v2" \
5     -d '{
6         "file_id": "file-abc123"
7     }'
```

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 Beta

```
GET https://api.openai.com/v1/vector_stores/{vector_store_id}/files
```

Returns a list of vector store files.

Path parameters

vector_store_id string Required

The ID of the vector store that the files belong to.

Query parameters

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.

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, ending 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`.

Returns

A list of **vector store file** objects.

Example request[curl](#)

```
1 curl https://api.openai.com/v1/vector_stores/vs_abc123/files \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -H "OpenAI-Beta: assistants=v2"
```

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 Beta

```
GET https://api.openai.com/v1/vector_stores/{vector_store_id}/files/{file_id}
```

Retrieves a vector store file.

Path parameters

vector_store_id string Required

The ID of the vector store that the file belongs to.

file_id string Required

The ID of the file being retrieved.

Returns

The **vector store file** object.

Example request

curl ▾



```
1 curl https://api.openai.com/v1/vector_stores/vs_abc123/files/file-abc123 \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -H "OpenAI-Beta: assistants=v2"
```

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 }
```

Delete vector store file Beta

```
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

vector_store_id string Required

The ID of the vector store that the file belongs to.

file_id string Required

The ID of the file to delete.

Returns

Deletion status

Example request

curl ▾



```
1 curl https://api.openai.com/v1/vector_stores/vs_abcd/files/file-abc123 \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -H "OpenAI-Beta: assistants=v2" \
5   -X DELETE
```

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.

id string

The identifier, which can be referenced in API endpoints.

object string

The object type, which is always `vector_store.file`.

usage_bytes integer

The total vector store usage in bytes. Note that this may be different from the original file size.

created_at integer

The Unix timestamp (in seconds) for when the vector store file was created.

vector_store_id string

The ID of the [vector store](#) that the [File](#) is attached to.

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.

last_error object or null

The last error associated with this vector store file. Will be `null` if there are no errors.

✗ Show properties

chunking_strategy object

The strategy used to chunk the file.

✗ Show possible types

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 Beta

Vector store file batches represent operations to add multiple files to a vector store.

Related guide: [File Search](#)

Create vector store file batch Beta

```
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.

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

`curl` ▾



```
1 curl https://api.openai.com/v1/vector_stores/vs_abc123/file_batches \
2     -H "Authorization: Bearer $OPENAI_API_KEY" \
3     -H "Content-Type: application/json" \
4     -H "OpenAI-Beta: assistants=v2" \
5     -d '{
6         "file_ids": ["file-abc123", "file-abc456"]
7     }'
```

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

Beta

```
GET https://api.openai.com/v1/vector_stores/{vector_store_id}/file_batches/{batch_id}
```

Retrieves a vector store file batch.

Path parameters

vector_store_id string Required

The ID of the vector store that the file batch belongs to.

batch_id string Required

The ID of the file batch being retrieved.

Returns

The **vector store file batch** object.

Example request

curl ▾



```
1 curl https://api.openai.com/v1/vector_stores/vs_abc123/files_batches/vsfb_abc123 \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -H "OpenAI-Beta: assistants=v2"
```

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 Beta

```
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

vector_store_id string Required

The ID of the vector store that the file batch belongs to.

batch_id string Required

The ID of the file batch to cancel.

Returns

The modified vector store file batch object.

Example request

curl ▾



```
1 curl https://api.openai.com/v1/vector_stores/vs_abc123/files_batches/vsfb_abc123/c
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -H "OpenAI-Beta: assistants=v2" \
5   -X POST
```

Response



```
1 {
2   "id": "vsfb_abc123",
3   "object": "vector_store.file_batch",
4   "created_at": 1699061776,
5   "vector_store_id": "vs_abc123",
6   "status": "cancelling",
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 Beta

```
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

vector_store_id string Required

The ID of the vector store that the files belong to.

batch_id string Required

The ID of the file batch that the files belong to.

Query parameters

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.

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, ending 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`.

Returns

A list of `vector_store_file` objects.

Example request

[curl](#)

```
1 curl https://api.openai.com/v1/vector_stores/vs_abc123/files_batches/vsfb_abc123/f
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -H "OpenAI-Beta: assistants=v2"
```

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.

id string

The identifier, which can be referenced in API endpoints.

object string

The object type, which is always `vector_store.file_batch`.

created_at integer

The Unix timestamp (in seconds) for when the vector store files batch was created.

vector_store_id string

The ID of the **vector store** that the **File** is attached to.

status string

The status of the vector store files batch, which can be either `in_progress`, `completed`, `cancelled` or `failed`.

file_counts object

▼ Show properties

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 }
```

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.

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`.

delta object

The delta containing the fields that have changed on the Message.

∨ Show properties

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.

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`.

delta object

The delta containing the fields that have changed on the run step.

✓ Show properties

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.

thread.created `data` is a `thread`

Occurs when a new `thread` is created.

thread.run.created `data` is a `run`

Occurs when a new `run` is created.

thread.run.queued `data` is a `run`

Occurs when a `run` moves to a `queued` status.

thread.run.in_progress `data` is a `run`

Occurs when a `run` moves to an `in_progress` status.

thread.run.requires_action `data` is a `run`

Occurs when a `run` moves to a `requires_action` status.

thread.run.completed `data` is a `run`

Occurs when a `run` is completed.

thread.run.incomplete `data` is a `run`

Occurs when a `run` ends with status `incomplete`.

thread.run.failed `[data]` is a `run`

Occurs when a `run` fails.

thread.run.cancelling `[data]` is a `run`

Occurs when a `run` moves to a `cancelling` status.

thread.run.cancelled `[data]` is a `run`

Occurs when a `run` is cancelled.

thread.run.expired `[data]` is a `run`

Occurs when a `run` expires.

thread.run.step.created `[data]` is a `run step`

Occurs when a `run step` is created.

thread.run.step.in_progress `[data]` is a `run step`

Occurs when a `run step` moves to an `in_progress` state.

thread.run.step.delta `[data]` is a `run step delta`

Occurs when parts of a `run step` are being streamed.

thread.run.step.completed `[data]` is a `run step`

Occurs when a `run step` is completed.

thread.run.step.failed `[data]` is a `run step`

Occurs when a `run step` fails.

thread.run.step.cancelled `[data]` is a `run step`

Occurs when a `run step` is cancelled.

thread.run.step.expired `[data]` is a `run step`

Occurs when a `run step` expires.

thread.message.created `[data]` is a `message`

Occurs when a `message` is created.

thread.message.in_progress `[data]` is a `message`

Occurs when a **message** moves to an `in_progress` state.

thread.message.delta `data` is a **message delta**

Occurs when parts of a **Message** are being streamed.

thread.message.completed `data` is a **message**

Occurs when a **message** is completed.

thread.message.incomplete `data` is a **message**

Occurs when a **message** ends before it is completed.

error `data` is an **error**

Occurs when an **error** occurs. This can happen due to an internal server error or a timeout.

done `data` is `[DONE]`

Occurs when a stream ends.

Overview

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](#)

Invites

Invite and manage invitations for an organization. Invited users are automatically added to the Default project.

List invites

```
GET https://api.openai.com/v1/organization/invites
```

Returns a list of invites in the organization.

Query parameters

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.

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.

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

[json](#) 

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

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": "user@example.com",
6     "role": "owner"
7   }'
```

Response

[json](#)

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

json ▾



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

json ▾



The invite object

Represents an individual `invite` to the organization.

object string

The object type, which is always `organization.invite`

id string

The identifier, which can be referenced in API endpoints

email string

The email address of the individual to whom the invite was sent

role string

`owner` or `reader`

status string

`accepted`, `expired`, or `pending`

invited_at integer

The Unix timestamp (in seconds) of when the invite was sent.

expires_at integer

The Unix timestamp (in seconds) of when the invite expires.

accepted_at integer

The Unix timestamp (in seconds) of when the invite was accepted.

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 }
```

Users

Manage users and their role in an organization. Users will be automatically added to the Default project.

List users

```
GET https://api.openai.com/v1/organization/users
```

Lists all of the users in the organization.

Query parameters

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.

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.

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

json ▾



Modify user

```
POST https://api.openai.com/v1/organization/users/{user_id}
```

Modifies a user's role in the organization.

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

json ▾



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

json ▾



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

json ▾



The user object

Represents an individual `user` within an organization.

object string

The object type, which is always `organization.user`

id string

The identifier, which can be referenced in API endpoints

name string

The name of the user

email string

The email address of the user

role string

`owner` or `reader`

added_at integer

The Unix timestamp (in seconds) of when the user was added.

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 or projects. The Default project cannot be modified or archived.

List projects

```
GET https://api.openai.com/v1/organization/projects
```

Returns a list of projects.

Query parameters

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.

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.

Returns

A list of **Project** objects.

Example request

curl ▾



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

Response

json ▾



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

json ▾



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

json ▾



Modify project

```
POST https://api.openai.com/v1/organization/projects/{project_id}
```

Modifies a project in the organization.

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
```

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

json ▾



The project object

Represents an individual project.

id string

The identifier, which can be referenced in API endpoints

object string

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

name string

The name of the project. This appears in reporting.

created_at integer

The Unix timestamp (in seconds) of when the project was created.

archived_at integer or null

The Unix timestamp (in seconds) of when the project was archived or `null`.

status string

`active` or `archived`

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. Users cannot be removed from the Default project, unless they are being removed from the organization.

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

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.

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.

Returns

A list of **ProjectUser** objects.

Example request

curl ▾



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

Response

json ▾



Create project user

```
POST https://api.openai.com/v1/organization/projects/{project_id}/users
```

Adds a user to the project. Users must already be members of the organization to be added to a project.

Path parameters

project_id string Required

The ID of the project.

Request body

user_id string Required

The ID of the user.

role string Required

owner or member

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

json ▾



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

json ▾ 

Modify project user

```
POST https://api.openai.com/v1/organization/projects/{project_id}/users/{user_id}
```

Modifies a user's role in the project.

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_a
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json" \
4   -d '{
5     "role": "owner"
6   }'
```

Response

json ▾



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
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json"
```

Response

json ▾ 

The project user object

Represents an individual user in a project.

object string

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

id string

The identifier, which can be referenced in API endpoints

name string

The name of the user

email string

The email address of the user

role string

`owner` or `member`

added_at integer

The Unix timestamp (in seconds) of when the project was added.

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

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.

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.

Returns

A list of **ProjectServiceAccount** objects.

Example request

curl ▾ 

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

Response

json ▾ 

Create project service account

```
POST https://api.openai.com/v1/organization/projects/{project_id}/service_accounts
```

Creates a new service account in the project. This also returns an unredacted API key for the service account.

Path parameters

project_id string Required

The ID of the project.

Request body

name string Required

The name of the service account being created.

Returns

The created **ProjectServiceAccount** object.

Example request

curl ▾ 

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

Response

json ▾ 

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
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
3   -H "Content-Type: application/json"
```

Response

json ▾



Delete project service account

```
DELETE https://api.openai.com/v1/organization/projects/{project_id}/service_account  
s/{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_ac  
2   -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \  
3   -H "Content-Type: application/json"
```

Response

json ▾



The project service account object

Represents an individual service account in a project.

object string

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

id string

The identifier, which can be referenced in API endpoints

name string

The name of the service account

role string

`owner` or `member`

created_at integer

The Unix timestamp (in seconds) of when the service account was created

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

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.

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.

Returns

A list of [ProjectApiKey](#) objects.

Example request

[curl](#) [Copy](#)

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

Response

[json](#) [Copy](#)

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

project_id string Required

The ID of the project.

key_id string Required

The ID of the API key.

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

json ▾



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

project_id string Required

The ID of the project.

key_id string Required

The ID of the API key.

Returns

Confirmation of the key's deletion or an error if the key belonged to a service account

Example request

curl ▾



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

Response

json ▾



The project API key object

Represents an individual API key in a project.

object string

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

redacted_value string

The redacted value of the API key

name string

The name of the API key

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

owner object

▼ Show properties

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 }
```

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
```

List user actions and configuration changes within this organization.

Query parameters

effective_at object Optional

Return only events whose `effective_at` (Unix seconds) is in this range.

∨ Show properties

project_ids[] array Optional

Return only events for these projects.

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](#).

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.

actor_emails[] array Optional

Return only events performed by users with these emails.

resource_ids[] array Optional

Return only events performed on these targets. For example, a project ID updated.

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.

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, ending with obj_foo, your subsequent call can include `before=obj_foo` in order to fetch the previous page of the list.

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/113.0.5671.135 Safari/537.36"
35                 }
36             },
37             "api_key.updated": {
38                 "id": "key_xxxx",
39                 "data": {
40                     "scopes": ["resource_2.operation_2"]
41                 }
42             },
43         }
44     ],
45     "first_id": "audit_log-xxx__20240101",
```

```
46      "last_id": "audit_log_yyy_20240101",
47      "has_more": true
48 }
```

The audit log object

A log of a user action or configuration change within this organization.

id string

The ID of this log.

type string

The event type.

effective_at integer

The Unix timestamp (in seconds) of the event.

project object

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

∨ Show properties

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.updated object

The details for events with this `type`.

∨ Show properties

api_key.deleted object

The details for events with this `type`.

∨ Show properties

invite.sent object

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

∨ Show properties

invite.accepted object

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

∨ Show properties

invite.deleted 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.created object

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

∨ Show properties

project.updated object

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

∨ Show properties

project.archived 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.updated object

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

∨ Show properties

service_account.deleted object

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

∨ Show properties

user.added object

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

∨ Show properties

user.updated object

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

∨ Show properties

user.deleted object

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

∨ Show properties

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             "user_agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/"
14         }
```

```
15     },
16     "api_key.created": {
17       "id": "key_xxxx",
18       "data": {
19         "scopes": ["resource.operation"]
20       }
21     }
22 }
```

Client events Beta

These are events that the OpenAI Realtime WebSocket server will accept from the client.

[Learn more about the Realtime API.](#)

session.update Beta

Send this event to update the session's default configuration.

event_id string

Optional client-generated ID used to identify this event.

type string

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

session object

Session configuration to update.

▼ Show properties

session.update



```
1  {
2    "event_id": "event_123",
3    "type": "session.update",
4    "session": {
5      "modalities": ["text", "audio"],
6      "instructions": "Your knowledge cutoff is 2023-10. You are a helpful assi
7      "voice": "alloy",
8      "input_audio_format": "pcm16",
9      "output_audio_format": "pcm16",
```

```
10     "input_audio_transcription": {
11         "enabled": true,
12         "model": "whisper-1"
13     },
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         {
22             "type": "function",
23             "name": "get_weather",
24             "description": "Get the current weather for a location.",
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_output_tokens": null
37 }
38 }
```

input_audio_buffer.append Beta

Send this event to append audio bytes to the input audio buffer.

event_id string

Optional client-generated ID used to identify this event.

type string

The event type, must be "input_audio_buffer.append".

audio string

Base64-encoded audio bytes.

input_audio_buffer.append



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

input_audio_buffer.commit Beta

Send this event to commit audio bytes to a user message.

event_id string

Optional client-generated ID used to identify this event.

type string

The event type, must be "input_audio_buffer.commit".

input_audio_buffer.commit



```
1 {  
2     "event_id": "event_789",  
3     "type": "input_audio_buffer.commit"  
4 }
```

input_audio_buffer.clear Beta

Send this event to clear the audio bytes in the buffer.

event_id string

Optional client-generated ID used to identify this event.

type string

The event type, must be "input_audio_buffer.clear".

input_audio_buffer.clear



```
1 {
```

```
2     "event_id": "event_012",
3     "type": "input_audio_buffer.clear"
4 }
```

conversation.item.create Beta

Send this event when adding an item to the conversation.

event_id string

Optional client-generated ID used to identify this event.

type string

The event type, must be "conversation.item.create".

previous_item_id string

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

item object

The item to add to the conversation.

∨ Show properties

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         "status": "completed",
9         "role": "user",
10        "content": [
11            {
12                "type": "input_text",
13                "text": "Hello, how are you?"
14            }
15        ]
16    }
17 }
```

conversation.item.truncate Beta

Send this event when you want to truncate a previous assistant message's audio.

event_id string

Optional client-generated ID used to identify this event.

type string

The event type, must be "conversation.item.truncate".

item_id string

The ID of the assistant message item to truncate.

content_index integer

The index of the content part to truncate.

audio_end_ms integer

Inclusive duration up to which audio is truncated, in milliseconds.

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 Beta

Send this event when you want to remove any item from the conversation history.

event_id string

Optional client-generated ID used to identify this event.

type string

The event type, must be "conversation.item.delete".

item_id string

The ID of the item to delete.

conversation.item.delete



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

response.create Beta

Send this event to trigger a response generation.

event_id string

Optional client-generated ID used to identify this event.

type string

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

response object

Configuration for the response.

✓ Show properties

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": "alloy",
8         "output_audio_format": "pcm16",
9         "tools": [
10            {
11                "type": "function",
12                "name": "calculate_sum",
13                "description": "Calculates the sum of two numbers."
14            }
15        ]
16    }
17}
```

```
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": 0.7,
26 "max_output_tokens": 150
27 }
28 }
```

response.cancel Beta

Send this event to cancel an in-progress response.

event_id string

Optional client-generated ID used to identify this event.

type string

The event type, must be "response.cancel".

response.cancel



```
1 {
2     "event_id": "event_567",
3     "type": "response.cancel"
4 }
```

Server events Beta

These are events emitted from the OpenAI Realtime WebSocket server to the client.

[Learn more about the Realtime API.](#)

error Beta

Returned when an error occurs.

event_id string

The unique ID of the server event.

type string

The event type, must be "error".

error object

Details of the error.

▼ Show properties

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 Beta

Returned when a session is created. Emitted automatically when a new connection is established.

event_id string

The unique ID of the server event.

type string

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

session object

The session resource.

✓ Show properties

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-2024-10-01",
8          "modalities": ["text", "audio"],
9          "instructions": "",
10         "voice": "alloy",
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_output_tokens": null
24     }
25 }
```

session.updated Beta

Returned when a session is updated.

event_id string

The unique ID of the server event.

type string

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

session object

The updated session resource.

✓ Show properties

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-2024-10-01",
8          "modalities": ["text"],
9          "instructions": "New instructions",
10         "voice": "alloy",
11         "input_audio_format": "pcm16",
12         "output_audio_format": "pcm16",
13         "input_audio_transcription": {
14             "enabled": true,
15             "model": "whisper-1"
16         },
17         "turn_detection": {
18             "type": "none"
19         },
20         "tools": [],
21         "tool_choice": "none",
22         "temperature": 0.7,
23         "max_output_tokens": 200
24     }
25 }
```

conversation.created Beta

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

event_id string

The unique ID of the server event.

type string

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

conversation object

The conversation resource.

✓ Show properties

conversation.created



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

input_audio_buffer.committed Beta

Returned when an input audio buffer is committed, either by the client or automatically in server VAD mode.

event_id string

The unique ID of the server event.

type string

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

previous_item_id string

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

item_id string

The ID of the user message item that will be created.

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 Beta

Returned when the input audio buffer is cleared by the client.

event_id string

The unique ID of the server event.

type string

The event type, must be "input_audio_buffer.cleared".

input_audio_buffer.cleared



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

input_audio_buffer.speech_started Beta

Returned in server turn detection mode when speech is detected.

event_id string

The unique ID of the server event.

type string

The event type, must be "input_audio_buffer.speech_started".

audio_start_ms integer

Milliseconds since the session started when speech was detected.

item_id string

The ID of the user message item that will be created when speech stops.

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 Beta

Returned in server turn detection mode when speech stops.

event_id string

The unique ID of the server event.

type string

The event type, must be "input_audio_buffer.speech_stopped".

audio_end_ms integer

Milliseconds since the session started when speech stopped.

item_id string

The ID of the user message item that will be created.

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 }
```

conversation.item.created Beta

Returned when a conversation item is created.

event_id string

The unique ID of the server event.

type string

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

previous_item_id string

The ID of the preceding item.

item object

The item that was created.

▽ Show properties

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": null
15             }
16         ]
17     }
18 }
```

conversation.item.input_audio_transcription.completed Beta

Returned when input audio transcription is enabled and a transcription succeeds.

event_id string

The unique ID of the server event.

type string

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

item_id string

The ID of the user message item.

content_index integer

The index of the content part containing the audio.

transcript string

The transcribed text.

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 Beta

Returned when input audio transcription is configured, and a transcription request for a user message failed.

event_id string

The unique ID of the server event.

type string

The event type, must be "conversation.item.input_audio_transcription.failed".

item_id string

The ID of the user message item.

content_index integer

The index of the content part containing the audio.

error object

Details of the transcription error.

✓ Show properties

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 Beta

Returned when an earlier assistant audio message item is truncated by the client.

event_id string

The unique ID of the server event.

type string

The event type, must be "conversation.item.truncated".

item_id string

The ID of the assistant message item that was truncated.

content_index integer

The index of the content part that was truncated.

audio_end_ms integer

The duration up to which the audio was truncated, in milliseconds.

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 Beta

Returned when an item in the conversation is deleted.

event_id string

The unique ID of the server event.

type string

The event type, must be "conversation.item.deleted".

item_id string

The ID of the item that was deleted.

conversation.item.deleted



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

response.created Beta

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.

type string

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

response object

The response resource.

▼ Show properties

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 Beta

Returned when a Response is done streaming. Always emitted, no matter the final state.

event_id string

The unique ID of the server event.

type string

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

response object

The response resource.

▼ Show properties

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": 50,
26              "input_tokens": 20,
27              "output_tokens": 30
28          }
29      }
30 }
```

response.output_item.added Beta

Returned when a new Item is created during response generation.

event_id string

The unique ID of the server event.

type string

The event type, must be "response.output_item.added".

response_id string

The ID of the response to which the item belongs.

output_index integer

The index of the output item in the response.

item object

The item that was added.

▼ Show properties

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 Beta

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.

type string

The event type, must be "response.output_item.done".

response_id string

The ID of the response to which the item belongs.

output_index integer

The index of the output item in the response.

item object

The completed item.

▼ Show properties

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 Beta

Returned when a new content part is added to an assistant message item during response generation.

event_id string

The unique ID of the server event.

type string

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

response_id string

The ID of the response.

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.

content_index integer

The index of the content part in the item's content array.

part object

The content part that was added.

▼ Show properties

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 Beta

Returned when a content part is done streaming in an assistant message item. Also emitted when a Response is interrupted, incomplete, or cancelled.

event_id string

The unique ID of the server event.

type string

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

response_id string

The ID of the response.

item_id string

The ID of the item.

output_index integer

The index of the output item in the response.

content_index integer

The index of the content part in the item's content array.

part object

The content part that is done.

▼ Show properties

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 Beta

Returned when the text value of a "text" content part is updated.

event_id string

The unique ID of the server event.

type string

The event type, must be "response.text.delta".

response_id string

The ID of the response.

item_id string

The ID of the item.

output_index integer

The index of the output item in the response.

content_index integer

The index of the content part in the item's content array.

delta string

The text delta.

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 Beta

Returned when the text value of a "text" content part is done streaming. Also emitted when a Response is interrupted, incomplete, or cancelled.

event_id string

The unique ID of the server event.

type string

The event type, must be "response.text.done".

response_id string

The ID of the response.

item_id string

The ID of the item.

output_index integer

The index of the output item in the response.

content_index integer

The index of the content part in the item's content array.

text string

The final text content.

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 Beta

Returned when the model-generated transcription of audio output is updated.

event_id string

The unique ID of the server event.

type string

The event type, must be "response.audio_transcript.delta".

response_id string

The ID of the response.

item_id string

The ID of the item.

output_index integer

The index of the output item in the response.

content_index integer

The index of the content part in the item's content array.

delta string

The transcript delta.

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 Beta

Returned when the model-generated transcription of audio output is done streaming. Also emitted when a Response is interrupted, incomplete, or cancelled.

event_id string

The unique ID of the server event.

type string

The event type, must be "response.audio_transcript.done".

response_id string

The ID of the response.

item_id string

The ID of the item.

output_index integer

The index of the output item in the response.

content_index integer

The index of the content part in the item's content array.

transcript string

The final transcript of the audio.

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 Beta

Returned when the model-generated audio is updated.

event_id string

The unique ID of the server event.

type string

The event type, must be "response.audio.delta".

response_id string

The ID of the response.

item_id string

The ID of the item.

output_index integer

The index of the output item in the response.

content_index integer

The index of the content part in the item's content array.

delta string

Base64-encoded audio data delta.

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 Beta

Returned when the model-generated audio is done. Also emitted when a Response is interrupted, incomplete, or cancelled.

event_id string

The unique ID of the server event.

type string

The event type, must be "response.audio.done".

response_id string

The ID of the response.

item_id string

The ID of the item.

output_index integer

The index of the output item in the response.

content_index integer

The index of the content part in the item's content array.

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 Beta

Returned when the model-generated function call arguments are updated.

event_id string

The unique ID of the server event.

type string

The event type, must be "response.function_call_arguments.delta".

response_id string

The ID of the response.

item_id string

The ID of the function call item.

output_index integer

The index of the output item in the response.

call_id string

The ID of the function call.

delta string

The arguments delta as a JSON string.

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 Beta

Returned when the model-generated function call arguments are done streaming. Also emitted when a Response is interrupted, incomplete, or cancelled.

event_id string

The unique ID of the server event.

type string

The event type, must be "response.function_call_arguments.done".

response_id string

The ID of the response.

item_id string

The ID of the function call item.

output_index integer

The index of the output item in the response.

call_id string

The ID of the function call.

arguments string

The final arguments as a JSON string.

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 Beta

Emitted after every "response.done" event to indicate the updated rate limits.

event_id string

The unique ID of the server event.

type string

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

rate_limits array

List of rate limit information.

✓ Show properties

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 }
```

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 ▾ curl ▾



```
1 curl https://api.openai.com/v1/completions \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -d '{
5     "model": "gpt-3.5-turbo-instruct",
6     "prompt": "Say this is a test",
7     "max_tokens": 7,
8     "temperature": 0
9   }'
```

Response

gpt-3.5-turbo-instruct ▾



```
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).

id string

A unique identifier for the completion.

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.

model string

The model used for 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.

object string

The object type, which is always "text_completion"

usage object

Usage statistics for the completion request.

✓ Show properties

The completion object



```
1  {
2      "id": "cmpl-uqkv1QyYK7bGYrRHQOeXlWi7",
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 }
```

Assistants (v1) Legacy

Build assistants that can call models and use tools to perform tasks.

[Get started with the Assistants API](#)

Create assistant (v1) Legacy

```
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. type: string

name string or null Optional

The name of the assistant. The maximum length is 256 characters.

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.

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`, `retrieval`, or `function`.

▼ Show possible types

file_ids array Optional Defaults to []

A list of `file` IDs attached to this assistant. There can be a maximum of 20 files attached to the assistant. Files are ordered by their creation date in ascending order.

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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

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.

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.

response_format string 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_object" }` enables JSON mode, which guarantees 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

Returns

An **assistant** object.

Code Interpreter Files

Example request

curl ▾



```
1 curl "https://api.openai.com/v1/assistants" \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -H "OpenAI-Beta: assistants=v1" \
5   -d '{
6     "instructions": "You are a personal math tutor. When asked a question, write
7       \"name\": \"Math Tutor\",
8       \"tools\": [\"type\": \"code_interpreter\"}],
9       \"model\": \"gpt-4-turbo"
10  }'
```

Response



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

```
17     "temperature": 1.0,  
18     "response_format": "auto"  
19 }
```

Create assistant file (v1) Legacy

```
POST https://api.openai.com/v1/assistants/{assistant_id}/files
```

Create an assistant file by attaching a [File](#) to an [assistant](#).

Path parameters

assistant_id string Required

The ID of the assistant for which to create a File.

Request body

file_id string Required

A [File](#) ID (with `purpose="assistants"`) that the assistant should use. Useful for tools like [retrieval](#) and [code_interpreter](#) that can access files.

Returns

An [assistant file](#) object.

Example request

[curl](#)

```
1 curl https://api.openai.com/v1/assistants/asst_abc123/files \  
2     -H 'Authorization: Bearer $OPENAI_API_KEY' \  
3     -H 'Content-Type: application/json' \  
4     -H 'OpenAI-Beta: assistants=v1' \  
5     -d '{  
6         "file_id": "file-abc123"  
7     }'
```

Response



```
1 {
2   "id": "file-abc123",
3   "object": "assistant.file",
4   "created_at": 1699055364,
5   "assistant_id": "asst_abc123"
6 }
```

List assistants (v1) Legacy

```
GET https://api.openai.com/v1/assistants
```

Returns a list of assistants.

Query parameters

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.

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, ending with obj_foo, your subsequent call can include `before=obj_foo` in order to fetch the previous page of the list.

Returns

A list of **assistant** objects.

Example request

curl ▾



```
1 curl "https://api.openai.com/v1/assistants?order=desc&limit=20" \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -H "OpenAI-Beta: assistants=v1"
```

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-4-turbo",
11      "instructions": "You are a helpful assistant designed to make me better at",
12      "tools": [],
13      "file_ids": [],
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-4-turbo",
26      "instructions": "You are a helpful assistant designed to make me better at",
27      "tools": [],
28      "file_ids": [],
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-4-turbo",
41     "instructions": null,
42     "tools": [],
43     "file_ids": [],
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 }
```

List assistant files (v1) Legacy

```
GET https://api.openai.com/v1/assistants/{assistant_id}/files
```

Returns a list of assistant files.

Path parameters

assistant_id string Required

The ID of the assistant the file belongs to.

Query parameters

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.

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, ending with `obj_foo`, your subsequent call can include `before=obj_foo` in order to fetch the previous page of the list.

Returns

A list of `assistant file` objects.

Example request

[curl](#)

```
1 curl https://api.openai.com/v1/assistants/asst_abc123/files \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -H "OpenAI-Beta: assistants=v1"
```

Response



```
1 {
2   "object": "list",
3   "data": [
4     {
5       "id": "file-abc123",
6       "object": "assistant.file",
7       "created_at": 1699060412,
8       "assistant_id": "asst_abc123"
9     },
10    {
11      "id": "file-abc456",
12      "object": "assistant.file",
13      "created_at": 1699060412,
14      "assistant_id": "asst_abc123"
15    }
16  ],
17  "first_id": "file-abc123",
18  "last_id": "file-abc456",
19  "has_more": false
20 }
```

Retrieve assistant (v1) Legacy

```
GET https://api.openai.com/v1/assistants/{assistant_id}
```

Retrieves an assistant.

Path parameters

assistant_id string Required

The ID of the assistant to retrieve.

Returns

The **assistant** object matching the specified ID.

Example request

curl ▾



```
1 curl https://api.openai.com/v1/assistants/asst_abc123 \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -H "OpenAI-Beta: assistants=v1"
```

Response



```
1 {
2   "id": "asst_abc123",
3   "object": "assistant",
4   "created_at": 1699009709,
5   "name": "HR Helper",
6   "description": null,
7   "model": "gpt-4-turbo",
8   "instructions": "You are an HR bot, and you have access to files to answer empl",
9   "tools": [
10     {
11       "type": "retrieval"
12     }
13   ],
14   "file_ids": [
15     "file-abc123"
16   ],
17   "metadata": {}
18 }
```

Retrieve assistant file (v1) Legacy

```
GET https://api.openai.com/v1/assistants/{assistant_id}/files/{file_id}
```

Retrieves an AssistantFile.

Path parameters

assistant_id string Required

The ID of the assistant who the file belongs to.

file_id string Required

The ID of the file we're getting.

Returns

The **assistant file** object matching the specified ID.

Example request

curl ▾



```
1 curl https://api.openai.com/v1/assistants/asst_abc123/files/file-abc123 \
2   -H 'Authorization: Bearer $OPENAI_API_KEY' \
3   -H 'Content-Type: application/json' \
4   -H 'OpenAI-Beta: assistants=v1'
```

Response



```
1 {
2   "id": "file-abc123",
3   "object": "assistant.file",
4   "created_at": 1699055364,
5   "assistant_id": "asst_abc123"
6 }
```

Modify assistant (v1) Legacy

```
POST https://api.openai.com/v1/assistants/{assistant_id}
```

Modifies an assistant.

Path parameters

assistant_id string Required

The ID of the assistant to modify.

Request body

model 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. type: string

name string or null Optional

The name of the assistant. The maximum length is 256 characters.

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.

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`, `retrieval`, or `function`.

▼ Show possible types

file_ids array Optional Defaults to []

A list of [File](#) IDs attached to this assistant. There can be a maximum of 20 files attached to the assistant. Files are ordered by their creation date in ascending order. If a file was previously attached to the list but does not show up in the list, it will be deleted from the assistant.

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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

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.

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.

response_format string 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_object" }` enables JSON mode, which guarantees 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

Returns

The modified **assistant** object.

Example request

curl ▾



```
1 curl https://api.openai.com/v1/assistants/asst_abc123 \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -H "OpenAI-Beta: assistants=v1" \
5   -d '{
6     "instructions": "You are an HR bot, and you have access to files to answer
7     "tools": [{"type": "retrieval"}],
8     "model": "gpt-4-turbo",
9     "file_ids": ["file-abc123", "file-abc456"]
10   }'
```

Response



```
1  {
2      "id": "asst_abc123",
3      "object": "assistant",
4      "created_at": 1699009709,
5      "name": "HR Helper",
6      "description": null,
7      "model": "gpt-4-turbo",
8      "instructions": "You are an HR bot, and you have access to files to answer empl
9      "tools": [
10         {
11             "type": "retrieval"
12         }
13     ],
14     "file_ids": [
15         "file-abc123",
16         "file-abc456"
17     ],
18     "metadata": {},
19     "top_p": 1.0,
20     "temperature": 1.0,
21     "response_format": "auto"
22 }
```

Delete assistant (v1) Legacy

```
DELETE https://api.openai.com/v1/assistants/{assistant_id}
```

Delete an assistant.

Path parameters

assistant_id string Required

The ID of the assistant to delete.

Returns

Deletion status

Example request

curl ▾



```
1 curl https://api.openai.com/v1/assistants/asst_abc123 \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -H "OpenAI-Beta: assistants=v1" \
5   -X DELETE
```

Response



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

Delete assistant file (v1) Legacy

```
DELETE https://api.openai.com/v1/assistants/{assistant_id}/files/{file_id}
```

Delete an assistant file.

Path parameters

assistant_id string Required

The ID of the assistant that the file belongs to.

file_id string Required

The ID of the file to delete.

Returns

Deletion status

Example request

curl ▾



```
1 curl https://api.openai.com/v1/assistants/asst_abc123/files/file-abc123 \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -H "OpenAI-Beta: assistants=v1" \
```

5 -X DELETE

Response



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

The assistant object (v1) Legacy

Represents an `assistant` that can call the model and use tools.

id string

The identifier, which can be referenced in API endpoints.

object string

The object type, which is always `assistant`.

created_at integer

The Unix timestamp (in seconds) for when the assistant was created.

name string or null

The name of the assistant. The maximum length is 256 characters.

description string or null

The description of the assistant. The maximum length is 512 characters.

model

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. type: string

instructions string or null

The system instructions that the assistant uses. The maximum length is 256,000 characters.

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`, `retrieval`, or `function`.

✓ Show possible types

file_ids array

A list of **file** IDs attached to this assistant. There can be a maximum of 20 files attached to the assistant. Files are ordered by their creation date in ascending order.

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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

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.

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.

response_format string 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_object" }` enables JSON mode, which guarantees 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

The assistant object (v1)



```
1  {
2    "id": "asst_abc123",
3    "object": "assistant",
4    "created_at": 1698984975,
5    "name": "Math Tutor",
6    "description": null,
```

```
7   "model": "gpt-4-turbo",
8   "instructions": "You are a personal math tutor. When asked a question, write an
9   "tools": [
10     {
11       "type": "code_interpreter"
12     }
13   ],
14   "file_ids": [],
15   "metadata": {},
16   "top_p": 1.0,
17   "temperature": 1.0,
18   "response_format": "auto"
19 }
```

The assistant file object (v1) Legacy

A list of **Files** attached to an `assistant`.

id string

The identifier, which can be referenced in API endpoints.

object string

The object type, which is always `assistant.file`.

created_at integer

The Unix timestamp (in seconds) for when the assistant file was created.

assistant_id string

The assistant ID that the file is attached to.

The assistant file object (v1)



```
1 {
2   "id": "file-abc123",
3   "object": "assistant.file",
4   "created_at": 1699055364,
5   "assistant_id": "asst_abc123"
6 }
```

Threads (v1) Legacy

Create threads that assistants can interact with.

Related guide: [Assistants](#)

Create thread (v1) Legacy

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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

Returns

A [thread](#) object.

[Empty](#) [Messages](#)

Example request

curl ▾



```
1 curl https://api.openai.com/v1/threads \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -H "OpenAI-Beta: assistants=v1" \
5   -d ''
```

Response



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

Retrieve thread (v1) Legacy

```
GET https://api.openai.com/v1/threads/{thread_id}
```

Retrieves a thread.

Path parameters

thread_id string Required

The ID of the thread to retrieve.

Returns

The **thread** object matching the specified ID.

Example request

curl ▾



```
1 curl https://api.openai.com/v1/threads/thread_abc123 \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -H "OpenAI-Beta: assistants=v1"
```

Response



```
1 {
2   "id": "thread_abc123",
3   "object": "thread",
4   "created_at": 1699014083,
5   "metadata": {}
```

6 }

Modify thread (v1) Legacy

```
POST https://api.openai.com/v1/threads/{thread_id}
```

Modifies a thread.

Path parameters

thread_id string Required

The ID of the thread to modify. Only the `metadata` can be modified.

Request body

metadata map Optional

Set of 16 key-value pairs that can be attached to an object. This can be useful for storing additional information about the object in a structured format. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

Returns

The modified `thread` object matching the specified ID.

Example request

curl ▾ 

```
1 curl https://api.openai.com/v1/threads/thread_abc123 \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -H "OpenAI-Beta: assistants=v1" \
5   -d '{
6     "metadata": {
7       "modified": "true",
8       "user": "abc123"
9     }
10   }'
```

Response



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

Delete thread (v1) Legacy

```
DELETE https://api.openai.com/v1/threads/{thread_id}
```

Delete a thread.

Path parameters

thread_id string Required

The ID of the thread to delete.

Returns

Deletion status

Example request

curl ▾

```
1 curl https://api.openai.com/v1/threads/thread_abc123 \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -H "OpenAI-Beta: assistants=v1" \
5   -X DELETE
```

Response



```
1 {
```

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

The thread object (v1) Legacy

Represents a thread that contains **messages**.

id string

The identifier, which can be referenced in API endpoints.

object string

The object type, which is always `thread`.

created_at integer

The Unix timestamp (in seconds) for when the thread was created.

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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

The thread object (v1)



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

Messages (v1) Legacy

Create messages within threads

Related guide: [Assistants](#)

Create message (v1) Legacy

```
POST https://api.openai.com/v1/threads/{thread_id}/messages
```

Create a message.

Path parameters

thread_id string Required

The ID of the **thread** to create a message for.

Request body

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.

content string Required

The content of the message.

file_ids array Optional Defaults to []

A list of **File** IDs that the message should use. There can be a maximum of 10 files attached to a message.

Useful for tools like `retrieval` and `code_interpreter` that can access and use 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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

Returns

A **message** object.

Example request

curl ▾



```
1 curl https://api.openai.com/v1/threads/thread_abc123/messages \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -H "OpenAI-Beta: assistants=v1" \
5   -d '{
6     "role": "user",
7     "content": "How does AI work? Explain it in simple terms."
8   }'
```

Response



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

List messages (v1) Legacy

```
GET https://api.openai.com/v1/threads/{thread_id}/messages
```

Returns a list of messages for a given thread.

Path parameters

thread_id string Required

The ID of the **thread** the messages belong to.

Query parameters

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.

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, ending with obj_foo, your subsequent call can include `before=obj_foo` in order to fetch the previous page of the list.

run_id string Optional

Filter messages by the run ID that generated them.

Returns

A list of **message** objects.

Example request

[curl](#)

```
1 curl https://api.openai.com/v1/threads/thread_abc123/messages \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -H "OpenAI-Beta: assistants=v1"
```

Response



```
1  {
2      "object": "list",
3      "data": [
4          {
5              "id": "msg_abc123",
6              "object": "thread.message",
7              "created_at": 1699016383,
8              "thread_id": "thread_abc123",
9              "role": "user",
10             "content": [
11                 {
12                     "type": "text",
13                     "text": {
14                         "value": "How does AI work? Explain it in simple terms.",
15                         "annotations": []
16                     }
17                 }
18             ],
19             "file_ids": [],
20             "assistant_id": null,
21             "run_id": null,
22             "metadata": {}
23         },
24         {
25             "id": "msg_abc456",
26             "object": "thread.message",
27             "created_at": 1699016383,
28             "thread_id": "thread_abc123",
29             "role": "user",
30             "content": [
31                 {
32                     "type": "text",
33                     "text": {
34                         "value": "Hello, what is AI?",
35                         "annotations": []
36                     }
37                 }
38             ],
39             "file_ids": [
40                 "file-abc123"
41             ],
42             "assistant_id": null,
43             "run_id": null,
44             "metadata": {}
45         }
46     ],
47     "first_id": "msg_abc123",
```

```
48     "last_id": "msg_abc456",
49     "has_more": false
50 }
```

List message files (v1) Legacy

```
GET https://api.openai.com/v1/threads/{thread_id}/messages/{message_id}/files
```

Returns a list of message files.

Path parameters

thread_id string Required

The ID of the thread that the message and files belong to.

message_id string Required

The ID of the message that the files belongs to.

Query parameters

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.

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, ending with obj_foo, your subsequent call can include `before=obj_foo` in order to fetch the previous page of the list.

Returns

A list of [message file](#) objects.

Example request

curl ▾ [Copy](#)

```
1 curl https://api.openai.com/v1/threads/thread_abc123/messages/msg_abc123/files \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -H "OpenAI-Beta: assistants=v1"
```

Response

[Copy](#)

```
1 {
2   "object": "list",
3   "data": [
4     {
5       "id": "file-abc123",
6       "object": "thread.message.file",
7       "created_at": 1699061776,
8       "message_id": "msg_abc123"
9     },
10    {
11      "id": "file-abc123",
12      "object": "thread.message.file",
13      "created_at": 1699061776,
14      "message_id": "msg_abc123"
15    }
16  ],
17  "first_id": "file-abc123",
18  "last_id": "file-abc123",
19  "has_more": false
20 }
```

Retrieve message (v1) Legacy

```
GET https://api.openai.com/v1/threads/{thread_id}/messages/{message_id}
```

Retrieve a message.

Path parameters

thread_id string Required

The ID of the **thread** to which this message belongs.

message_id string Required

The ID of the message to retrieve.

Returns

The **message** object matching the specified ID.

Example request

curl ▾



```
1 curl https://api.openai.com/v1/threads/thread_abc123/messages/msg_abc123 \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -H "OpenAI-Beta: assistants=v1"
```

Response



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

Retrieve message file (v1) Legacy

```
GET https://api.openai.com/v1/threads/{thread_id}/messages/{message_id}/files/{file_id}
```

Retrieves a message file.

Path parameters

thread_id string Required

The ID of the thread to which the message and File belong.

message_id string Required

The ID of the message the file belongs to.

file_id string Required

The ID of the file being retrieved.

Returns

The **message file** object.

Example request

curl ▾



```
1 curl https://api.openai.com/v1/threads/thread_abc123/messages/msg_abc123/files/file_id
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -H "OpenAI-Beta: assistants=v1"
```

Response



```
1 {
2   "id": "file-abc123",
3   "object": "thread.message.file",
4   "created_at": 1699061776,
5   "message_id": "msg_abc123"
6 }
```

Modify message (v1) Legacy

```
POST https://api.openai.com/v1/threads/{thread_id}/messages/{message_id}
```

Modifies a message.

Path parameters

thread_id string Required

The ID of the thread to which this message belongs.

message_id string Required

The ID of the message to modify.

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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

Returns

The modified **message** object.

Example request

curl ▾ 

```
1 curl https://api.openai.com/v1/threads/thread_abc123/messages/msg_abc123 \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -H "OpenAI-Beta: assistants=v1" \
5   -d '{
6     "metadata": {
7       "modified": "true",
8       "user": "abc123"
9     }
10   }'
```

Response



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

The message object (v1) Legacy

Represents a message within a **thread**.

id string

The identifier, which can be referenced in API endpoints.

object string

The object type, which is always `thread.message`.

created_at integer

The Unix timestamp (in seconds) for when the message was created.

thread_id string

The **thread** ID that this message belongs to.

status string

The status of the message, which can be either `in_progress`, `incomplete`, or `completed`.

incomplete_details object or null

On an incomplete message, details about why the message is incomplete.

∨ Show properties

completed_at integer or null

The Unix timestamp (in seconds) for when the message was completed.

incomplete_at integer or null

The Unix timestamp (in seconds) for when the message was marked as incomplete.

role string

The entity that produced the message. One of `user` or `assistant`.

content array

The content of the message in array of text and/or images.

∨ Show possible types

assistant_id string or null

If applicable, the ID of the `assistant` that authored this message.

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.

file_ids array

A list of `file` IDs that the assistant should use. Useful for tools like retrieval and code_interpreter that can access files. A maximum of 10 files can be attached to a message.

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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

The message object (v1)



```
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  "file_ids": [],
17  "assistant_id": "asst_abc123",
18  "run_id": "run_abc123",
19  "metadata": {}
20 }
```

The message file object (v1) Legacy

A list of files attached to a `message`.

id string

The identifier, which can be referenced in API endpoints.

object string

The object type, which is always `thread.message.file`.

created_at integer

The Unix timestamp (in seconds) for when the message file was created.

message_id string

The ID of the `message` that the `File` is attached to.

The message file object (v1)



```
1 {
2   "id": "file-abc123",
3   "object": "thread.message.file",
4   "created_at": 1698107661,
5   "message_id": "message_QLoItBbqwyAJEz1Ty4y9k0MM",
```

```
6   "file_id": "file-abc123"  
7 }
```

Runs (v1) Legacy

Represents an execution run on a thread.

Related guide: [Assistants](#)

Create run (v1) Legacy

```
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.

Request body

assistant_id string Required

The ID of the [assistant](#) to use to execute this run.

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.

instructions string or null Optional

Overrides the [instructions](#) of the assistant. This is useful for modifying the behavior on a per-run basis.

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

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

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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

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.

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.

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.

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 `complete`. See `incomplete_details` for more info.

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 `complete`. See `incomplete_details` for more info.

truncation_strategy object Optional

>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 a tool. Specifying a particular tool like `{"type": "TOOL_TYPE"}` or `{"type": "function", "function": {"name": "my_function"}}` forces the model to call that tool.

Show possible types

response_format string 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_object" }` enables JSON mode, which guarantees 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

Returns

A **run** object.

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

Example request

curl ▾



```
1 curl https://api.openai.com/v1/threads/thread_abc123/runs \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -H "OpenAI-Beta: assistants=v1" \
5   -d '{
6     "assistant_id": "asst_abc123"
7   }'
```

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-4-turbo",
15     "instructions": null,
16     "incomplete_details": null,
17     "tools": [
18         {
19             "type": "code_interpreter"
20         }
21     ],
22     "file_ids": [
23         "file-abc123",
24         "file-abc456"
25     ],
26     "metadata": {},
27     "usage": null,
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 }
```

Create thread and run (v1) Legacy

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.

thread object Optional

>Show properties

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.

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.

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.

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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

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.

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.

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.

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 `complete`. See `incomplete_details` for more info.

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 `complete`. See `incomplete_details` for more info.

truncation_strategy object Optional

>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 a tool. Specifying a particular tool like `{"type": "TOOL_TYPE"}` or `{"type": "function", "function": {"name": "my_function"}}` forces the model to call that tool.

Show possible types

response_format string 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_object" }` enables JSON mode, which guarantees 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

Returns

A `run` object.

Default Streaming Streaming with Functions

Example request

curl ▾



```
1 curl https://api.openai.com/v1/threads/runs \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -H "OpenAI-Beta: assistants=v1" \
5   -d '{
6     "assistant_id": "asst_abc123",
7     "thread": {
8       "messages": [
9         {"role": "user", "content": "Explain deep learning to a 5 year old."}
10      ]
11    }
12  }'
```

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  "last_error": null,
14  "model": "gpt-4-turbo",
15  "instructions": "You are a helpful assistant.",
16  "tools": [],
17  "file_ids": [],
18  "metadata": {},
19  "usage": null,
20  "temperature": 1
21 }
```

List runs (v1) Legacy

```
GET https://api.openai.com/v1/threads/{thread_id}/runs
```

Returns a list of runs belonging to a thread.

Path parameters

thread_id string Required

The ID of the thread the run belongs to.

Query parameters

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.

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, ending with obj_foo, your subsequent call can include `before=obj_foo` in order to fetch the previous page of the list.

Returns

A list of `run` objects.

Example request

curl ▾



```
1 curl https://api.openai.com/v1/threads/thread_abc123/runs \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -H "OpenAI-Beta: assistants=v1"
```

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-4-turbo",
18             "instructions": null,
19             "incomplete_details": null,
20             "tools": [
21                 {
22                     "type": "code_interpreter"
23                 }
24             ],
25             "file_ids": [
26                 "file-abc123",
27                 "file-abc456"
28             ],
29             "metadata": {},
30             "usage": {
31                 "prompt_tokens": 123,
32                 "completion_tokens": 456,
33                 "total_tokens": 579
34             },
35             "temperature": 1.0,
36             "top_p": 1.0,
37             "max_prompt_tokens": 1000,
38             "max_completion_tokens": 1000,
39             "truncation_strategy": {
40                 "type": "auto",
41                 "last_messages": null
42             },
43             "response_format": "auto",
44             "tool_choice": "auto"
45         },
46     ]
47 }
```

```
46      {
47        "id": "run_abc456",
48        "object": "thread.run",
49        "created_at": 1699063290,
50        "assistant_id": "asst_abc123",
51        "thread_id": "thread_abc123",
52        "status": "completed",
53        "started_at": 1699063290,
54        "expires_at": null,
55        "cancelled_at": null,
56        "failed_at": null,
57        "completed_at": 1699063291,
58        "last_error": null,
59        "model": "gpt-4-turbo",
60        "instructions": null,
61        "incomplete_details": null,
62        "tools": [
63          {
64            "type": "code_interpreter"
65          }
66        ],
67        "file_ids": [
68          "file-abc123",
69          "file-abc456"
70        ],
71        "metadata": {},
72        "usage": {
73          "prompt_tokens": 123,
74          "completion_tokens": 456,
75          "total_tokens": 579
76        },
77        "temperature": 1.0,
78        "top_p": 1.0,
79        "max_prompt_tokens": 1000,
80        "max_completion_tokens": 1000,
81        "truncation_strategy": {
82          "type": "auto",
83          "last_messages": null
84        },
85        "response_format": "auto",
86        "tool_choice": "auto"
87      }
88    ],
89    "first_id": "run_abc123",
90    "last_id": "run_abc456",
91    "has_more": false
92  }
```

List run steps (v1) Legacy

```
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

thread_id string Required

The ID of the thread the run and run steps belong to.

run_id string Required

The ID of the run the run steps belong to.

Query parameters

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.

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, ending with obj_foo, your subsequent call can include `before=obj_foo` in order to fetch the previous page of the list.

Returns

A list of **run step** objects.

Example request

curl ▾ 

```
1 curl https://api.openai.com/v1/threads/thread_abc123/runs/run_abc123/steps \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -H "OpenAI-Beta: assistants=v1"
```

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 (v1) Legacy

```
GET https://api.openai.com/v1/threads/{thread_id}/runs/{run_id}
```

Retrieves a run.

Path parameters

thread_id string Required

The ID of the **thread** that was run.

run_id string Required

The ID of the run to retrieve.

Returns

The **run** object matching the specified ID.

Example request

curl ▾



```
1 curl https://api.openai.com/v1/threads/thread_abc123/runs/run_abc123 \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "OpenAI-Beta: assistants=v1"
```

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-4-turbo",
15     "instructions": null,
16     "incomplete_details": null,
17     "tools": [
18       {
19         "type": "code_interpreter"
20       }
21     ],
22     "file_ids": [
23       "file-abc123",
24       "file-abc456"
25     ],
26     "metadata": {},
27     "usage": {
28       "prompt_tokens": 123,
29       "completion_tokens": 456,
30       "total_tokens": 579
31     },
32     "temperature": 1.0,
33     "top_p": 1.0,
34     "max_prompt_tokens": 1000,
35     "max_completion_tokens": 1000,
36     "truncation_strategy": {
37       "type": "auto",
38       "last_messages": null
39     },
40     "response_format": "auto",
41     "tool_choice": "auto"
42 }
```

Retrieve run step (v1) Legacy

```
GET https://api.openai.com/v1/threads/{thread_id}/runs/{run_id}/steps/{step_id}
```

Retrieves a run step.

Path parameters

thread_id string Required

The ID of the thread to which the run and run step belongs.

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.

Returns

The **run step** object matching the specified ID.

Example request

[curl](#)

```
1 curl https://api.openai.com/v1/threads/thread_abc123/runs/run_abc123/steps/step_ab
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -H "OpenAI-Beta: assistants=v1"
```

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 }
```

Modify run (v1) Legacy

```
POST https://api.openai.com/v1/threads/{thread_id}/runs/{run_id}
```

Modifies a run.

Path parameters

thread_id string Required

The ID of the **thread** that was run.

run_id string Required

The ID of the run to modify.

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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

Returns

The modified **run** object matching the specified ID.

Example request

curl ▾



```
1 curl https://api.openai.com/v1/threads/thread_abc123/runs/run_abc123 \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -H "OpenAI-Beta: assistants=v1" \
5   -d '{
6     "metadata": {
7       "user_id": "user_abc123"
```

```
8     }
9 }
```

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-4-turbo",
15  "instructions": null,
16  "incomplete_details": null,
17  "tools": [
18    {
19      "type": "code_interpreter"
20    }
21  ],
22  "file_ids": [
23    "file-abc123",
24    "file-abc456"
25  ],
26  "metadata": {
27    "user_id": "user_abc123"
28  },
29  "usage": {
30    "prompt_tokens": 123,
31    "completion_tokens": 456,
32    "total_tokens": 579
33  },
34  "temperature": 1.0,
35  "top_p": 1.0,
36  "max_prompt_tokens": 1000,
37  "max_completion_tokens": 1000,
38  "truncation_strategy": {
39    "type": "auto",
40    "last_messages": null
41  },
```

```
42   "response_format": "auto",
43   "tool_choice": "auto"
44 }
```

Submit tool outputs to run (v1) Legacy

POST https://api.openai.com/v1/threads/{thread_id}/runs/{run_id}/submit_tool_output

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

thread_id string Required

The ID of the `thread` to which this run belongs.

run_id string Required

The ID of the run that requires the tool output submission.

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

curl ▾

```
1 curl https://api.openai.com/v1/threads/thread_123/runs/run_123/submit_tool_output
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "Content-Type: application/json" \
4   -H "OpenAI-Beta: assistants=v1" \
5   -d '{
6     "tool_outputs": [
7       {
8         "tool_call_id": "call_001",
9         "output": "70 degrees and sunny."
10      }
11    ]
12 }'
```

Response

2

```
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-4-turbo",
15     "instructions": null,
16     "incomplete_details": null,
17     "tools": [
18         {
19             "type": "function",
20             "function": {
21                 "name": "get_current_weather",
22                 "description": "Get the current weather in a given location",
23                 "parameters": {
24                     "type": "object",
25                     "properties": {
26                         "location": {
27                             "type": "string",
28                             "description": "The city and state, e.g. San Francisco, CA"
29                         }
29                     }
29                 }
29             }
29         }
29     ]
29 }
```

```
30         "unit": {
31             "type": "string",
32             "enum": ["celsius", "fahrenheit"]
33         }
34     },
35     "required": ["location"]
36   }
37 }
38 ]
39 ],
40 "file_ids": [],
41 "metadata": {},
42 "usage": null,
43 "temperature": 1.0,
44 "top_p": 1.0,
45 "max_prompt_tokens": 1000,
46 "max_completion_tokens": 1000,
47 "truncation_strategy": {
48     "type": "auto",
49     "last_messages": null
50 },
51 "response_format": "auto",
52 "tool_choice": "auto"
53 }
```

Cancel a run (v1) Legacy

POST https://api.openai.com/v1/threads/{thread_id}/runs/{run_id}/cancel

Cancels a run that is `in_progress`.

Path parameters

thread_id string Required

The ID of the thread to which this run belongs.

run_id string Required

The ID of the run to cancel.

Returns

The modified `run` object matching the specified ID.

Example request

[curl ▾](#)

```
1 curl https://api.openai.com/v1/threads/thread_abc123/runs/run_abc123/cancel \
2   -H "Authorization: Bearer $OPENAI_API_KEY" \
3   -H "OpenAI-Beta: assistants=v1" \
4   -X POST
```

Response



```
1 {
2   "id": "run_abc123",
3   "object": "thread.run",
4   "created_at": 1699076126,
5   "assistant_id": "asst_abc123",
6   "thread_id": "thread_abc123",
7   "status": "cancelling",
8   "started_at": 1699076126,
9   "expires_at": 1699076726,
10  "cancelled_at": null,
11  "failed_at": null,
12  "completed_at": null,
13  "last_error": null,
14  "model": "gpt-4-turbo",
15  "instructions": "You summarize books.",
16  "tools": [
17    {
18      "type": "retrieval"
19    }
20  ],
21  "file_ids": [],
22  "metadata": {},
23  "usage": null,
24  "temperature": 1.0,
25  "top_p": 1.0,
26 }
```

The run object (v1) Legacy

Represents an execution run on a `thread`.

id string

The identifier, which can be referenced in API endpoints.

object string

The object type, which is always `thread.run`.

created_at integer

The Unix timestamp (in seconds) for when the run was created.

thread_id string

The ID of the `thread` that was executed on as a part of this run.

assistant_id string

The ID of the `assistant` used for execution of this run.

status string

The status of the run, which can be either `queued`, `in_progress`, `requires_action`, `cancelling`, `cancelled`, `failed`, `completed`, or `expired`.

required_action object or null

Details on the action required to continue the run. Will be `null` if no action is required.

✗ Show properties

last_error object or null

The last error associated with this run. Will be `null` if there are no errors.

✗ Show properties

expires_at integer or null

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

started_at integer or null

The Unix timestamp (in seconds) for when the run was started.

cancelled_at integer or null

The Unix timestamp (in seconds) for when the run was cancelled.

failed_at integer or null

The Unix timestamp (in seconds) for when the run failed.

completed_at integer or null

The Unix timestamp (in seconds) for when the run was completed.

incomplete_details object or null

Details on why the run is incomplete. Will be `null` if the run is not incomplete.

∨ Show properties

model string

The model that the **assistant** used for this run.

instructions string

The instructions that the **assistant** used for this run.

tools array

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

∨ Show possible types

file_ids array

The list of **File** IDs the **assistant** used for this 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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

usage**temperature** number or null

The sampling temperature used for this run. If not set, defaults to 1.

top_p number or null

The nucleus sampling value used for this run. If not set, defaults to 1.

max_prompt_tokens integer or null

The maximum number of prompt tokens specified to have been used over the course of the run.

max_completion_tokens integer or null

The maximum number of completion tokens specified to have been used over the course of the run.

truncation_strategy object

✓ Show properties

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 a tool. Specifying a particular tool like `{"type": "TOOL_TYPE"}` or `{"type": "function", "function": {"name": "my_function"}}` forces the model to call that tool.

✓ Show possible types

response_format string 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_object" }` enables JSON mode, which guarantees 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

The run object (v1)



```
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-4-turbo",
15   "instructions": null,
16   "tools": [{"type": "retrieval"}, {"type": "code_interpreter"}],
17   "file_ids": [],
18   "metadata": {},
19   "incomplete_details": null,
```

```
20 "usage": {  
21     "prompt_tokens": 123,  
22     "completion_tokens": 456,  
23     "total_tokens": 579  
24 },  
25 "temperature": 1.0,  
26 "top_p": 1.0,  
27 "max_prompt_tokens": 1000,  
28 "max_completion_tokens": 1000,  
29 "truncation_strategy": {  
30     "type": "auto",  
31     "last_messages": null  
32 },  
33 "response_format": "auto",  
34 "tool_choice": "auto"  
35 }
```

The run step object (v1) Legacy

Represents a step in execution of a run.

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`.

created_at integer

The Unix timestamp (in seconds) for when the run step was created.

assistant_id string

The ID of the **assistant** associated with the run step.

thread_id string

The ID of the **thread** that was run.

run_id string

The ID of the **run** that this run step is a part of.

type string

The type of run step, which can be either `message_creation` or `tool_calls`.

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

last_error object or null

The last error associated with this run step. Will be `null` if there are no errors.

✓ Show properties

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.

cancelled_at integer or null

The Unix timestamp (in seconds) for when the run step was cancelled.

failed_at integer or null

The Unix timestamp (in seconds) for when the run step failed.

completed_at integer or null

The Unix timestamp (in seconds) for when the run step completed.

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. Keys can be a maximum of 64 characters long and values can be a maximum of 512 characters long.

usage

The run step object (v1)



```
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 (v1) Legacy

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 (v1) Legacy

Represents a message delta i.e. any changed fields on a message during streaming.

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`.

delta object

The delta containing the fields that have changed on the Message.

∨ Show properties

The message delta object (v1)



```
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 (v1) Legacy

Represents a run step delta i.e. any changed fields on a run step during streaming.

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`.

delta object

The delta containing the fields that have changed on the run step.

∨ Show properties

The run step delta object (v1)



```
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 (v1) Legacy

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.

thread.created `data` is a `thread`

Occurs when a new `thread` is created.

thread.run.created `data` is a `run`

Occurs when a new **run** is created.

thread.run.queued `[data]` is a **run**

Occurs when a **run** moves to a `queued` status.

thread.run.in_progress `[data]` is a **run**

Occurs when a **run** moves to an `in_progress` status.

thread.run.requires_action `[data]` is a **run**

Occurs when a **run** moves to a `requires_action` status.

thread.run.completed `[data]` is a **run**

Occurs when a **run** is completed.

thread.run.failed `[data]` is a **run**

Occurs when a **run** fails.

thread.run.cancelling `[data]` is a **run**

Occurs when a **run** moves to a `cancelling` status.

thread.run.cancelled `[data]` is a **run**

Occurs when a **run** is cancelled.

thread.run.expired `[data]` is a **run**

Occurs when a **run** expires.

thread.run.step.created `[data]` is a **run step**

Occurs when a **run step** is created.

thread.run.step.in_progress `[data]` is a **run step**

Occurs when a **run step** moves to an `in_progress` state.

thread.run.step.delta `[data]` is a **run step delta**

Occurs when parts of a **run step** are being streamed.

thread.run.step.completed `[data]` is a [run step](#)

Occurs when a [run step](#) is completed.

thread.run.step.failed `[data]` is a [run step](#)

Occurs when a [run step](#) fails.

thread.run.step.cancelled `[data]` is a [run step](#)

Occurs when a [run step](#) is cancelled.

thread.run.step.expired `[data]` is a [run step](#)

Occurs when a [run step](#) expires.

thread.message.created `[data]` is a [message](#)

Occurs when a [message](#) is created.

thread.message.in_progress `[data]` is a [message](#)

Occurs when a [message](#) moves to an `[in_progress]` state.

thread.message.delta `[data]` is a [message delta](#)

Occurs when parts of a [Message](#) are being streamed.

thread.message.completed `[data]` is a [message](#)

Occurs when a [message](#) is completed.

thread.message.incomplete `[data]` is a [message](#)

Occurs when a [message](#) ends before it is completed.

error `[data]` is an [error](#)

Occurs when an [error](#) occurs. This can happen due to an internal server error or a timeout.

done `[data]` is `[DONE]`

Occurs when a stream ends.