OpenAI Platform

# 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:

```
curl https://api.openai.com/v1/models \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "OpenAI-Organization: org-ebXo5XY3YfwYOkuUeOksAWVu" \
  -H "OpenAI-Project: $PROJECT_ID"
```

Example with the `openai` Python package:

```
from openai import OpenAI

client = OpenAI(
  organization='org-ebXo5XY3YfwYOkuUeOksAWVu',
  project='$PROJECT_ID',
)
```

Example with the `openai` Node.js package:

```
import OpenAI from "openai";

const openai = new OpenAI({
    organization: "org-ebXo5XY3YfwYOkuUeOksAWVu",
    project: "$PROJECT_ID",
});
```

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.

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

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:

```
{
    "id": "chatcmpl-abc123",
    "object": "chat.completion",
    "created": 1677858242,
    "model": "gpt-4o-mini",
    "usage": {
        "prompt_tokens": 13,
        "completion_tokens": 7,
        "total_tokens": 20,
        "completion_tokens_details": {
            "reasoning_tokens": 0
        }
    },
    "choices": [
        {
            "message": {
                "role": "assistant",
                "content": "\n\nThis is a test!"
            },
            "logprobs": null,
            "finish_reason": "stop",
            "index": 0
        }
```

```
    ]
}
```

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:

```python
from openai import OpenAI

client = OpenAI()

stream = client.chat.completions.create(
    model="gpt-4o-mini",
    messages=[{"role": "user", "content": "Say this is a test"}],
    stream=True,
)
for chunk in stream:
    if chunk.choices[0].delta.content is not None:
        print(chunk.choices[0].delta.content, end="")
```

In Node / Typescript, a streaming request looks like:

```javascript
import OpenAI from "openai";

const openai = new OpenAI();

async function main() {
    const stream = await openai.chat.completions.create({
```

```
        model: "gpt-4o-mini",
        messages: [{ role: "user", content: "Say this is a test" }],
        stream: true,
    });
    for await (const chunk of stream) {
        process.stdout.write(chunk.choices[0]?.delta?.content || "");
    }
}

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**

```python
from openai import OpenAI
client = OpenAI()

response = client.chat.completions.create(
    messages=[{
        "role": "user",
        "content": "Say this is a test",
    }],
    model="gpt-4o-mini",
)

print(response._request_id)
```

**Request ID in JavaScript**

```javascript
import OpenAI from 'openai';
const client = new OpenAI();

const response = await client.chat.completions.create({
    messages: [{ role: 'user', content: 'Say this is a test' }],
    model: 'gpt-4o-mini'
});

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**.

```python
from openai import OpenAI
client = OpenAI()

response = client.chat.completions.with_raw_response.create(
    messages=[{
        "role": "user",
        "content": "Say this is a test",
    }],
    model="gpt-4o-mini",
)
print(response.headers.get('x-ratelimit-limit-tokens'))

# get the object that `chat.completions.create()` would have returned
completion = response.parse()
print(completion)
```

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

```javascript
import OpenAI from 'openai';
const client = new OpenAI();

const response = await client.chat.completions.create({
    messages: [{ role: 'user', content: 'Say this is a test' }],
    model: 'gpt-4o-mini'
}).asResponse();

// access the underlying Response object
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

```curl
curl https://api.openai.com/v1/audio/speech \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "model": "tts-1",
    "input": "The quick brown fox jumped over the lazy dog.",
    "voice": "alloy"
  }' \
  --output speech.mp3
```

# Create transcription

`POST https://api.openai.com/v1/audio/transcriptions`

Transcribes audio into the input language.

## Request body

**`file`** file Required
The audio file object (not file name) to transcribe, in one of these formats: flac, mp3, mp4, mpeg, mpga, m4a, ogg, wav, or webm.

**`model`** string Required
ID of the model to use. 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

```
curl https://api.openai.com/v1/audio/transcriptions \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -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, mpga, 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

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

Response

```
{
  "text": "Hello, my name is Wolfgang and I come from Germany. Where are you headi
}
```

## 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)

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

## 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)


```
{
  "task": "transcribe",
  "language": "english",
  "duration": 8.470000267028809,
  "text": "The beach was a popular spot on a hot summer day. People were swimming
  "segments": [
    {
      "id": 0,
      "seek": 0,
      "start": 0.0,
      "end": 3.319999933242798,
      "text": " The beach was a popular spot on a hot summer day.",
      "tokens": [
        50364, 440, 7534, 390, 257, 3743, 4008, 322, 257, 2368, 4266, 786, 13, 50
      ],
      "temperature": 0.0,
      "avg_logprob": -0.2860786020755768,
      "compression_ratio": 1.2363636493682861,
      "no_speech_prob": 0.00985979475080967
    },
    ...
  ]
}
```


# 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. Learn more in the **text generation**, **vision**, and **audio** guides.

## 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.

---

**modalities** array or null Optional
Output types that you would like the model to generate for this request. Most models are capable of generating text, which is the default:

`["text"]`

The `gpt-4o-audio-preview` model can also be used to **generate audio**. To request that this model generate both text and audio responses, you can use:

`["text", "audio"]`

---

**audio** object or null Optional
Parameters for audio output. Required when audio output is requested with `modalities: ["audio"]`. **Learn more**.

Show properties

---

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

> If set to 'auto', and the Project is Scale tier enabled, the system will utilize scale tier credits until they are exhausted.
>
> If set to 'auto', and the Project is not Scale tier enabled, the request will be processed using the default service tier with a lower uptime SLA and no latency guarentee.
>
> If set to 'default', the request will be processed using the default service tier with a lower uptime SLA and no latency guarentee.
>
> 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-4o curl

```bash
curl https://api.openai.com/v1/chat/completions \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -d '{
    "model": "gpt-4o",
    "messages": [
      {
        "role": "system",
        "content": "You are a helpful assistant."
      },
      {
        "role": "user",
        "content": "Hello!"
      }
    ]
  }'
```

Response

```json
{
  "id": "chatcmpl-123",
  "object": "chat.completion",
  "created": 1677652288,
  "model": "gpt-4o-mini",
  "system_fingerprint": "fp_44709d6fcb",
  "choices": [{
    "index": 0,
    "message": {
      "role": "assistant",
      "content": "\n\nHello there, how may I assist you today?",
    },
    "logprobs": null,
    "finish_reason": "stop"
  }],
  "usage": {
    "prompt_tokens": 9,
    "completion_tokens": 12,
    "total_tokens": 21,
    "completion_tokens_details": {
      "reasoning_tokens": 0
    }
  }
}
```

## 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

```
{
  "id": "chatcmpl-123456",
  "object": "chat.completion",
  "created": 1728933352,
  "model": "gpt-4o-2024-08-06",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "Hi there! How can I assist you today?",
        "refusal": null
      },
      "logprobs": null,
```

```
      "finish_reason": "stop"
    }
  ],
  "usage": {
    "prompt_tokens": 19,
    "completion_tokens": 10,
    "total_tokens": 29,
    "prompt_tokens_details": {
      "cached_tokens": 0
    },
    "completion_tokens_details": {
      "reasoning_tokens": 0
    }
  },
  "system_fingerprint": "fp_6b68a8204b"
}
```


## 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

```curl
curl https://api.openai.com/v1/embeddings \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "input": "The food was delicious and the waiter...",
    "model": "text-embedding-ada-002",
    "encoding_format": "float"
  }'
```

Response

```
{
  "object": "list",
  "data": [
    {
      "object": "embedding",
      "embedding": [
        0.0023064255,
        -0.009327292,
        .... (1536 floats total for ada-002)
        -0.0028842222,
      ],
      "index": 0
    }
  ],
  "model": "text-embedding-ada-002",
```

```
  "usage": {
    "prompt_tokens": 8,
    "total_tokens": 8
  }
}
```

## The embedding object

Represents an embedding vector returned by embedding endpoint.

**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

```
{
  "object": "embedding",
  "embedding": [
    0.0023064255,
    -0.009327292,
    .... (1536 floats total for ada-002)
    -0.0028842222,
  ],
  "index": 0
}
```

## 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:7p4lURel`.

**`validation_file`** string or null Optional
The ID of an uploaded file that contains validation data.

If you provide this file, the data is used to generate validation metrics periodically during fine-tuning. These metrics can be viewed in the fine-tuning results file. The same data should not be present in both train and validation files.

Your dataset must be formatted as a JSONL file. You must upload your file with the purpose `fine-tune`.

See the **fine-tuning guide** for more details.

**`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

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

Response

```
{
  "object": "fine_tuning.job",
  "id": "ftjob-abc123",
  "model": "gpt-4o-mini-2024-07-18",
  "created_at": 1721764800,
  "fine_tuned_model": null,
  "organization_id": "org-123",
  "result_files": [],
  "status": "queued",
  "validation_file": null,
  "training_file": "file-abc123",
}
```

# 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

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

Response


```
{
  "object": "list",
  "data": [
    {
      "object": "fine_tuning.job.event",
      "id": "ft-event-TjXOlMfOniCZX64t9PUQT5hn",
      "created_at": 1689813489,
      "level": "warn",
      "message": "Fine tuning process stopping due to job cancellation",
      "data": null,
      "type": "message"
    },
    { ... },
    { ... }
  ], "has_more": true
}
```


# List fine-tuning events

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

Get status updates for a fine-tuning job.

## Path parameters

**fine_tuning_job_id** string Required
The ID of the fine-tuning job to get events for.

## Query parameters

**after** string Optional
Identifier for the last event from the previous pagination request.

**limit** integer Optional Defaults to 20
Number of events to retrieve.

## Returns

A list of fine-tuning event objects.

Example request curl

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

Response


```
{
  "object": "list",
  "data": [
    {
      "object": "fine_tuning.job.event",
      "id": "ft-event-ddTJfwuMVpfLXseOOAmOGqjm",
      "created_at": 1721764800,
      "level": "info",
      "message": "Fine tuning job successfully completed",
      "data": null,
      "type": "message"
    },
    {
      "object": "fine_tuning.job.event",
      "id": "ft-event-tyiGuB72evQncpH87xe505Sv",
      "created_at": 1721764800,
      "level": "info",
      "message": "New fine-tuned model created: ft:gpt-4o-mini:openai::7p4lURel",
      "data": null,
      "type": "message"
    }
```

```
  ],
  "has_more": true
}
```

# List fine-tuning checkpoints

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

List checkpoints for a fine-tuning job.

## Path parameters

**`fine_tuning_job_id`** string Required
The ID of the fine-tuning job to get checkpoints for.

## Query parameters

**`after`** string Optional
Identifier for the last checkpoint ID from the previous pagination request.

**`limit`** integer Optional Defaults to 10
Number of checkpoints to retrieve.

## Returns

A list of fine-tuning **checkpoint objects** for a fine-tuning job.

Example request curl

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

Response

```
{
  "object": "list"
  "data": [
    {
      "object": "fine_tuning.job.checkpoint",
      "id": "ftckpt_zc4Q7MP6XxulcVzj4MZdwsAB",
      "created_at": 1721764867,
```

```
      "fine_tuned_model_checkpoint": "ft:gpt-4o-mini-2024-07-18:my-org:custom-suf
      "metrics": {
        "full_valid_loss": 0.134,
        "full_valid_mean_token_accuracy": 0.874
      },
      "fine_tuning_job_id": "ftjob-abc123",
      "step_number": 2000,
    },
    {
      "object": "fine_tuning.job.checkpoint",
      "id": "ftckpt_enQCFmOTGj3syEpYVhBRLTSy",
      "created_at": 1721764800,
      "fine_tuned_model_checkpoint": "ft:gpt-4o-mini-2024-07-18:my-org:custom-suf
      "metrics": {
        "full_valid_loss": 0.167,
        "full_valid_mean_token_accuracy": 0.781
      },
      "fine_tuning_job_id": "ftjob-abc123",
      "step_number": 1000,
    },
  ],
  "first_id": "ftckpt_zc4Q7MP6XxulcVzj4MZdwsAB",
  "last_id": "ftckpt_enQCFmOTGj3syEpYVhBRLTSy",
  "has_more": true
}
```


# 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

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

Response

```json
{
  "object": "fine_tuning.job",
  "id": "ftjob-abc123",
  "model": "davinci-002",
  "created_at": 1692661014,
  "finished_at": 1692661190,
  "fine_tuned_model": "ft:davinci-002:my-org:custom_suffix:7q8mpxmy",
  "organization_id": "org-123",
  "result_files": [
      "file-abc123"
  ],
  "status": "succeeded",
  "validation_file": null,
  "training_file": "file-abc123",
  "hyperparameters": {
      "n_epochs": 4,
      "batch_size": 1,
      "learning_rate_multiplier": 1.0
  },
  "trained_tokens": 5768,
  "integrations": [],
  "seed": 0,
  "estimated_finish": 0
}
```

# Cancel fine-tuning

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

Immediately cancel a fine-tune job.

## Path parameters

**`fine_tuning_job_id`** string Required
The ID of the fine-tuning job to cancel.

## Returns

The cancelled **fine-tuning** object.

Example request curl

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

Response


```
{
  "object": "fine_tuning.job",
  "id": "ftjob-abc123",
  "model": "gpt-4o-mini-2024-07-18",
  "created_at": 1721764800,
  "fine_tuned_model": null,
  "organization_id": "org-123",
  "result_files": [],
  "hyperparameters": {
    "n_epochs":  "auto"
  },
  "status": "cancelled",
  "validation_file": "file-abc123",
  "training_file": "file-abc123"
}
```


# 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

```
{
  "messages": [
    { "role": "user", "content": "What is the weather in San Francisco?" },
    {
      "role": "assistant",
      "tool_calls": [
        {
          "id": "call_id",
          "type": "function",
          "function": {
            "name": "get_current_weather",
            "arguments": "{\"location\": \"San Francisco, USA\", \"format\": \"ce
          }
        }
      ]
    }
  ],
  "parallel_tool_calls": false,
  "tools": [
    {
      "type": "function",
      "function": {
        "name": "get_current_weather",
        "description": "Get the current weather",
        "parameters": {
          "type": "object",
          "properties": {
            "location": {
                "type": "string",
                "description": "The city and country, eg. San Francisco, USA"
            },
            "format": { "type": "string", "enum": ["celsius", "fahrenheit"] }
          },
          "required": ["location", "format"]
        }
      }
    }
  ]
}
```

# Training format for 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

```
{
  "prompt": "What is the answer to 2+2",
  "completion": "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

```
{
  "object": "fine_tuning.job",
  "id": "ftjob-abc123",
  "model": "davinci-002",
  "created_at": 1692661014,
  "finished_at": 1692661190,
  "fine_tuned_model": "ft:davinci-002:my-org:custom_suffix:7q8mpxmy",
  "organization_id": "org-123",
  "result_files": [
      "file-abc123"
  ],
```

```
  "status": "succeeded",
  "validation_file": null,
  "training_file": "file-abc123",
  "hyperparameters": {
      "n_epochs": 4,
      "batch_size": 1,
      "learning_rate_multiplier": 1.0
  },
  "trained_tokens": 5768,
  "integrations": [],
  "seed": 0,
  "estimated_finish": 0
}
```

## 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

```
{
  "object": "fine_tuning.job.event",
  "id": "ftevent-abc123"
  "created_at": 1677610602,
  "level": "info",
  "message": "Created fine-tuning job"
}
```

## 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


```
{
  "object": "fine_tuning.job.checkpoint",
  "id": "ftckpt_qtZ5Gyk4BLq1SfLFWp3RtO3P",
  "created_at": 1712211699,
  "fine_tuned_model_checkpoint": "ft:gpt-4o-mini-2024-07-18:my-org:custom_suffix:
  "fine_tuning_job_id": "ftjob-fpbNQ3H1GrMehXRf8cO97xTN",
  "metrics": {
    "step": 88,
    "train_loss": 0.478,
    "train_mean_token_accuracy": 0.924,
    "valid_loss": 10.112,
    "valid_mean_token_accuracy": 0.145,
    "full_valid_loss": 0.567,
    "full_valid_mean_token_accuracy": 0.944
  },
  "step_number": 88
}
```


# 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

```curl
curl https://api.openai.com/v1/batches \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "input_file_id": "file-abc123",
    "endpoint": "/v1/chat/completions",

```

```
    "completion_window": "24h"
```

Response


```
{
  "id": "batch_abc123",
  "object": "batch",
  "endpoint": "/v1/chat/completions",
  "errors": null,
  "input_file_id": "file-abc123",
  "completion_window": "24h",
  "status": "validating",
  "output_file_id": null,
  "error_file_id": null,
  "created_at": 1711471533,
  "in_progress_at": null,
  "expires_at": null,
  "finalizing_at": null,
  "completed_at": null,
  "failed_at": null,
  "expired_at": null,
  "cancelling_at": null,
  "cancelled_at": null,
  "request_counts": {
    "total": 0,
    "completed": 0,
    "failed": 0
  },
  "metadata": {
    "customer_id": "user_123456789",
    "batch_description": "Nightly eval job",
  }
}
```


# Retrieve batch

`GET https://api.openai.com/v1/batches/{batch_id}`

Retrieves a batch.

## Path parameters

**batch_id** string Required
The ID of the batch to retrieve.

## Returns

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

Example request curl

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

Response


```
{
  "id": "batch_abc123",
  "object": "batch",
  "endpoint": "/v1/completions",
  "errors": null,
  "input_file_id": "file-abc123",
  "completion_window": "24h",
  "status": "completed",
  "output_file_id": "file-cvaTdG",
  "error_file_id": "file-HOWS94",
  "created_at": 1711471533,
  "in_progress_at": 1711471538,
  "expires_at": 1711557933,
  "finalizing_at": 1711493133,
  "completed_at": 1711493163,
  "failed_at": null,
  "expired_at": null,
  "cancelling_at": null,
  "cancelled_at": null,
  "request_counts": {
    "total": 100,
    "completed": 95,
    "failed": 5
  },
  "metadata": {
    "customer_id": "user_123456789",
    "batch_description": "Nightly eval job",
  }
}
```


# Cancel batch

`POST https://api.openai.com/v1/batches/{batch_id}/cancel`

Cancels an in-progress batch. The batch will be in status `cancelling` for up to 10 minutes, before changing to `cancelled`, where it will have partial results (if any) available in the output file.

## Path parameters

**batch_id** string Required
The ID of the batch to cancel.

## Returns

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

Example request curl

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

Response

```response
{
  "id": "batch_abc123",
  "object": "batch",
  "endpoint": "/v1/chat/completions",
  "errors": null,
  "input_file_id": "file-abc123",
  "completion_window": "24h",
  "status": "cancelling",
  "output_file_id": null,
  "error_file_id": null,
  "created_at": 1711471533,
  "in_progress_at": 1711471538,
  "expires_at": 1711557933,
  "finalizing_at": null,
  "completed_at": null,
  "failed_at": null,
  "expired_at": null,
  "cancelling_at": 1711475133,
  "cancelled_at": null,
  "request_counts": {
    "total": 100,
    "completed": 23,
    "failed": 1
  },
```

```
  "metadata": {
    "customer_id": "user_123456789",
    "batch_description": "Nightly eval job",
  }
}
```

# List batch

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

List your organization's batches.

## Query parameters

**after** string Optional
A cursor for use in pagination. `after` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, ending with obj_foo, your subsequent call can include after=obj_foo in order to fetch the next page of the list.

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

## Returns

A list of paginated **Batch** objects.

Example request curl

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

Response

```
{
  "object": "list",
  "data": [
    {
      "id": "batch_abc123",
      "object": "batch",
      "endpoint": "/v1/chat/completions",
      "errors": null,
      "input_file_id": "file-abc123",
```

```
      "completion_window": "24h",
      "status": "completed",
      "output_file_id": "file-cvaTdG",
      "error_file_id": "file-HOWS94",
      "created_at": 1711471533,
      "in_progress_at": 1711471538,
      "expires_at": 1711557933,
      "finalizing_at": 1711493133,
      "completed_at": 1711493163,
      "failed_at": null,
      "expired_at": null,
      "cancelling_at": null,
      "cancelled_at": null,
      "request_counts": {
        "total": 100,
        "completed": 95,
        "failed": 5
      },
      "metadata": {
        "customer_id": "user_123456789",
        "batch_description": "Nightly job",
      }
    },
    { ... },
  ],
  "first_id": "batch_abc123",
  "last_id": "batch_abc456",
  "has_more": true
}
```


# The batch object

**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


```
{
  "id": "batch_abc123",
  "object": "batch",
  "endpoint": "/v1/completions",
  "errors": null,
  "input_file_id": "file-abc123",
  "completion_window": "24h",
  "status": "completed",
  "output_file_id": "file-cvaTdG",
  "error_file_id": "file-HOWS94",
  "created_at": 1711471533,
  "in_progress_at": 1711471538,
  "expires_at": 1711557933,
  "finalizing_at": 1711493133,
  "completed_at": 1711493163,
  "failed_at": null,
  "expired_at": null,
  "cancelling_at": null,
  "cancelled_at": null,
  "request_counts": {
    "total": 100,
    "completed": 95,
    "failed": 5
  },
  "metadata": {
    "customer_id": "user_123456789",
    "batch_description": "Nightly eval job",
  }
}
```


# The request input object

The per-line object of the batch input file

**`custom_id`** string
A developer-provided per-request id that will be used to match outputs to inputs. Must be unique for each request in a batch.

**`method`** string
The HTTP method to be used for the request. Currently only `POST` is supported.

**`url`** string
The OpenAI API relative URL to be used for the request. Currently `/v1/chat/completions`, `/v1/embeddings`, and `/v1/completions` are supported.

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

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

Response

```response
{
  "id": "file-abc123",
  "object": "file",
  "bytes": 120000,
  "created_at": 1677610602,
  "filename": "mydata.jsonl",
```

```
  "purpose": "fine-tune",
}
```

# List files

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

Returns a list of files 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

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

Response

```
{
  "data": [
    {
      "id": "file-abc123",
      "object": "file",
      "bytes": 175,
      "created_at": 1613677385,
      "filename": "salesOverview.pdf",
      "purpose": "assistants",
    },
    {
      "id": "file-abc123",
      "object": "file",
      "bytes": 140,
      "created_at": 1613779121,
      "filename": "puppy.jsonl",
      "purpose": "fine-tune",
    }
  ],
```

```
  "object": "list"
}
```

# Retrieve file

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

Returns information about a specific file.

## Path parameters

**`file_id`** string Required
The ID of the file to use for this request.

## Returns

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

Example request curl

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

Response

```
{
  "id": "file-abc123",
  "object": "file",
  "bytes": 120000,
  "created_at": 1677610602,
  "filename": "mydata.jsonl",
  "purpose": "fine-tune",
}
```

# Delete file

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

Delete a file.

## Path parameters

**file_id** string Required
The ID of the file to use for this request.

## Returns

Deletion status.

Example request curl

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

Response

```
{
  "id": "file-abc123",
  "object": "file",
  "deleted": true
}
```

# Retrieve file content

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

Returns the contents of the specified file.

## Path parameters

**file_id** string Required
The ID of the file to use for this request.

## Returns

The file content.

Example request curl

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

## The file object

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

**`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

```
{
  "id": "file-abc123",
  "object": "file",
  "bytes": 120000,
  "created_at": 1677610602,
  "filename": "salesOverview.pdf",

```

```
    "purpose": "assistants",
}
```

# Uploads

Allows you to upload large files in multiple parts.

---

# Create upload

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

Creates an intermediate **Upload** object that you can add **Parts** to. Currently, an Upload can accept at most 8 GB in total and expires after an hour after you create it.

Once you complete the Upload, we will create a **File** object that contains all the parts you uploaded. This File is usable in the rest of our platform as a regular File object.

For certain `purpose` 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

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

Response

```
{
  "id": "upload_abc123",
  "object": "upload",
  "bytes": 2147483648,
  "created_at": 1719184911,
  "filename": "training_examples.jsonl",
  "purpose": "fine-tune",
  "status": "pending",
  "expires_at": 1719127296
}
```

# Add upload part

`POST https://api.openai.com/v1/uploads/{upload_id}/parts`

Adds a **Part** to an **Upload** object. A Part represents a chunk of bytes from the file you are trying to upload.

Each Part can be at most 64 MB, and you can add Parts until you hit the Upload maximum of 8 GB.

It is possible to add multiple Parts in parallel. You can decide the intended order of the Parts when you **complete the Upload**.

## Path parameters

**`upload_id`** string Required
The ID of the Upload.

### Request body

**data** file Required
The chunk of bytes for this Part.

### Returns

The upload **Part** object.

Example request curl

```
curl https://api.openai.com/v1/uploads/upload_abc123/parts
  -F data="aHROcHM6Ly9hcGkub3BlbmFpLmNvbS92MS91cGxvYWRz..."
```

Response

```
{
  "id": "part_def456",
  "object": "upload.part",
  "created_at": 1719185911,
  "upload_id": "upload_abc123"
}
```

## Complete upload

`POST https://api.openai.com/v1/uploads/{upload_id}/complete`

Completes the **Upload**.

Within the returned Upload object, there is a nested **File** object that is ready to use in the rest of the platform.

You can specify the order of the Parts by passing in an ordered list of the Part IDs.

The number of bytes uploaded upon completion must match the number of bytes initially specified when creating the Upload object. No Parts may be added after an Upload is completed.

### Path parameters

**upload_id** string Required

The ID of the Upload.

## Request body

**part_ids** array Required
The ordered list of Part IDs.

**md5** string Optional
The optional md5 checksum for the file contents to verify if the bytes uploaded matches what you expect.

## Returns

The **Upload** object with status `completed` with an additional `file` property containing the created usable File object.

Example request curl

```
curl https://api.openai.com/v1/uploads/upload_abc123/complete
  -d '{
    "part_ids": ["part_def456", "part_ghi789"]
  }'
```

Response

```
{
  "id": "upload_abc123",
  "object": "upload",
  "bytes": 2147483648,
  "created_at": 1719184911,
  "filename": "training_examples.jsonl",
  "purpose": "fine-tune",
  "status": "completed",
  "expires_at": 1719127296,
  "file": {
    "id": "file-xyz321",
    "object": "file",
    "bytes": 2147483648,
    "created_at": 1719186911,
    "filename": "training_examples.jsonl",
    "purpose": "fine-tune",
  }
}
```

# Cancel upload

`POST https://api.openai.com/v1/uploads/{upload_id}/cancel`

Cancels the Upload. No Parts may be added after an Upload is cancelled.

## Path parameters

**upload_id** string Required
The ID of the Upload.

## Returns

The **Upload** object with status `cancelled`.

Example request curl

```
curl https://api.openai.com/v1/uploads/upload_abc123/cancel
```

Response

```
{
  "id": "upload_abc123",
  "object": "upload",
  "bytes": 2147483648,
  "created_at": 1719184911,
  "filename": "training_examples.jsonl",
  "purpose": "fine-tune",
  "status": "cancelled",
  "expires_at": 1719127296
}
```

# The upload object

The Upload object can accept byte chunks in the form of Parts.

**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

```
{
  "id": "upload_abc123",
  "object": "upload",
  "bytes": 2147483648,
  "created_at": 1719184911,
  "filename": "training_examples.jsonl",
  "purpose": "fine-tune",
  "status": "completed",
  "expires_at": 1719127296,
  "file": {
    "id": "file-xyz321",
    "object": "file",
    "bytes": 2147483648,
    "created_at": 1719186911,
    "filename": "training_examples.jsonl",
    "purpose": "fine-tune",
  }
}
```

# The upload part object

The upload Part represents a chunk of bytes we can add to an Upload object.

**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

```
{
    "id": "part_def456",
    "object": "upload.part",
    "created_at": 1719186911,
    "upload_id": "upload_abc123"
}
```

# 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

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

Response

```
{
  "created": 1589478378,
  "data": [
    {
      "url": "https://..."
    },
    {
      "url": "https://..."
    }
  ]
}
```

# 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

```curl
curl https://api.openai.com/v1/images/edits \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -F image="@otter.png" \
  -F mask="@mask.png" \
  -F prompt="A cute baby sea otter wearing a beret" \
  -F n=2 \
  -F size="1024x1024"
```

Response

```response
{
  "created": 1589478378,
  "data": [
    {
      "url": "https://..."
    },
    {
      "url": "https://..."
    }
  ]
}
```

# 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

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

Response

```response
{
  "created": 1589478378,
  "data": [
    {
      "url": "https://..."
    },
    {
      "url": "https://..."
    }
  ]
}
```

# 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

```
{
  "url": "...",
  "revised_prompt": "..."
}
```

# 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

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

Response

```json
{
  "object": "list",
  "data": [
    {
      "id": "model-id-0",
      "object": "model",
      "created": 1686935002,
      "owned_by": "organization-owner"
    },
    {
      "id": "model-id-1",
      "object": "model",
      "created": 1686935002,
      "owned_by": "organization-owner",
    },
    {
      "id": "model-id-2",
      "object": "model",
      "created": 1686935002,
      "owned_by": "openai"
    },
  ],
  "object": "list"
}
```

# Retrieve model

`GET https://api.openai.com/v1/models/{model}`

Retrieves a model instance, providing basic information about the model such as the owner and permissioning.

## Path parameters

**model** string Required
The ID of the model to use for this request

## Returns

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

Example request gpt-3.5-turbo-instruct curl

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

Response gpt-3.5-turbo-instruct

```
{
  "id": "gpt-3.5-turbo-instruct",
  "object": "model",
  "created": 1686935002,
  "owned_by": "openai"
}
```

# Delete a fine-tuned model

`DELETE https://api.openai.com/v1/models/{model}`

Delete a fine-tuned model. You must have the Owner role in your organization to delete a model.

## Path parameters

**model** string Required
The model to delete

## Returns

Deletion status.

Example request curl

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

Response

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

## The model object

Describes an OpenAI model offering that can be used with the API.

**`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

```
{
  "id": "davinci",
  "object": "model",
  "created": 1686935002,
  "owned_by": "openai"
}
```

## Moderations

Given text and/or image inputs, classifies if those inputs are potentially harmful across several categories. Related guide: **Moderations**

## Create moderation

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

Classifies if text and/or image inputs are potentially harmful. Learn more in the **moderation guide**.

## Request body

**input** string or array Required
Input (or inputs) to classify. Can be a single string, an array of strings, or an array of multi-modal input objects similar to other models.

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

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

Response

```
{
  "id": "modr-AB8CjOTu2jiq12hp1AQPfeqFWaORR",
  "model": "text-moderation-007",
  "results": [
    {
      "flagged": true,
      "categories": {
        "sexual": false,
        "hate": false,
        "harassment": true,
        "self-harm": false,
        "sexual/minors": false,
        "hate/threatening": false,
        "violence/graphic": false,
```

```
        "self-harm/intent": false,
        "self-harm/instructions": false,
        "harassment/threatening": true,
        "violence": true
      },
      "category_scores": {
        "sexual": 0.000011726012417057063,
        "hate": 0.22706663608551025,
        "harassment": 0.5215635299682617,
        "self-harm": 2.227119921371923e-6,
        "sexual/minors": 7.107352217872176e-8,
        "hate/threatening": 0.023547329008579254,
        "violence/graphic": 0.00003391829886822961,
        "self-harm/intent": 1.646940972932498e-6,
        "self-harm/instructions": 1.1198755256458526e-9,
        "harassment/threatening": 0.5694745779037476,
        "violence": 0.9971134662628174
      }
    }
  ]
}
```


## The moderation object

Represents if a given text input is potentially harmful.

**`id`** string
The unique identifier for the moderation request.

**`model`** string
The model used to generate the moderation results.

**`results`** array
A list of moderation objects.

Show properties

The moderation object


```
{
  "id": "modr-0d9740456c391e43c445bf0f010940c7",
  "model": "omni-moderation-latest",
  "results": [
    {
      "flagged": true,
      "categories": {
        "harassment": true,
        "harassment/threatening": true,
```

```
        "sexual": false,
        "hate": false,
        "hate/threatening": false,
        "illicit": false,
        "illicit/violent": false,
        "self-harm/intent": false,
        "self-harm/instructions": false,
        "self-harm": false,
        "sexual/minors": false,
        "violence": true,
        "violence/graphic": true
      },
      "category_scores": {
        "harassment": 0.8189693396524255,
        "harassment/threatening": 0.804985420696006,
        "sexual": 1.573112165348997e-6,
        "hate": 0.007562942636942845,
        "hate/threatening": 0.004208854591835476,
        "illicit": 0.030535955153511665,
        "illicit/violent": 0.008925306722380033,
        "self-harm/intent": 0.00023023930975076432,
        "self-harm/instructions": 0.0002293869201073356,
        "self-harm": 0.012598046106750154,
        "sexual/minors": 2.212566909570261e-8,
        "violence": 0.9999992735124786,
        "violence/graphic": 0.843064871157054
      },
      "category_applied_input_types": {
        "harassment": [
          "text"
        ],
        "harassment/threatening": [
          "text"
        ],
        "sexual": [
          "text",
          "image"
        ],
        "hate": [
          "text"
        ],
        "hate/threatening": [
          "text"
        ],
        "illicit": [
          "text"
        ],
        "illicit/violent": [
          "text"
        ],
        "self-harm/intent": [
          "text",
```

```
          "image"
        ],
        "self-harm/instructions": [
          "text",
          "image"
        ],
        "self-harm": [
          "text",
          "image"
        ],
        "sexual/minors": [
          "text"
        ],
        "violence": [
          "text",
          "image"
        ],
        "violence/graphic": [
          "text",
          "image"
        ]
      }
    }
  ]
}
```

## 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 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

An **assistant** object.

Code Interpreter | Files

Example request curl

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

Response

```
{
  "id": "asst_abc123",
  "object": "assistant",
  "created_at": 1698984975,
  "name": "Math Tutor",
  "description": null,
  "model": "gpt-4o",
  "instructions": "You are a personal math tutor. When asked a question, write ar
  "tools": [
    {
      "type": "code_interpreter"
    }
  ],
  "metadata": {},
  "top_p": 1.0,
  "temperature": 1.0,
  "response_format": "auto"
}
```

# List assistants Beta

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

Returns a list of assistants.

## Query parameters

**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

```curl
curl "https://api.openai.com/v1/assistants?order=desc&limit=20" \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "OpenAI-Beta: assistants=v2"
```

Response

```
{
  "object": "list",
  "data": [
    {
```

```
      "id": "asst_abc123",
      "object": "assistant",
      "created_at": 1698982736,
      "name": "Coding Tutor",
      "description": null,
      "model": "gpt-4o",
      "instructions": "You are a helpful assistant designed to make me better at
      "tools": [],
      "tool_resources": {},
      "metadata": {},
      "top_p": 1.0,
      "temperature": 1.0,
      "response_format": "auto"
    },
    {
      "id": "asst_abc456",
      "object": "assistant",
      "created_at": 1698982718,
      "name": "My Assistant",
      "description": null,
      "model": "gpt-4o",
      "instructions": "You are a helpful assistant designed to make me better at
      "tools": [],
      "tool_resources": {},
      "metadata": {},
      "top_p": 1.0,
      "temperature": 1.0,
      "response_format": "auto"
    },
    {
      "id": "asst_abc789",
      "object": "assistant",
      "created_at": 1698982643,
      "name": null,
      "description": null,
      "model": "gpt-4o",
      "instructions": null,
      "tools": [],
      "tool_resources": {},
      "metadata": {},
      "top_p": 1.0,
      "temperature": 1.0,
      "response_format": "auto"
    }
  ],
  "first_id": "asst_abc123",
  "last_id": "asst_abc789",
  "has_more": false
}
```

# Retrieve assistant Beta

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

Retrieves an assistant.

## Path parameters

**assistant_id** string Required
The ID of the assistant to retrieve.

## Returns

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

Example request curl

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

Response

```response
{
  "id": "asst_abc123",
  "object": "assistant",
  "created_at": 1699009709,
  "name": "HR Helper",
  "description": null,
  "model": "gpt-4o",
  "instructions": "You are an HR bot, and you have access to files to answer empl
  "tools": [
    {
      "type": "file_search"
    }
  ],
  "metadata": {},
  "top_p": 1.0,
  "temperature": 1.0,
  "response_format": "auto"
}
```

# Modify assistant Beta

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

Modifies an assistant.

## Path parameters

**`assistant_id`** string Required
The ID of the assistant to modify.

## Request body

**`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

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

Response

```
{
  "id": "asst_123",
  "object": "assistant",
  "created_at": 1699009709,
  "name": "HR Helper",
  "description": null,
  "model": "gpt-4o",
  "instructions": "You are an HR bot, and you have access to files to answer empl
  "tools": [
    {
      "type": "file_search"
    }
  ],
  "tool_resources": {
    "file_search": {
      "vector_store_ids": []
    }
  },
  "metadata": {},
  "top_p": 1.0,
  "temperature": 1.0,
  "response_format": "auto"
}
```


## Delete assistant Beta

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

Delete an assistant.

### Path parameters

**assistant_id** string Required
The ID of the assistant to delete.

### Returns

Deletion status

Example request curl

```
curl https://api.openai.com/v1/assistants/asst_abc123 \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
```

```
  -H "OpenAI-Beta: assistants=v2" \
  -X DELETE
```

Response

```
{
  "id": "asst_abc123",
  "object": "assistant.deleted",
  "deleted": true
}
```

## The assistant object Beta

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

**`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

```
{
  "id": "asst_abc123",
  "object": "assistant",
  "created_at": 1698984975,
  "name": "Math Tutor",
  "description": null,
  "model": "gpt-4o",
  "instructions": "You are a personal math tutor. When asked a question, write ar
  "tools": [
```

```
    {
      "type": "code_interpreter"
    }
  ],
  "metadata": {},
  "top_p": 1.0,
  "temperature": 1.0,
  "response_format": "auto"
}
```

## Threads Beta

Create threads that assistants can interact with.

Related guide: **Assistants**

## Create thread Beta

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

Create a thread.

### Request body

**messages** array Optional
A list of **messages** to start the thread with.

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

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

Response

```
{
  "id": "thread_abc123",
  "object": "thread",
  "created_at": 1699012949,
  "metadata": {},
  "tool_resources": {}
}
```

# Retrieve thread Beta

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

Retrieves a thread.

## Path parameters

**thread_id** string Required
The ID of the thread to retrieve.

## Returns

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

Example request curl

```
curl https://api.openai.com/v1/threads/thread_abc123 \
  -H "Content-Type: application/json" \
```

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

Response

```
{
  "id": "thread_abc123",
  "object": "thread",
  "created_at": 1699014083,
  "metadata": {},
  "tool_resources": {
    "code_interpreter": {
      "file_ids": []
    }
  }
}
```

# Modify thread Beta

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

Modifies a thread.

## Path parameters

**`thread_id`** string Required
The ID of the thread to modify. Only the `metadata` can be modified.

## Request body

**`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

```curl
curl https://api.openai.com/v1/threads/thread_abc123 \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "OpenAI-Beta: assistants=v2" \
  -d '{
      "metadata": {
        "modified": "true",
        "user": "abc123"
      }
    }'
```

Response

```response
{
  "id": "thread_abc123",
  "object": "thread",
  "created_at": 1699014083,
  "metadata": {
    "modified": "true",
    "user": "abc123"
  },
  "tool_resources": {}
}
```

# Delete thread Beta

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

Delete a thread.

## Path parameters

**thread_id** string Required
The ID of the thread to delete.

## Returns

Deletion status

Example request curl

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

Response

```response
{
  "id": "thread_abc123",
  "object": "thread.deleted",
  "deleted": true
}
```

## 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

```
{
  "id": "thread_abc123",
  "object": "thread",
  "created_at": 1698107661,
  "metadata": {}
}
```

## Messages Beta

Create messages within threads

Related guide: **Assistants**

## Create message Beta

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

Create a message.

### Path parameters

**`thread_id`** string Required
The ID of the **thread** to create a message for.

### Request body

**`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

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

Response

```
{
  "id": "msg_abc123",
  "object": "thread.message",
  "created_at": 1713226573,
  "assistant_id": null,
  "thread_id": "thread_abc123",
  "run_id": null,
  "role": "user",
  "content": [
    {
      "type": "text",
      "text": {
        "value": "How does AI work? Explain it in simple terms.",
        "annotations": []
      }
    }
  ],
  "attachments": [],
  "metadata": {}
}
```

# List messages Beta

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

Returns a list of messages for a given thread.

## Path parameters

**`thread_id`** string Required
The ID of the **thread** the messages belong to.

## Query parameters

**`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

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

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

Response


```
{
  "object": "list",
  "data": [
    {
      "id": "msg_abc123",
      "object": "thread.message",
      "created_at": 1699016383,
      "assistant_id": null,
      "thread_id": "thread_abc123",
      "run_id": null,
      "role": "user",
      "content": [
        {
          "type": "text",
          "text": {
            "value": "How does AI work? Explain it in simple terms.",
            "annotations": []
          }
        }
      ],
      "attachments": [],
      "metadata": {}
    },
    {
      "id": "msg_abc456",
      "object": "thread.message",
      "created_at": 1699016383,
      "assistant_id": null,
      "thread_id": "thread_abc123",
      "run_id": null,
      "role": "user",
      "content": [
        {
          "type": "text",
          "text": {
            "value": "Hello, what is AI?",
            "annotations": []
          }
        }
      ],
      "attachments": [],
      "metadata": {}
    }
  ],
  "first_id": "msg_abc123",
  "last_id": "msg_abc456",
```

```
  "has_more": false
}
```

## Retrieve message Beta

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

Retrieve a message.

### Path parameters

**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

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

Response

```
{
  "id": "msg_abc123",
  "object": "thread.message",
  "created_at": 1699017614,
  "assistant_id": null,
  "thread_id": "thread_abc123",
  "run_id": null,
  "role": "user",
  "content": [
    {
      "type": "text",
      "text": {
        "value": "How does AI work? Explain it in simple terms.",
```

```
        "annotations": []
      }
    }
  ],
  "attachments": [],
  "metadata": {}
}
```

# Modify message Beta

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

Modifies a message.

## Path parameters

**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

```
curl https://api.openai.com/v1/threads/thread_abc123/messages/msg_abc123 \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "OpenAI-Beta: assistants=v2" \
  -d '{
      "metadata": {
        "modified": "true",
        "user": "abc123"
```

```
        }
    }'
```

Response


```
{
  "id": "msg_abc123",
  "object": "thread.message",
  "created_at": 1699017614,
  "assistant_id": null,
  "thread_id": "thread_abc123",
  "run_id": null,
  "role": "user",
  "content": [
    {
      "type": "text",
      "text": {
        "value": "How does AI work? Explain it in simple terms.",
        "annotations": []
      }
    }
  ],
  "file_ids": [],
  "metadata": {
    "modified": "true",
    "user": "abc123"
  }
}
```


## Delete message Beta

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

Deletes a message.

### Path parameters

**`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

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

Response

```
{
  "id": "msg_abc123",
  "object": "thread.message.deleted",
  "deleted": true
}
```

## The message object Beta

Represents a message within a **thread**.

**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


```
{
  "id": "msg_abc123",
  "object": "thread.message",
  "created_at": 1698983503,
  "thread_id": "thread_abc123",
  "role": "assistant",
  "content": [
    {
      "type": "text",
      "text": {
        "value": "Hi! How can I help you today?",
        "annotations": []
      }
    }
  ],
  "assistant_id": "asst_abc123",
  "run_id": "run_abc123",
```

```
  "attachments": [],
  "metadata": {}
}
```

## Runs Beta

Represents an execution run on a thread.

Related guide: **Assistants**

## Create run Beta

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

Create a run.

### Path parameters

**`thread_id`** string Required
The ID of the thread to run.

### Query parameters

**`include[]`** array Optional
A list of additional fields to include in the response. Currently the only supported value is `step_details.tool_calls[*].file_search.results[*].content` to fetch the file search result content.

See the **file search tool documentation** for more information.

### Request body

**`assistant_id`** string Required
The ID of the **assistant** to use to execute this run.

**`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 intial 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

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

```
  -d '{
    "assistant_id": "asst_abc123"
  }'
```

Response

```
{
  "id": "run_abc123",
  "object": "thread.run",
  "created_at": 1699063290,
  "assistant_id": "asst_abc123",
  "thread_id": "thread_abc123",
  "status": "queued",
  "started_at": 1699063290,
  "expires_at": null,
  "cancelled_at": null,
  "failed_at": null,
  "completed_at": 1699063291,
  "last_error": null,
  "model": "gpt-4o",
  "instructions": null,
  "incomplete_details": null,
  "tools": [
    {
      "type": "code_interpreter"
    }
  ],
  "metadata": {},
  "usage": null,
  "temperature": 1.0,
  "top_p": 1.0,
  "max_prompt_tokens": 1000,
  "max_completion_tokens": 1000,
  "truncation_strategy": {
    "type": "auto",
    "last_messages": null
  },
  "response_format": "auto",
  "tool_choice": "auto",
  "parallel_tool_calls": true
}
```

## Create thread and run Beta

`POST https://api.openai.com/v1/threads/runs`

Create a thread and run it in one request.

## Request body

**`assistant_id`** string Required
The ID of the **assistant** to use to execute this run.

**`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 intial 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

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

Response


```
{
  "id": "run_abc123",
  "object": "thread.run",
  "created_at": 1699076792,
  "assistant_id": "asst_abc123",
  "thread_id": "thread_abc123",
  "status": "queued",
  "started_at": null,
  "expires_at": 1699077392,
  "cancelled_at": null,
  "failed_at": null,
  "completed_at": null,
  "required_action": null,
  "last_error": null,
  "model": "gpt-4o",
  "instructions": "You are a helpful assistant.",
  "tools": [],
  "tool_resources": {},
  "metadata": {},
  "temperature": 1.0,
  "top_p": 1.0,
  "max_completion_tokens": null,
  "max_prompt_tokens": null,
  "truncation_strategy": {
    "type": "auto",
    "last_messages": null
```

```
  },
  "incomplete_details": null,
  "usage": null,
  "response_format": "auto",
  "tool_choice": "auto",
  "parallel_tool_calls": true
}
```

## List runs Beta

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

Returns a list of runs belonging to a thread.

### Path parameters

**`thread_id`** string Required
The ID of the thread the run belongs to.

### Query parameters

**`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

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

Response


```
{
  "object": "list",
  "data": [
    {
      "id": "run_abc123",
      "object": "thread.run",
      "created_at": 1699075072,
      "assistant_id": "asst_abc123",
      "thread_id": "thread_abc123",
      "status": "completed",
      "started_at": 1699075072,
      "expires_at": null,
      "cancelled_at": null,
      "failed_at": null,
      "completed_at": 1699075073,
      "last_error": null,
      "model": "gpt-4o",
      "instructions": null,
      "incomplete_details": null,
      "tools": [
        {
          "type": "code_interpreter"
        }
      ],
      "tool_resources": {
        "code_interpreter": {
          "file_ids": [
            "file-abc123",
            "file-abc456"
          ]
        }
      },
      "metadata": {},
      "usage": {
        "prompt_tokens": 123,
        "completion_tokens": 456,
        "total_tokens": 579
      },
```

```
      "temperature": 1.0,
      "top_p": 1.0,
      "max_prompt_tokens": 1000,
      "max_completion_tokens": 1000,
      "truncation_strategy": {
        "type": "auto",
        "last_messages": null
      },
      "response_format": "auto",
      "tool_choice": "auto",
      "parallel_tool_calls": true
    },
    {
      "id": "run_abc456",
      "object": "thread.run",
      "created_at": 1699063290,
      "assistant_id": "asst_abc123",
      "thread_id": "thread_abc123",
      "status": "completed",
      "started_at": 1699063290,
      "expires_at": null,
      "cancelled_at": null,
      "failed_at": null,
      "completed_at": 1699063291,
      "last_error": null,
      "model": "gpt-4o",
      "instructions": null,
      "incomplete_details": null,
      "tools": [
        {
          "type": "code_interpreter"
        }
      ],
      "tool_resources": {
        "code_interpreter": {
          "file_ids": [
            "file-abc123",
            "file-abc456"
          ]
        }
      },
      "metadata": {},
      "usage": {
        "prompt_tokens": 123,
        "completion_tokens": 456,
        "total_tokens": 579
      },
      "temperature": 1.0,
      "top_p": 1.0,
      "max_prompt_tokens": 1000,
      "max_completion_tokens": 1000,
      "truncation_strategy": {
```

```
        "type": "auto",
        "last_messages": null
      },
      "response_format": "auto",
      "tool_choice": "auto",
      "parallel_tool_calls": true
    }
  ],
  "first_id": "run_abc123",
  "last_id": "run_abc456",
  "has_more": false
}
```

## Retrieve run Beta

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

Retrieves a run.

### Path parameters

**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

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

Response

```
{
  "id": "run_abc123",
  "object": "thread.run",
  "created_at": 1699075072,
```

```
  "assistant_id": "asst_abc123",
  "thread_id": "thread_abc123",
  "status": "completed",
  "started_at": 1699075072,
  "expires_at": null,
  "cancelled_at": null,
  "failed_at": null,
  "completed_at": 1699075073,
  "last_error": null,
  "model": "gpt-4o",
  "instructions": null,
  "incomplete_details": null,
  "tools": [
    {
      "type": "code_interpreter"
    }
  ],
  "metadata": {},
  "usage": {
    "prompt_tokens": 123,
    "completion_tokens": 456,
    "total_tokens": 579
  },
  "temperature": 1.0,
  "top_p": 1.0,
  "max_prompt_tokens": 1000,
  "max_completion_tokens": 1000,
  "truncation_strategy": {
    "type": "auto",
    "last_messages": null
  },
  "response_format": "auto",
  "tool_choice": "auto",
  "parallel_tool_calls": true
}
```


## Modify run Beta

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

Modifies a run.

### Path parameters

**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

```
curl https://api.openai.com/v1/threads/thread_abc123/runs/run_abc123 \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -H "OpenAI-Beta: assistants=v2" \
  -d '{
    "metadata": {
      "user_id": "user_abc123"
    }
  }'
```

Response

```
{
  "id": "run_abc123",
  "object": "thread.run",
  "created_at": 1699075072,
  "assistant_id": "asst_abc123",
  "thread_id": "thread_abc123",
  "status": "completed",
  "started_at": 1699075072,
  "expires_at": null,
  "cancelled_at": null,
  "failed_at": null,
  "completed_at": 1699075073,
  "last_error": null,
  "model": "gpt-4o",
  "instructions": null,
  "incomplete_details": null,
  "tools": [
    {
```

```
      "type": "code_interpreter"
    }
  ],
  "tool_resources": {
    "code_interpreter": {
      "file_ids": [
        "file-abc123",
        "file-abc456"
      ]
    }
  },
  "metadata": {
    "user_id": "user_abc123"
  },
  "usage": {
    "prompt_tokens": 123,
    "completion_tokens": 456,
    "total_tokens": 579
  },
  "temperature": 1.0,
  "top_p": 1.0,
  "max_prompt_tokens": 1000,
  "max_completion_tokens": 1000,
  "truncation_strategy": {
    "type": "auto",
    "last_messages": null
  },
  "response_format": "auto",
  "tool_choice": "auto",
  "parallel_tool_calls": true
}
```


## Submit tool outputs to run Beta

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

When a run has the `status: "requires_action"` and `required_action.type` is `submit_tool_outputs`, this endpoint can be used to submit the outputs from the tool calls once they're all completed. All outputs must be submitted in a single request.

### Path parameters

**`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

```
curl https://api.openai.com/v1/threads/thread_123/runs/run_123/submit_tool_output
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -H "OpenAI-Beta: assistants=v2" \
  -d '{
    "tool_outputs": [
      {
        "tool_call_id": "call_001",
        "output": "70 degrees and sunny."
      }
    ]
  }'
```

Response

```
{
  "id": "run_123",
  "object": "thread.run",
  "created_at": 1699075592,
  "assistant_id": "asst_123",
  "thread_id": "thread_123",
  "status": "queued",
  "started_at": 1699075592,
  "expires_at": 1699076192,
```

```
  "cancelled_at": null,
  "failed_at": null,
  "completed_at": null,
  "last_error": null,
  "model": "gpt-4o",
  "instructions": null,
  "tools": [
    {
      "type": "function",
      "function": {
        "name": "get_current_weather",
        "description": "Get the current weather in a given location",
        "parameters": {
          "type": "object",
          "properties": {
            "location": {
              "type": "string",
              "description": "The city and state, e.g. San Francisco, CA"
            },
            "unit": {
              "type": "string",
              "enum": ["celsius", "fahrenheit"]
            }
          },
          "required": ["location"]
        }
      }
    }
  ],
  "metadata": {},
  "usage": null,
  "temperature": 1.0,
  "top_p": 1.0,
  "max_prompt_tokens": 1000,
  "max_completion_tokens": 1000,
  "truncation_strategy": {
    "type": "auto",
    "last_messages": null
  },
  "response_format": "auto",
  "tool_choice": "auto",
  "parallel_tool_calls": true
}
```


## Cancel a run Beta

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

Cancels a run that is `in_progress`.

## Path parameters

**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

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

Response

```response
{
  "id": "run_abc123",
  "object": "thread.run",
  "created_at": 1699076126,
  "assistant_id": "asst_abc123",
  "thread_id": "thread_abc123",
  "status": "cancelling",
  "started_at": 1699076126,
  "expires_at": 1699076726,
  "cancelled_at": null,
  "failed_at": null,
  "completed_at": null,
  "last_error": null,
  "model": "gpt-4o",
  "instructions": "You summarize books.",
  "tools": [
    {
      "type": "file_search"
    }
  ],
  "tool_resources": {
    "file_search": {
      "vector_store_ids": ["vs_123"]
    }
  },
```

```
  "metadata": {},
  "usage": null,
  "temperature": 1.0,
  "top_p": 1.0,
  "response_format": "auto",
  "tool_choice": "auto",
  "parallel_tool_calls": true
}
```

## 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 intial 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

```
{
  "id": "run_abc123",
  "object": "thread.run",
  "created_at": 1698107661,
  "assistant_id": "asst_abc123",
  "thread_id": "thread_abc123",
  "status": "completed",
  "started_at": 1699073476,
  "expires_at": null,
```

```
  "cancelled_at": null,
  "failed_at": null,
  "completed_at": 1699073498,
  "last_error": null,
  "model": "gpt-4o",
  "instructions": null,
  "tools": [{"type": "file_search"}, {"type": "code_interpreter"}],
  "metadata": {},
  "incomplete_details": null,
  "usage": {
    "prompt_tokens": 123,
    "completion_tokens": 456,
    "total_tokens": 579
  },
  "temperature": 1.0,
  "top_p": 1.0,
  "max_prompt_tokens": 1000,
  "max_completion_tokens": 1000,
  "truncation_strategy": {
    "type": "auto",
    "last_messages": null
  },
  "response_format": "auto",
  "tool_choice": "auto",
  "parallel_tool_calls": true
}
```


## Run steps Beta

Represents the steps (model and tool calls) taken during the run.

Related guide: **Assistants**

## List run steps Beta

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

Returns a list of run steps belonging to a run.

### Path parameters

**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

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

Response

```
{
  "object": "list",
```

```
  "data": [
    {
      "id": "step_abc123",
      "object": "thread.run.step",
      "created_at": 1699063291,
      "run_id": "run_abc123",
      "assistant_id": "asst_abc123",
      "thread_id": "thread_abc123",
      "type": "message_creation",
      "status": "completed",
      "cancelled_at": null,
      "completed_at": 1699063291,
      "expired_at": null,
      "failed_at": null,
      "last_error": null,
      "step_details": {
        "type": "message_creation",
        "message_creation": {
          "message_id": "msg_abc123"
        }
      },
      "usage": {
        "prompt_tokens": 123,
        "completion_tokens": 456,
        "total_tokens": 579
      }
    }
  ],
  "first_id": "step_abc123",
  "last_id": "step_abc456",
  "has_more": false
}
```


## Retrieve run step Beta

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

Retrieves a run step.

### Path parameters

**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

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

Response

```json
{
  "id": "step_abc123",
  "object": "thread.run.step",
  "created_at": 1699063291,
  "run_id": "run_abc123",
  "assistant_id": "asst_abc123",
  "thread_id": "thread_abc123",
  "type": "message_creation",
  "status": "completed",
  "cancelled_at": null,
  "completed_at": 1699063291,
  "expired_at": null,
  "failed_at": null,
  "last_error": null,
  "step_details": {
    "type": "message_creation",
    "message_creation": {
      "message_id": "msg_abc123"
    }
  },
  "usage": {
    "prompt_tokens": 123,
```

```
    "completion_tokens": 456,
    "total_tokens": 579
  }
}
```

## The run step object Beta

Represents a step in execution of a run.

**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

```
{
  "id": "step_abc123",
  "object": "thread.run.step",
  "created_at": 1699063291,
  "run_id": "run_abc123",
  "assistant_id": "asst_abc123",
  "thread_id": "thread_abc123",
  "type": "message_creation",
  "status": "completed",
  "cancelled_at": null,
  "completed_at": 1699063291,
  "expired_at": null,
  "failed_at": null,
  "last_error": null,
  "step_details": {
    "type": "message_creation",
    "message_creation": {
      "message_id": "msg_abc123"
    }
  },
  "usage": {
    "prompt_tokens": 123,
    "completion_tokens": 456,
    "total_tokens": 579
```

```
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

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

Response


```
{
  "id": "vs_abc123",
  "object": "vector_store",
  "created_at": 1699061776,
  "name": "Support FAQ",
  "bytes": 139920,
  "file_counts": {
    "in_progress": 0,
    "completed": 3,
    "failed": 0,
    "cancelled": 0,
    "total": 3
  }
}
```


# List vector stores 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

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

Response

```response
{
  "object": "list",
  "data": [
    {
      "id": "vs_abc123",
      "object": "vector_store",
      "created_at": 1699061776,
      "name": "Support FAQ",
      "bytes": 139920,
      "file_counts": {
        "in_progress": 0,
        "completed": 3,
        "failed": 0,
        "cancelled": 0,
        "total": 3
      }
    },
    {
      "id": "vs_abc456",
      "object": "vector_store",
      "created_at": 1699061776,
      "name": "Support FAQ v2",
      "bytes": 139920,
      "file_counts": {
        "in_progress": 0,
```

```
        "completed": 3,
        "failed": 0,
        "cancelled": 0,
        "total": 3
      }
    }
  ],
  "first_id": "vs_abc123",
  "last_id": "vs_abc456",
  "has_more": false
}
```

## Retrieve vector store 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

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

Response

```
{
  "id": "vs_abc123",
  "object": "vector_store",
  "created_at": 1699061776
}
```

# Modify vector store 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

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

Response

```
{
  "id": "vs_abc123",
  "object": "vector_store",
  "created_at": 1699061776,
  "name": "Support FAQ",
  "bytes": 139920,
  "file_counts": {
    "in_progress": 0,
    "completed": 3,
    "failed": 0,
    "cancelled": 0,
    "total": 3
  }
}
```

## Delete vector store 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

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

Response

```
{
  id: "vs_abc123",
  object: "vector_store.deleted",
```

```
  deleted: true
}
```

## 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


```
{
  "id": "vs_123",
  "object": "vector_store",
  "created_at": 1698107661,
  "usage_bytes": 123456,
  "last_active_at": 1698107661,
  "name": "my_vector_store",
  "status": "completed",
  "file_counts": {
    "in_progress": 0,
    "completed": 100,
    "cancelled": 0,
    "failed": 0,
    "total": 100
  },
  "metadata": {},
  "last_used_at": 1698107661
}
```


## 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

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

Response

```
{
  "id": "file-abc123",
  "object": "vector_store.file",
  "created_at": 1699061776,
  "usage_bytes": 1234,
  "vector_store_id": "vs_abcd",
  "status": "completed",
  "last_error": null
}
```

# List vector store files 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

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

Response

```
{
  "object": "list",
  "data": [
    {
```

```
      "id": "file-abc123",
      "object": "vector_store.file",
      "created_at": 1699061776,
      "vector_store_id": "vs_abc123"
    },
    {
      "id": "file-abc456",
      "object": "vector_store.file",
      "created_at": 1699061776,
      "vector_store_id": "vs_abc123"
    }
  ],
  "first_id": "file-abc123",
  "last_id": "file-abc456",
  "has_more": false
}
```

## Retrieve vector store file 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

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

Response

```
{
  "id": "file-abc123",
  "object": "vector_store.file",
  "created_at": 1699061776,
  "vector_store_id": "vs_abcd",
  "status": "completed",
  "last_error": null
}
```

## 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

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

Response

```
{
  id: "file-abc123",
  object: "vector_store.file.deleted",
```

```
  deleted: true
}
```

## The vector store file object Beta

A list of files attached to a vector store.

**`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

```
{
  "id": "file-abc123",
  "object": "vector_store.file",
  "usage_bytes": 1234,
  "created_at": 1698107661,
  "vector_store_id": "vs_abc123",
  "status": "completed",
  "last_error": null,
```

```
  "chunking_strategy": {
    "type": "static",
    "static": {
      "max_chunk_size_tokens": 800,
      "chunk_overlap_tokens": 400
    }
  }
}
```

## Vector store file batches 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

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

Response

```json
{
  "id": "vsfb_abc123",
  "object": "vector_store.file_batch",
  "created_at": 1699061776,
  "vector_store_id": "vs_abc123",
  "status": "in_progress",
  "file_counts": {
    "in_progress": 1,
    "completed": 1,
    "failed": 0,
    "cancelled": 0,
    "total": 0,
  }
}
```

# Retrieve vector store file batch 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

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

Response

```response
{
  "id": "vsfb_abc123",
  "object": "vector_store.file_batch",
  "created_at": 1699061776,
  "vector_store_id": "vs_abc123",
  "status": "in_progress",
  "file_counts": {
    "in_progress": 1,
    "completed": 1,
    "failed": 0,
    "cancelled": 0,
    "total": 0,
  }
}
```

## Cancel vector store file batch 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

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

Response

```
{
  "id": "vsfb_abc123",
  "object": "vector_store.file_batch",
  "created_at": 1699061776,
  "vector_store_id": "vs_abc123",
  "status": "cancelling",
  "file_counts": {
    "in_progress": 12,
    "completed": 3,
    "failed": 0,
    "cancelled": 0,
    "total": 15,
  }
}
```

# List vector store files in a batch Beta

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

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

Response

```
{
  "object": "list",
  "data": [
    {
      "id": "file-abc123",
      "object": "vector_store.file",
      "created_at": 1699061776,
      "vector_store_id": "vs_abc123"
    },
```

```
    {
      "id": "file-abc456",
      "object": "vector_store.file",
      "created_at": 1699061776,
      "vector_store_id": "vs_abc123"
    }
  ],
  "first_id": "file-abc123",
  "last_id": "file-abc456",
  "has_more": false
}
```

## The vector store files batch object Beta

A batch of files attached to a vector store.

**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

```
{
  "id": "vsfb_123",
  "object": "vector_store.files_batch",
  "created_at": 1698107661,
  "vector_store_id": "vs_abc123",
  "status": "completed",
  "file_counts": {
    "in_progress": 0,
```

```
    "completed": 100,
    "failed": 0,
    "cancelled": 0,
    "total": 100
  }
}
```

## 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

```
{
  "id": "msg_123",
  "object": "thread.message.delta",
  "delta": {
    "content": [
      {
        "index": 0,
        "type": "text",
        "text": { "value": "Hello", "annotations": [] }
      }
    ]


```

```
    }
  }
```

## The run step delta object Beta

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

**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

```
{
  "id": "step_123",
  "object": "thread.run.step.delta",
  "delta": {
    "step_details": {
      "type": "tool_calls",
      "tool_calls": [
        {
          "index": 0,
          "id": "call_123",
          "type": "code_interpreter",
          "code_interpreter": { "input": "", "outputs": [] }
        }
      ]
    }
  }
}
```

## Assistant stream events Beta

Represents an event emitted when streaming a Run.

Each event in a server-sent events stream has an `event` and `data` property:

```
event: thread.created
```

```
data: {"id": "thread_123", "object": "thread", ...}
```

We emit events whenever a new object is created, transitions to a new state, or is being streamed in parts (deltas). For example, we emit `thread.run.created` when a new run is created, `thread.run.completed` when a run completes, and so on. When an Assistant chooses to create a message during a run, we emit a `thread.message.created event`, a `thread.message.in_progress` event, many `thread.message.delta` events, and finally a `thread.message.completed` event.

We may add additional events over time, so we recommend handling unknown events gracefully in your code. See the **Assistants API quickstart** to learn how to integrate the Assistants API with streaming.

---

**`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.

# Administration

Programmatically manage your organization. The Audit Logs endpoint provides a log of all actions taken in the organization for security and monitoring purposes. To access these endpoints please generate an Admin API Key through the **API Platform Organization overview**. Admin API keys cannot be used for non-administration endpoints. For best practices on setting up your organization, please refer to this **guide**

# 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

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

Response 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

```
curl -X POST https://api.openai.com/v1/organization/invites \
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -H "Content-Type: application/json" \
  -d '{
      "email": "user@example.com",
      "role": "owner"
  }'
```

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

```
curl https://api.openai.com/v1/organization/invites/invite-abc \
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -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

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

Response 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

```
{
  "object": "organization.invite",
  "id": "invite-abc",
  "email": "user@example.com",
  "role": "owner",
  "status": "accepted",
  "invited_at": 1711471533,
  "expires_at": 1711471533,
  "accepted_at": 1711471533
}
```

# 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

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

Response json

## Modify user

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

Modifies a user's role in the organization.

### Path parameters

**`user_id`** string Required
The ID of the user.

### Request body

**`role`** string Required
`owner` or `reader`

### Returns

The updated **User** object.

Example request curl

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

Response 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

```
curl https://api.openai.com/v1/organization/users/user_abc \
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -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

```
curl -X DELETE https://api.openai.com/v1/organization/users/user_abc \
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -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

```
{
    "object": "organization.user",
    "id": "user_abc",
    "name": "First Last",
    "email": "user@example.com",
    "role": "owner",
    "added_at": 1711471533
}
```

# Projects

Manage the projects within an orgnanization includes creation, updating, and archiving or projects. The Default project cannot be 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

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

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

Response 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

```
curl https://api.openai.com/v1/organization/projects/proj_abc \
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -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.

## Path parameters

**project_id** string Required
The ID of the project.

## Request body

**name** string Required
The updated name of the project, this name appears in reports.

### Returns

The updated **Project** object.

Example request curl

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

## Archive project

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

Archives a project in the organization. Archived projects cannot be used or updated.

### Path parameters

**`project_id`** string Required
The ID of the project.

### Returns

The archived **Project** object.

Example request curl

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

Response 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

```
{
    "id": "proj_abc",
    "object": "organization.project",
    "name": "Project example",
    "created_at": 1711471533,
    "archived_at": null,
    "status": "active"
}
```

# Project users

Manage users within a project, including adding, updating roles, and removing users. 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

```
curl https://api.openai.com/v1/organization/projects/proj_abc/users?after=user_abc
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -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

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

Response 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

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

Response 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.

### Path parameters

**`project_id`** string Required
The ID of the project.

**`user_id`** string Required
The ID of the user.

### Request body

**`role`** string Required
`owner` or `member`

### Returns

The updated **ProjectUser** object.

Example request curl

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

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

```
curl -X DELETE https://api.openai.com/v1/organization/projects/proj_abc/users/user
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -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

```
{
    "object": "organization.project.user",
    "id": "user_abc",
    "name": "First Last",
    "email": "user@example.com",
    "role": "owner",
    "added_at": 1711471533
}
```

## Project service accounts

Manage service accounts within a project. A service account is a bot user that is not associated with a user. If a user leaves an organization, their keys and membership in projects will no

longer work. Service accounts do not have this limitation. However, service accounts can also be deleted from a project.

## List project service accounts

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

Returns a list of service accounts in the project.

### Path parameters

**`project_id`** string Required
The ID of the project.

### Query parameters

**`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

```
curl https://api.openai.com/v1/organization/projects/proj_abc/service_accounts?aft
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -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

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

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

```
curl https://api.openai.com/v1/organization/projects/proj_abc/service_accounts/svc
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -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

```
curl -X DELETE https://api.openai.com/v1/organization/projects/proj_abc/service_ac
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -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

```
{
    "object": "organization.project.service_account",
    "id": "svc_acct_abc",
    "name": "Service Account",
    "role": "owner",
```

```
    "created_at": 1711471533
}
```

# Project API keys

Manage API keys for a given project. Supports listing and deleting keys for users. This API does not allow issuing keys for users, as users need to authorize themselves to generate keys.

---

# List project API keys

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

Returns a list of API keys in the project.

## Path parameters

**`project_id`** string Required
The ID of the project.

## Query parameters

**`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

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

Response json

# 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

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

Response 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

```
curl -X DELETE https://api.openai.com/v1/organization/projects/proj_abc/api_keys/k
  -H "Authorization: Bearer $OPENAI_ADMIN_KEY" \
  -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


```
{
    "object": "organization.project.api_key",
    "redacted_value": "sk-abc...def",
    "name": "My API Key",
    "created_at": 1711471533,
    "id": "key_abc",
    "owner": {
        "type": "user",
        "user": {
            "object": "organization.project.user",
            "id": "user_abc",
            "name": "First Last",
            "email": "user@example.com",
            "role": "owner",
            "created_at": 1711471533
        }
    }
}
```


## 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

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

Response

```
{
    "object": "list",
    "data": [
        {
```

```
            "id": "audit_log-xxx_yyyymmdd",
            "type": "project.archived",
            "effective_at": 1722461446,
            "actor": {
                "type": "api_key",
                "api_key": {
                    "type": "user",
                    "user": {
                        "id": "user-xxx",
                        "email": "user@example.com"
                    }
                }
            },
            "project.archived": {
                "id": "proj_abc"
            },
        },
        {
            "id": "audit_log-yyy__20240101",
            "type": "api_key.updated",
            "effective_at": 1720804190,
            "actor": {
                "type": "session",
                "session": {
                    "user": {
                        "id": "user-xxx",
                        "email": "user@example.com"
                    },
                    "ip_address": "127.0.0.1",
                    "user_agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) Appl
                }
            },
            "api_key.updated": {
                "id": "key_xxxx",
                "data": {
                    "scopes": ["resource_2.operation_2"]
                }
            },
        }
    ],
    "first_id": "audit_log-xxx__20240101",
    "last_id": "audit_log_yyy__20240101",
    "has_more": true
}
```


## The audit log object

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

**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

```
{
    "id": "req_xxx_20240101",
    "type": "api_key.created",
    "effective_at": 1720804090,
    "actor": {
        "type": "session",
        "session": {
            "user": {
                "id": "user-xxx",
                "email": "user@example.com"
            },
            "ip_address": "127.0.0.1",
            "user_agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/
        }
    },
    "api_key.created": {
        "id": "key_xxxx",
        "data": {
            "scopes": ["resource.operation"]
        }
    }
}
```

# Realtime Beta

Communicate with a GPT-4o class model live, in real time, over WebSocket. Produces both audio and text transcriptions. **Learn more about the Realtime API**.

## Client events

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

## session.update

Send this event to update the session's default configuration. The client may send this event at any time to update the session configuration, and any field may be updated at any time, except for "voice". The server will respond with a `session.updated` event that shows the full effective configuration. Only fields that are present are updated, thus the correct way to clear a field like "instructions" is to pass an empty string.

**`event_id`** string
Optional client-generated ID used to identify this event.

**`type`** string
The event type, must be "session.update".

**`session`** object
Realtime session object configuration.

Show properties

session.update

```
{
    "event_id": "event_123",
    "type": "session.update",
    "session": {
        "modalities": ["text", "audio"],
        "instructions": "Your knowledge cutoff is 2023-10. You are a helpful assi
        "voice": "alloy",
        "input_audio_format": "pcm16",
        "output_audio_format": "pcm16",
        "input_audio_transcription": {
            "model": "whisper-1"
        },
        "turn_detection": {
            "type": "server_vad",
            "threshold": 0.5,
            "prefix_padding_ms": 300,
            "silence_duration_ms": 500
        },
        "tools": [
            {
                "type": "function",
                "name": "get_weather",
                "description": "Get the current weather for a location, tell the
                "parameters": {
                    "type": "object",
                    "properties": {
                        "location": { "type": "string" }
                    },
                    "required": ["location"]
                }
```

```
            }
        ],
        "tool_choice": "auto",
        "temperature": 0.8,
        "max_response_output_tokens": "inf"
    }
}
```

## input_audio_buffer.append

Send this event to append audio bytes to the input audio buffer. The audio buffer is temporary storage you can write to and later commit. In Server VAD mode, the audio buffer is used to detect speech and the server will decide when to commit. When Server VAD is disabled, you must commit the audio buffer manually. The client may choose how much audio to place in each event up to a maximum of 15 MiB, for example streaming smaller chunks from the client may allow the VAD to be more responsive. Unlike made other client events, the server will not send a confirmation response to this event.

**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. This must be in the format specified by the `input_audio_format` field in the session configuration.

input_audio_buffer.append

```
{
    "event_id": "event_456",
    "type": "input_audio_buffer.append",
    "audio": "Base64EncodedAudioData"
}
```

## input_audio_buffer.commit

Send this event to commit the user input audio buffer, which will create a new user message item in the conversation. This event will produce an error if the input audio buffer is empty. When in Server VAD mode, the client does not need to send this event, the server will commit the audio buffer automatically. Committing the input audio buffer will trigger input audio

transcription (if enabled in session configuration), but it will not create a response from the model. The server will respond with an `input_audio_buffer.committed` event.

**event_id** string
Optional client-generated ID used to identify this event.

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

input_audio_buffer.commit

```
{
    "event_id": "event_789",
    "type": "input_audio_buffer.commit"
}
```

## input_audio_buffer.clear

Send this event to clear the audio bytes in the buffer. The server will respond with an `input_audio_buffer.cleared` event.

**event_id** string
Optional client-generated ID used to identify this event.

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

input_audio_buffer.clear

```
{
    "event_id": "event_012",
    "type": "input_audio_buffer.clear"
}
```

## conversation.item.create

Add a new Item to the Conversation's context, including messages, function calls, and function call responses. This event can be used both to populate a "history" of the conversation and to add new items mid-stream, but has the current limitation that it cannot populate assistant

audio messages. If successful, the server will respond with a `conversation.item.created` event, otherwise an `error` event will be sent.

**`event_id`** string
Optional client-generated ID used to identify this event.

**`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. If not set, the new item will be appended to the end of the conversation. If set, it allows an item to be inserted mid-conversation. If the ID cannot be found, an error will be returned and the item will not be added.

**`item`** object
The item to add to the conversation.

Show properties

conversation.item.create

```
{
    "event_id": "event_345",
    "type": "conversation.item.create",
    "previous_item_id": null,
    "item": {
        "id": "msg_001",
        "type": "message",
        "role": "user",
        "content": [
            {
                "type": "input_text",
                "text": "Hello, how are you?"
            }
        ]
    }
}
```

## conversation.item.truncate

Send this event to truncate a previous assistant message's audio. The server will produce audio faster than realtime, so this event is useful when the user interrupts to truncate audio that has already been sent to the client but not yet played. This will synchronize the server's understanding of the audio with the client's playback. Truncating audio will delete the server-side text transcript to ensure there is not text in the context that hasn't been heard by the user. If successful, the server will respond with a `conversation.item.truncated` event.

**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. Only assistant message items can be truncated.

**content_index** integer
The index of the content part to truncate. Set this to 0.

**audio_end_ms** integer
Inclusive duration up to which audio is truncated, in milliseconds. If the audio_end_ms is greater than the actual audio duration, the server will respond with an error.

conversation.item.truncate

```
{
    "event_id": "event_678",
    "type": "conversation.item.truncate",
    "item_id": "msg_002",
    "content_index": 0,
    "audio_end_ms": 1500
}
```

## conversation.item.delete

Send this event when you want to remove any item from the conversation history. The server will respond with a `conversation.item.deleted` event, unless the item does not exist in the conversation history, in which case the server will respond with an error.

**event_id** string
Optional client-generated ID used to identify this event.

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

**item_id** string
The ID of the item to delete.

conversation.item.delete

```
{
    "event_id": "event_901",
    "type": "conversation.item.delete",
    "item_id": "msg_003"
}
```

## response.create

This event instructs the server to create a Response, which means triggering model inference. When in Server VAD mode, the server will create Responses automatically. A Response will include at least one Item, and may have two, in which case the second will be a function call. These Items will be appended to the conversation history. The server will respond with a `response.created` event, events for Items and content created, and finally a `response.done` event to indicate the Response is complete. The `response.create` event includes inference configuration like `instructions`, and `temperature`. These fields will override the Session's configuration for this Response only.

**`event_id`** string
Optional client-generated ID used to identify this event.

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

**`response`** object
The response resource.

Show properties

response.create

```
{
    "event_id": "event_234",
    "type": "response.create",
    "response": {
        "modalities": ["text", "audio"],
        "instructions": "Please assist the user.",
        "voice": "alloy",
        "output_audio_format": "pcm16",
        "tools": [
            {
                "type": "function",
                "name": "calculate_sum",
                "description": "Calculates the sum of two numbers.",
                "parameters": {
```

```
                    "type": "object",
                    "properties": {
                        "a": { "type": "number" },
                        "b": { "type": "number" }
                    },
                    "required": ["a", "b"]
                }
            }
        ],
        "tool_choice": "auto",
        "temperature": 0.7,
        "max_output_tokens": 150
    }
}
```

## response.cancel

Send this event to cancel an in-progress response. The server will respond with a `response.cancelled` event or an error if there is no response to cancel.

**event_id** string
Optional client-generated ID used to identify this event.

**type** string
The event type, must be `response.cancel`.

response.cancel

```
{
    "event_id": "event_567",
    "type": "response.cancel"
}
```

## Server events

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

## error

Returned when an error occurs, which could be a client problem or a server problem. Most errors are recoverable and the session will stay open, we recommend to implementors to monitor and log error messages by default.

**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

```
{
    "event_id": "event_890",
    "type": "error",
    "error": {
        "type": "invalid_request_error",
        "code": "invalid_event",
        "message": "The 'type' field is missing.",
        "param": null,
        "event_id": "event_567"
    }
}
```

## session.created

Returned when a Session is created. Emitted automatically when a new connection is established as the first server event. This event will contain the default Session configuration.

**event_id** string
The unique ID of the server event.

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

**session** object
Realtime session object configuration.

Show properties

session.created

```
{
    "event_id": "event_1234",
    "type": "session.created",
    "session": {
        "id": "sess_001",
        "object": "realtime.session",
        "model": "gpt-4o-realtime-preview-2024-10-01",
        "modalities": ["text", "audio"],
        "instructions": "",
        "voice": "alloy",
        "input_audio_format": "pcm16",
        "output_audio_format": "pcm16",
        "input_audio_transcription": null,
        "turn_detection": {
            "type": "server_vad",
            "threshold": 0.5,
            "prefix_padding_ms": 300,
            "silence_duration_ms": 200
        },
        "tools": [],
        "tool_choice": "auto",
        "temperature": 0.8,
        "max_response_output_tokens": null
    }
}
```

## session.updated

Returned when a session is updated with a `session.update` event, unless there is an error.

**event_id** string
The unique ID of the server event.

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

**session** object
Realtime session object configuration.

Show properties

session.updated

```
{
    "event_id": "event_5678",
    "type": "session.updated",
    "session": {
```

```
        "id": "sess_001",
        "object": "realtime.session",
        "model": "gpt-4o-realtime-preview-2024-10-01",
        "modalities": ["text"],
        "instructions": "New instructions",
        "voice": "alloy",
        "input_audio_format": "pcm16",
        "output_audio_format": "pcm16",
        "input_audio_transcription": {
            "model": "whisper-1"
        },
        "turn_detection": null,
        "tools": [],
        "tool_choice": "none",
        "temperature": 0.7,
        "max_response_output_tokens": 200
    }
}
```

## conversation.created

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

```
{
    "event_id": "event_9101",
    "type": "conversation.created",
    "conversation": {
        "id": "conv_001",
        "object": "realtime.conversation"
    }
}
```

# conversation.item.created

Returned when a conversation item is created. There are several scenarios that produce this event:

> The server is generating a Response, which if successful will produce either one or two Items, which will be of type `message` (role `assistant`) or type `function_call`.
>
> The input audio buffer has been committed, either by the client or the server (in `server_vad` mode). The server will take the content of the input audio buffer and add it to a new user message Item.
>
> The client has sent a `conversation.item.create` event to add a new Item to the Conversation.

---

**`event_id`** string
The unique ID of the server event.

---

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

---

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

---

**`item`** object
The item to add to the conversation.

Show properties

conversation.item.created

```
{
    "event_id": "event_1920",
    "type": "conversation.item.created",
    "previous_item_id": "msg_002",
    "item": {
        "id": "msg_003",
        "object": "realtime.item",
        "type": "message",
        "status": "completed",
        "role": "user",
        "content": [
            {
                "type": "input_audio",
                "transcript": "hello how are you",
                "audio": "base64encodedaudio=="
```

```
            }
        ]
    }
}
```

## conversation.item.input_audio_transcription.completed

This event is the output of audio transcription for user audio written to the user audio buffer. Transcription begins when the input audio buffer is committed by the client or server (in `server_vad` mode). Transcription runs asynchronously with Response creation, so this event may come before or after the Response events. Realtime API models accept audio natively, and thus input transcription is a separate process run on a separate ASR (Automatic Speech Recognition) model, currently always `whisper-1`. Thus the transcript may diverge somewhat from the model's interpretation, and should be treated as a rough guide.

**`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 containing the audio.

**`content_index`** integer
The index of the content part containing the audio.

**`transcript`** string
The transcribed text.

conversation.item.input_audio_transcription.completed

```
{
    "event_id": "event_2122",
    "type": "conversation.item.input_audio_transcription.completed",
    "item_id": "msg_003",
    "content_index": 0,
    "transcript": "Hello, how are you?"
}
```

## conversation.item.input_audio_transcription.failed

Returned when input audio transcription is configured, and a transcription request for a user message failed. These events are separate from other `error` events so that the client can identify the related Item.

**`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

```
{
    "event_id": "event_2324",
    "type": "conversation.item.input_audio_transcription.failed",
    "item_id": "msg_003",
    "content_index": 0,
    "error": {
        "type": "transcription_error",
        "code": "audio_unintelligible",
        "message": "The audio could not be transcribed.",
        "param": null
    }
}
```

## conversation.item.truncated

Returned when an earlier assistant audio message item is truncated by the client with a `conversation.item.truncate` event. This event is used to synchronize the server's understanding of the audio with the client's playback. This action will truncate the audio and remove the server-side text transcript to ensure there is no text in the context that hasn't been heard by the user.

**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

```
{
    "event_id": "event_2526",
    "type": "conversation.item.truncated",
    "item_id": "msg_004",
    "content_index": 0,
    "audio_end_ms": 1500
}
```

## conversation.item.deleted

Returned when an item in the conversation is deleted by the client with a `conversation.item.delete` event. This event is used to synchronize the server's understanding of the conversation history with the client's view.

**event_id** string
The unique ID of the server event.

**type** string
The event type, must be `conversation.item.deleted`.

**item_id** string
The ID of the item that was deleted.

conversation.item.deleted

```
{
    "event_id": "event_2728",
    "type": "conversation.item.deleted",
    "item_id": "msg_005"
}
```

## input_audio_buffer.committed

Returned when an input audio buffer is committed, either by the client or automatically in server VAD mode. The `item_id` property is the ID of the user message item that will be created, thus a `conversation.item.created` event will also be sent to the client.

**event_id** string
The unique ID of the server event.

**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

```
{
    "event_id": "event_1121",
    "type": "input_audio_buffer.committed",
    "previous_item_id": "msg_001",
    "item_id": "msg_002"
}
```

## input_audio_buffer.cleared

Returned when the input audio buffer is cleared by the client with a `input_audio_buffer.clear` event.

**event_id** string
The unique ID of the server event.

**type** string
The event type, must be `input_audio_buffer.cleared`.

input_audio_buffer.cleared

```
{
    "event_id": "event_1314",
    "type": "input_audio_buffer.cleared"
}
```

## input_audio_buffer.speech_started

Sent by the server when in `server_vad` mode to indicate that speech has been detected in the audio buffer. This can happen any time audio is added to the buffer (unless speech is already detected). The client may want to use this event to interrupt audio playback or provide visual feedback to the user. The client should expect to receive a `input_audio_buffer.speech_stopped` event when speech stops. The `item_id` property is the ID of the user message item that will be created when speech stops and will also be included in the `input_audio_buffer.speech_stopped` event (unless the client manually commits the audio buffer during VAD activation).

**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 from the start of all audio written to the buffer during the session when speech was first detected. This will correspond to the beginning of audio sent to the model, and thus includes the `prefix_padding_ms` configured in the Session.

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

input_audio_buffer.speech_started

```
{
    "event_id": "event_1516",
    "type": "input_audio_buffer.speech_started",
    "audio_start_ms": 1000,

```

```
    "item_id": "msg_003"
}
```

## input_audio_buffer.speech_stopped

Returned in `server_vad` mode when the server detects the end of speech in the audio buffer. The server will also send an `conversation.item.created` event with the user message item that is created from the audio buffer.

**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. This will correspond to the end of audio sent to the model, and thus includes the `min_silence_duration_ms` configured in the Session.

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

input_audio_buffer.speech_stopped

```
{
    "event_id": "event_1718",
    "type": "input_audio_buffer.speech_stopped",
    "audio_end_ms": 2000,
    "item_id": "msg_003"
}
```

## response.created

Returned when a new Response is created. The first event of response creation, where the response is in an initial state of `in_progress`.

**event_id** string
The unique ID of the server event.

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

**response** object

The response resource.

Show properties

response.created

```
{
    "event_id": "event_2930",
    "type": "response.created",
    "response": {
        "id": "resp_001",
        "object": "realtime.response",
        "status": "in_progress",
        "status_details": null,
        "output": [],
        "usage": null
    }
}
```

## response.done

Returned when a Response is done streaming. Always emitted, no matter the final state. The Response object included in the `response.done` event will include all output Items in the Response but will omit the raw audio data.

**event_id** string
The unique ID of the server event.

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

**response** object
The response resource.

Show properties

response.done

```
{
    "event_id": "event_3132",
    "type": "response.done",
    "response": {
        "id": "resp_001",
        "object": "realtime.response",
        "status": "completed",
        "status_details": null,
        "output": [
```

```
            {
                "id": "msg_006",
                "object": "realtime.item",
                "type": "message",
                "status": "completed",
                "role": "assistant",
                "content": [
                    {
                        "type": "text",
                        "text": "Sure, how can I assist you today?"
                    }
                ]
            }
        ],
        "usage": {
            "total_tokens":275,
            "input_tokens":127,
            "output_tokens":148,
            "input_token_details": {
                "cached_tokens":0,
                "text_tokens":119,
                "audio_tokens":8
            },
            "output_token_details": {
              "text_tokens":36,
              "audio_tokens":112
            }
        }
    }
}
```


## response.output_item.added

Returned when a new Item is created during Response generation.

**event_id** string
The unique ID of the server event.

**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 to add to the conversation.

Show properties

response.output_item.added


```
{
    "event_id": "event_3334",
    "type": "response.output_item.added",
    "response_id": "resp_001",
    "output_index": 0,
    "item": {
        "id": "msg_007",
        "object": "realtime.item",
        "type": "message",
        "status": "in_progress",
        "role": "assistant",
        "content": []
    }
}
```


## response.output_item.done

Returned when an Item is done streaming. Also emitted when a Response is interrupted, incomplete, or cancelled.

**event_id** string
The unique ID of the server event.

**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 item to add to the conversation.

Show properties

response.output_item.done

```
{
    "event_id": "event_3536",
    "type": "response.output_item.done",
    "response_id": "resp_001",
    "output_index": 0,
    "item": {
        "id": "msg_007",
        "object": "realtime.item",
        "type": "message",
        "status": "completed",
        "role": "assistant",
        "content": [
            {
                "type": "text",
                "text": "Sure, I can help with that."
            }
        ]
    }
}
```

## response.content_part.added

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

**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

```
{
    "event_id": "event_3738",
    "type": "response.content_part.added",
    "response_id": "resp_001",
    "item_id": "msg_007",
    "output_index": 0,
    "content_index": 0,
    "part": {
        "type": "text",
        "text": ""
    }
}
```

## response.content_part.done

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

**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

```
{
    "event_id": "event_3940",
    "type": "response.content_part.done",
    "response_id": "resp_001",
    "item_id": "msg_007",
    "output_index": 0,
    "content_index": 0,
    "part": {
        "type": "text",
        "text": "Sure, I can help with that."
    }
}
```

## response.text.delta

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

**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

```
{
    "event_id": "event_4142",
    "type": "response.text.delta",
    "response_id": "resp_001",
    "item_id": "msg_007",
    "output_index": 0,
    "content_index": 0,
    "delta": "Sure, I can h"
}
```


## response.text.done

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

**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


```
{
    "event_id": "event_4344",
    "type": "response.text.done",
    "response_id": "resp_001",
    "item_id": "msg_007",
    "output_index": 0,
```

```
    "content_index": 0,
    "text": "Sure, I can help with that."
}
```

## response.audio_transcript.delta

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

**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

```
{
    "event_id": "event_4546",
    "type": "response.audio_transcript.delta",
    "response_id": "resp_001",
    "item_id": "msg_008",
    "output_index": 0,
    "content_index": 0,
    "delta": "Hello, how can I a"
}
```

## response.audio_transcript.done

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

**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

```
{
    "event_id": "event_4748",
    "type": "response.audio_transcript.done",
    "response_id": "resp_001",
    "item_id": "msg_008",
    "output_index": 0,
    "content_index": 0,
    "transcript": "Hello, how can I assist you today?"
}
```

# response.audio.delta

Returned when the model-generated audio is updated.

**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

```
{
    "event_id": "event_4950",
    "type": "response.audio.delta",
    "response_id": "resp_001",
    "item_id": "msg_008",
    "output_index": 0,
    "content_index": 0,
    "delta": "Base64EncodedAudioDelta"
}
```

## response.audio.done

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

```
{
    "event_id": "event_5152",
    "type": "response.audio.done",
    "response_id": "resp_001",
    "item_id": "msg_008",
    "output_index": 0,
    "content_index": 0
}
```

## response.function_call_arguments.delta

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

**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


```
{
    "event_id": "event_5354",
    "type": "response.function_call_arguments.delta",
    "response_id": "resp_002",
    "item_id": "fc_001",
    "output_index": 0,
    "call_id": "call_001",
    "delta": "{\"location\": \"San\""
}
```


## response.function_call_arguments.done

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

**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

```
{
    "event_id": "event_5556",
    "type": "response.function_call_arguments.done",
    "response_id": "resp_002",
    "item_id": "fc_001",
    "output_index": 0,
    "call_id": "call_001",
    "arguments": "{\"location\": \"San Francisco\"}"
}
```

## rate_limits.updated

Emitted at the beginning of a Response to indicate the updated rate limits. When a Response is created some tokens will be "reserved" for the output tokens, the rate limits shown here reflect that reservation, which is then adjusted accordingly once the Response is completed.

**event_id** string
The unique ID of the server event.

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

**rate_limits** array
List of rate limit information.

Show properties

rate_limits.updated

```
{
    "event_id": "event_5758",
    "type": "rate_limits.updated",
    "rate_limits": [
        {
            "name": "requests",
            "limit": 1000,
            "remaining": 999,
            "reset_seconds": 60
        },
        {
            "name": "tokens",
            "limit": 50000,
            "remaining": 49950,
            "reset_seconds": 60
        }
```

```
    ]
}
```

# 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

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

Response gpt-3.5-turbo-instruct


```
{
  "id": "cmpl-uqkvlQyYK7bGYrRHQOeXlWi7",
  "object": "text_completion",
  "created": 1589478378,
  "model": "gpt-3.5-turbo-instruct",
  "system_fingerprint": "fp_44709d6fcb",
  "choices": [
    {
      "text": "\n\nThis is indeed a test",
      "index": 0,
      "logprobs": null,
      "finish_reason": "length"
    }
  ],
  "usage": {
    "prompt_tokens": 5,
    "completion_tokens": 7,
    "total_tokens": 12
  }
}
```


## The completion object Legacy

Represents a completion response from the API. Note: both the streamed and non-streamed response objects share the same shape (unlike the chat endpoint).

**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


```
{
  "id": "cmpl-uqkvlQyYK7bGYrRHQOeXlWi7",
  "object": "text_completion",
  "created": 1589478378,
  "model": "gpt-4-turbo",
  "choices": [
    {
      "text": "\n\nThis is indeed a test",
      "index": 0,
      "logprobs": null,
      "finish_reason": "length"
    }
  ],
  "usage": {
    "prompt_tokens": 5,
    "completion_tokens": 7,
    "total_tokens": 12
  }
}
```


## 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 maxium 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

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

Response


```
{
  "id": "asst_abc123",
  "object": "assistant",
  "created_at": 1698984975,
  "name": "Math Tutor",
  "description": null,
  "model": "gpt-4-turbo",
  "instructions": "You are a personal math tutor. When asked a question, write ar
  "tools": [
    {
      "type": "code_interpreter"
    }
  ],
  "file_ids": [],
  "metadata": {},
  "top_p": 1.0,
  "temperature": 1.0,
```

```
  "response_format": "auto"
}
```

# 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

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

Response

```
{
  "id": "file-abc123",
  "object": "assistant.file",
  "created_at": 1699055364,
  "assistant_id": "asst_abc123"
}
```

# 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

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

Response

```
{
  "object": "list",
  "data": [
    {
```

```
      "id": "asst_abc123",
      "object": "assistant",
      "created_at": 1698982736,
      "name": "Coding Tutor",
      "description": null,
      "model": "gpt-4-turbo",
      "instructions": "You are a helpful assistant designed to make me better at
      "tools": [],
      "file_ids": [],
      "metadata": {},
      "top_p": 1.0,
      "temperature": 1.0,
      "response_format": "auto"
    },
    {
      "id": "asst_abc456",
      "object": "assistant",
      "created_at": 1698982718,
      "name": "My Assistant",
      "description": null,
      "model": "gpt-4-turbo",
      "instructions": "You are a helpful assistant designed to make me better at
      "tools": [],
      "file_ids": [],
      "metadata": {},
      "top_p": 1.0,
      "temperature": 1.0,
      "response_format": "auto"
    },
    {
      "id": "asst_abc789",
      "object": "assistant",
      "created_at": 1698982643,
      "name": null,
      "description": null,
      "model": "gpt-4-turbo",
      "instructions": null,
      "tools": [],
      "file_ids": [],
      "metadata": {},
      "top_p": 1.0,
      "temperature": 1.0,
      "response_format": "auto"
    }
  ],
  "first_id": "asst_abc123",
  "last_id": "asst_abc789",
  "has_more": false
}
```

# 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

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

Response


```
{
  "object": "list",
  "data": [
    {
      "id": "file-abc123",
      "object": "assistant.file",
      "created_at": 1699060412,
      "assistant_id": "asst_abc123"
    },
    {
      "id": "file-abc456",
      "object": "assistant.file",
      "created_at": 1699060412,
      "assistant_id": "asst_abc123"
    }
  ],
  "first_id": "file-abc123",
  "last_id": "file-abc456",
  "has_more": false
}
```


# 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

```
curl https://api.openai.com/v1/assistants/asst_abc123 \
  -H "Content-Type: application/json" \

```

```
  -H "Authorization: Bearer $OPENAI_API_KEY" \
```

Response


```
{
  "id": "asst_abc123",
  "object": "assistant",
  "created_at": 1699009709,
  "name": "HR Helper",
  "description": null,
  "model": "gpt-4-turbo",
  "instructions": "You are an HR bot, and you have access to files to answer empl
  "tools": [
    {
      "type": "retrieval"
    }
  ],
  "file_ids": [
    "file-abc123"
  ],
  "metadata": {}
}
```


## 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

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

Response

```
{
  "id": "file-abc123",
  "object": "assistant.file",
  "created_at": 1699055364,
  "assistant_id": "asst_abc123"
}
```

# 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 maxium 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

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

Response


```json
{
  "id": "asst_abc123",
  "object": "assistant",
  "created_at": 1699009709,
  "name": "HR Helper",
  "description": null,
  "model": "gpt-4-turbo",
  "instructions": "You are an HR bot, and you have access to files to answer empl
  "tools": [
    {
      "type": "retrieval"
    }
  ],
  "file_ids": [
    "file-abc123",
    "file-abc456"
  ],
  "metadata": {},
  "top_p": 1.0,
  "temperature": 1.0,
  "response_format": "auto"
}
```


## 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

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

Response

```
{
  "id": "asst_abc123",
  "object": "assistant.deleted",
  "deleted": true
}
```

# 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

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

Response

```
{
  id: "file-abc123",
  object: "assistant.file.deleted",
  deleted: true
}
```

## 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 maxium 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)

```
{
  "id": "asst_abc123",
  "object": "assistant",
  "created_at": 1698984975,
  "name": "Math Tutor",
  "description": null,
  "model": "gpt-4-turbo",
  "instructions": "You are a personal math tutor. When asked a question, write ar
```

```
  "tools": [
    {
      "type": "code_interpreter"
    }
  ],
  "file_ids": [],
  "metadata": {},
  "top_p": 1.0,
  "temperature": 1.0,
  "response_format": "auto"
}
```

## 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)

```
{
  "id": "file-abc123",
  "object": "assistant.file",
  "created_at": 1699055364,
  "assistant_id": "asst_abc123"
}
```

## 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 maxium of 512 characters long.

## Returns

A **thread** object.

Empty Messages

Example request curl

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

Response

```response
{
  "id": "thread_abc123",
  "object": "thread",
  "created_at": 1699012949,
  "metadata": {}
}
```

# 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

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

Response

```response
{
  "id": "thread_abc123",
  "object": "thread",
  "created_at": 1699014083,
  "metadata": {}
}
```

# 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 maxium of 512 characters long.

## Returns

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

Example request curl

```curl
curl https://api.openai.com/v1/threads/thread_abc123 \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "OpenAI-Beta: assistants=v1" \
  -d '{
      "metadata": {
        "modified": "true",
        "user": "abc123"
      }
    }'
```

Response

```response
{
  "id": "thread_abc123",
  "object": "thread",
  "created_at": 1699014083,
  "metadata": {
    "modified": "true",
    "user": "abc123"
  }
}
```

# 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

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

Response

```response
{
  "id": "thread_abc123",
  "object": "thread.deleted",
  "deleted": true
}
```

# 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 maxium of 512 characters long.

The thread object (v1)

```
{
  "id": "thread_abc123",
  "object": "thread",
  "created_at": 1698107661,
  "metadata": {}
}
```

## 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 maxium of 512 characters long.

## Returns

A **message** object.

Example request curl

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

Response

```response
{
  "id": "msg_abc123",
  "object": "thread.message",
  "created_at": 1699017614,
  "thread_id": "thread_abc123",
  "role": "user",
  "content": [
    {
      "type": "text",
      "text": {
        "value": "How does AI work? Explain it in simple terms.",
        "annotations": []
      }
    }
  ],
  "file_ids": [],
  "assistant_id": null,
  "run_id": null,
```

```
  "metadata": {}
}
```

## 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

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

Response


```
{
  "object": "list",
  "data": [
    {
      "id": "msg_abc123",
      "object": "thread.message",
      "created_at": 1699016383,
      "thread_id": "thread_abc123",
      "role": "user",
      "content": [
        {
          "type": "text",
          "text": {
            "value": "How does AI work? Explain it in simple terms.",
            "annotations": []
          }
        }
      ],
      "file_ids": [],
      "assistant_id": null,
      "run_id": null,
      "metadata": {}
    },
    {
      "id": "msg_abc456",
      "object": "thread.message",
      "created_at": 1699016383,
      "thread_id": "thread_abc123",
      "role": "user",
      "content": [
        {
          "type": "text",
          "text": {
            "value": "Hello, what is AI?",
            "annotations": []
          }
        }
      ],
      "file_ids": [
        "file-abc123"
      ],
```

```
      "assistant_id": null,
      "run_id": null,
      "metadata": {}
    }
  ],
  "first_id": "msg_abc123",
  "last_id": "msg_abc456",
  "has_more": false
}
```


## 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

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

Response

```
{
  "object": "list",
  "data": [
    {
      "id": "file-abc123",
      "object": "thread.message.file",
      "created_at": 1699061776,
      "message_id": "msg_abc123"
    },
    {
      "id": "file-abc123",
      "object": "thread.message.file",
      "created_at": 1699061776,
      "message_id": "msg_abc123"
    }
  ],
  "first_id": "file-abc123",
  "last_id": "file-abc123",
  "has_more": false
}
```

# 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

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

Response

```json
{
  "id": "msg_abc123",
  "object": "thread.message",
  "created_at": 1699017614,
  "thread_id": "thread_abc123",
  "role": "user",
  "content": [
    {
      "type": "text",
      "text": {
        "value": "How does AI work? Explain it in simple terms.",
        "annotations": []
      }
    }
  ],
  "file_ids": [],
  "assistant_id": null,
  "run_id": null,
  "metadata": {}
}
```

# 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

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

Response

```
{
  "id": "file-abc123",
  "object": "thread.message.file",
  "created_at": 1699061776,
  "message_id": "msg_abc123"
}
```

# 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 maxium of 512 characters long.

## Returns

The modified **message** object.

Example request curl

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

Response

```
{
  "id": "msg_abc123",
  "object": "thread.message",
  "created_at": 1699017614,
  "thread_id": "thread_abc123",
  "role": "user",
  "content": [
    {
      "type": "text",
      "text": {
        "value": "How does AI work? Explain it in simple terms.",
        "annotations": []
      }
    }
  ],
  "file_ids": [],
```

```
  "assistant_id": null,
  "run_id": null,
  "metadata": {
    "modified": "true",
    "user": "abc123"
  }
}
```

## 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 maxium of 512 characters long.

The message object (v1)

```
{
  "id": "msg_abc123",
  "object": "thread.message",
  "created_at": 1698983503,
  "thread_id": "thread_abc123",
  "role": "assistant",
  "content": [
    {
      "type": "text",
      "text": {
        "value": "Hi! How can I help you today?",
        "annotations": []
      }
    }
  ],
  "file_ids": [],
  "assistant_id": "asst_abc123",
  "run_id": "run_abc123",
  "metadata": {}
}
```

## 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)

```
{
  "id": "file-abc123",
  "object": "thread.message.file",
  "created_at": 1698107661,
  "message_id": "message_QLoItBbqwyAJEzlTy4y9kOMM",
  "file_id": "file-abc123"
}
```

## 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 maxium 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

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

```
    "assistant_id": "asst_abc123"
  }'
```

Response

```
{
  "id": "run_abc123",
  "object": "thread.run",
  "created_at": 1699063290,
  "assistant_id": "asst_abc123",
  "thread_id": "thread_abc123",
  "status": "queued",
  "started_at": 1699063290,
  "expires_at": null,
  "cancelled_at": null,
  "failed_at": null,
  "completed_at": 1699063291,
  "last_error": null,
  "model": "gpt-4-turbo",
  "instructions": null,
  "incomplete_details": null,
  "tools": [
    {
      "type": "code_interpreter"
    }
  ],
  "file_ids": [
    "file-abc123",
    "file-abc456"
  ],
  "metadata": {},
  "usage": null,
  "temperature": 1.0,
  "top_p": 1.0,
  "max_prompt_tokens": 1000,
  "max_completion_tokens": 1000,
  "truncation_strategy": {
    "type": "auto",
    "last_messages": null
  },
  "response_format": "auto",
  "tool_choice": "auto"
}
```

## 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 maxium 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

```
curl https://api.openai.com/v1/threads/runs \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -H "OpenAI-Beta: assistants=v1" \
  -d '{
      "assistant_id": "asst_abc123",
```

```
    "thread": {
      "messages": [
        {"role": "user", "content": "Explain deep learning to a 5 year old."}
      ]
    }
  }'
```

Response

```
{
  "id": "run_abc123",
  "object": "thread.run",
  "created_at": 1699076792,
  "assistant_id": "asst_abc123",
  "thread_id": "thread_abc123",
  "status": "queued",
  "started_at": null,
  "expires_at": 1699077392,
  "cancelled_at": null,
  "failed_at": null,
  "completed_at": null,
  "last_error": null,
  "model": "gpt-4-turbo",
  "instructions": "You are a helpful assistant.",
  "tools": [],
  "file_ids": [],
  "metadata": {},
  "usage": null,
  "temperature": 1
}
```

## 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

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

Response

```response
{
  "object": "list",
  "data": [
    {
      "id": "run_abc123",
      "object": "thread.run",
      "created_at": 1699075072,
      "assistant_id": "asst_abc123",
      "thread_id": "thread_abc123",
      "status": "completed",
      "started_at": 1699075072,
      "expires_at": null,
      "cancelled_at": null,
      "failed_at": null,
      "completed_at": 1699075073,
      "last_error": null,
```

```
    "model": "gpt-4-turbo",
    "instructions": null,
    "incomplete_details": null,
    "tools": [
      {
        "type": "code_interpreter"
      }
    ],
    "file_ids": [
      "file-abc123",
      "file-abc456"
    ],
    "metadata": {},
    "usage": {
      "prompt_tokens": 123,
      "completion_tokens": 456,
      "total_tokens": 579
    },
    "temperature": 1.0,
    "top_p": 1.0,
    "max_prompt_tokens": 1000,
    "max_completion_tokens": 1000,
    "truncation_strategy": {
      "type": "auto",
      "last_messages": null
    },
    "response_format": "auto",
    "tool_choice": "auto"
  },
  {
    "id": "run_abc456",
    "object": "thread.run",
    "created_at": 1699063290,
    "assistant_id": "asst_abc123",
    "thread_id": "thread_abc123",
    "status": "completed",
    "started_at": 1699063290,
    "expires_at": null,
    "cancelled_at": null,
    "failed_at": null,
    "completed_at": 1699063291,
    "last_error": null,
    "model": "gpt-4-turbo",
    "instructions": null,
    "incomplete_details": null,
    "tools": [
      {
        "type": "code_interpreter"
      }
    ],
    "file_ids": [
      "file-abc123",
```

```
        "file-abc456"
      ],
      "metadata": {},
      "usage": {
        "prompt_tokens": 123,
        "completion_tokens": 456,
        "total_tokens": 579
      },
      "temperature": 1.0,
      "top_p": 1.0,
      "max_prompt_tokens": 1000,
      "max_completion_tokens": 1000,
      "truncation_strategy": {
        "type": "auto",
        "last_messages": null
      },
      "response_format": "auto",
      "tool_choice": "auto"
    }
  ],
  "first_id": "run_abc123",
  "last_id": "run_abc456",
  "has_more": false
}
```


## 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

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

Response

```json
{
  "object": "list",
  "data": [
    {
      "id": "step_abc123",
      "object": "thread.run.step",
      "created_at": 1699063291,
      "run_id": "run_abc123",
      "assistant_id": "asst_abc123",
      "thread_id": "thread_abc123",
      "type": "message_creation",
      "status": "completed",
      "cancelled_at": null,
      "completed_at": 1699063291,
      "expired_at": null,
      "failed_at": null,
      "last_error": null,
      "step_details": {
        "type": "message_creation",
        "message_creation": {
```

```
          "message_id": "msg_abc123"
        }
      },
      "usage": {
        "prompt_tokens": 123,
        "completion_tokens": 456,
        "total_tokens": 579
      }
    }
  ],
  "first_id": "step_abc123",
  "last_id": "step_abc456",
  "has_more": false
}
```

## Retrieve run (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

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

Response

```
{
  "id": "run_abc123",
```

```
  "object": "thread.run",
  "created_at": 1699075072,
  "assistant_id": "asst_abc123",
  "thread_id": "thread_abc123",
  "status": "completed",
  "started_at": 1699075072,
  "expires_at": null,
  "cancelled_at": null,
  "failed_at": null,
  "completed_at": 1699075073,
  "last_error": null,
  "model": "gpt-4-turbo",
  "instructions": null,
  "incomplete_details": null,
  "tools": [
    {
      "type": "code_interpreter"
    }
  ],
  "file_ids": [
    "file-abc123",
    "file-abc456"
  ],
  "metadata": {},
  "usage": {
    "prompt_tokens": 123,
    "completion_tokens": 456,
    "total_tokens": 579
  },
  "temperature": 1.0,
  "top_p": 1.0,
  "max_prompt_tokens": 1000,
  "max_completion_tokens": 1000,
  "truncation_strategy": {
    "type": "auto",
    "last_messages": null
  },
  "response_format": "auto",
  "tool_choice": "auto"
}
```


## 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

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

Response

```
{
  "id": "step_abc123",
  "object": "thread.run.step",
  "created_at": 1699063291,
  "run_id": "run_abc123",
  "assistant_id": "asst_abc123",
  "thread_id": "thread_abc123",
  "type": "message_creation",
  "status": "completed",
  "cancelled_at": null,
  "completed_at": 1699063291,
  "expired_at": null,
  "failed_at": null,
  "last_error": null,
  "step_details": {
    "type": "message_creation",
    "message_creation": {
      "message_id": "msg_abc123"
    }
  },
  "usage": {
    "prompt_tokens": 123,
    "completion_tokens": 456,
    "total_tokens": 579
```

```
  }
}
```

## 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 maxium of 512 characters long.

### Returns

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

Example request curl

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

Response

```
{
  "id": "run_abc123",
  "object": "thread.run",
  "created_at": 1699075072,
  "assistant_id": "asst_abc123",
  "thread_id": "thread_abc123",
  "status": "completed",
  "started_at": 1699075072,
  "expires_at": null,
  "cancelled_at": null,
  "failed_at": null,
  "completed_at": 1699075073,
  "last_error": null,
  "model": "gpt-4-turbo",
  "instructions": null,
  "incomplete_details": null,
  "tools": [
    {
      "type": "code_interpreter"
    }
  ],
  "file_ids": [
    "file-abc123",
    "file-abc456"
  ],
  "metadata": {
    "user_id": "user_abc123"
  },
  "usage": {
    "prompt_tokens": 123,
    "completion_tokens": 456,
    "total_tokens": 579
  },
  "temperature": 1.0,
  "top_p": 1.0,
  "max_prompt_tokens": 1000,
  "max_completion_tokens": 1000,
  "truncation_strategy": {
    "type": "auto",
    "last_messages": null
  },
  "response_format": "auto",
  "tool_choice": "auto"
}
```


## Submit tool outputs to run (v1) Legacy

```
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

```curl
curl https://api.openai.com/v1/threads/thread_123/runs/run_123/submit_tool_output
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -H "OpenAI-Beta: assistants=v1" \
  -d '{
    "tool_outputs": [
      {
        "tool_call_id": "call_001",
        "output": "70 degrees and sunny."
```

```
      }
    ]
  }'
```

Response

```
{
  "id": "run_123",
  "object": "thread.run",
  "created_at": 1699075592,
  "assistant_id": "asst_123",
  "thread_id": "thread_123",
  "status": "queued",
  "started_at": 1699075592,
  "expires_at": 1699076192,
  "cancelled_at": null,
  "failed_at": null,
  "completed_at": null,
  "last_error": null,
  "model": "gpt-4-turbo",
  "instructions": null,
  "incomplete_details": null,
  "tools": [
    {
      "type": "function",
      "function": {
        "name": "get_current_weather",
        "description": "Get the current weather in a given location",
        "parameters": {
          "type": "object",
          "properties": {
            "location": {
              "type": "string",
              "description": "The city and state, e.g. San Francisco, CA"
            },
            "unit": {
              "type": "string",
              "enum": ["celsius", "fahrenheit"]
            }
          },
          "required": ["location"]
        }
      }
    }
  ],
  "file_ids": [],
  "metadata": {},
  "usage": null,
  "temperature": 1.0,
  "top_p": 1.0,
  "max_prompt_tokens": 1000,
  "max_completion_tokens": 1000,
```

```
  "truncation_strategy": {
    "type": "auto",
    "last_messages": null
  },
  "response_format": "auto",
  "tool_choice": "auto"
}
```

## 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

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

Response

```
{
  "id": "run_abc123",
  "object": "thread.run",
  "created_at": 1699076126,
  "assistant_id": "asst_abc123",
  "thread_id": "thread_abc123",
  "status": "cancelling",
  "started_at": 1699076126,
```

```
  "expires_at": 1699076726,
  "cancelled_at": null,
  "failed_at": null,
  "completed_at": null,
  "last_error": null,
  "model": "gpt-4-turbo",
  "instructions": "You summarize books.",
  "tools": [
    {
      "type": "retrieval"
    }
  ],
  "file_ids": [],
  "metadata": {},
  "usage": null,
  "temperature": 1.0,
  "top_p": 1.0,
}
```

## 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 maxium 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)

```
{
  "id": "run_abc123",
  "object": "thread.run",
  "created_at": 1698107661,
  "assistant_id": "asst_abc123",
  "thread_id": "thread_abc123",
```

```
  "status": "completed",
  "started_at": 1699073476,
  "expires_at": null,
  "cancelled_at": null,
  "failed_at": null,
  "completed_at": 1699073498,
  "last_error": null,
  "model": "gpt-4-turbo",
  "instructions": null,
  "tools": [{"type": "retrieval"}, {"type": "code_interpreter"}],
  "file_ids": [],
  "metadata": {},
  "incomplete_details": null,
  "usage": {
    "prompt_tokens": 123,
    "completion_tokens": 456,
    "total_tokens": 579
  },
  "temperature": 1.0,
  "top_p": 1.0,
  "max_prompt_tokens": 1000,
  "max_completion_tokens": 1000,
  "truncation_strategy": {
    "type": "auto",
    "last_messages": null
  },
  "response_format": "auto",
  "tool_choice": "auto"
}
```

## 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 maxium of 512 characters long.

**usage**

The run step object (v1)

```
{
  "id": "step_abc123",
```

```
  "object": "thread.run.step",
  "created_at": 1699063291,
  "run_id": "run_abc123",
  "assistant_id": "asst_abc123",
  "thread_id": "thread_abc123",
  "type": "message_creation",
  "status": "completed",
  "cancelled_at": null,
  "completed_at": 1699063291,
  "expired_at": null,
  "failed_at": null,
  "last_error": null,
  "step_details": {
    "type": "message_creation",
    "message_creation": {
      "message_id": "msg_abc123"
    }
  },
  "usage": {
    "prompt_tokens": 123,
    "completion_tokens": 456,
    "total_tokens": 579
  }
}
```

## Streaming (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)

```
{
  "id": "msg_123",
  "object": "thread.message.delta",
  "delta": {
    "content": [
      {
        "index": 0,
        "type": "text",
        "text": { "value": "Hello", "annotations": [] }
      }
    ]
  }
}
```

## 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)

```
{
  "id": "step_123",
  "object": "thread.run.step.delta",
  "delta": {
    "step_details": {
      "type": "tool_calls",
      "tool_calls": [
        {
          "index": 0,
          "id": "call_123",
```

```
            "type": "code_interpreter",
            "code_interpreter": { "input": "", "outputs": [] }
          }
        ]
      }
    }
}
```

## Assistant stream events (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.