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@^4.0.0
```

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,

> **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.
>
> **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-R4EOK4IlOxoFL8jKhBVr9u0b" \
  -H "OpenAI-Project: $PROJECT_ID"
```

Example with the `openai` Python package:

```python
from openai import OpenAI

client = OpenAI(
  organization='org-R4EOK4IIOxoFL8jKhBVr9u0b',
  project='$PROJECT_ID',
)
```

Example with the `openai` Node.js package:

```javascript
import OpenAI from "openai";

const openai = new OpenAI({
    organization: "org-R4EOK4IIOxoFL8jKhBVr9u0b",
    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.

```bash
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
    },
    "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 in 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.

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 python

```python
from pathlib import Path
import openai

speech_file_path = Path(__file__).parent / "speech.mp3"
response = openai.audio.speech.create(
  model="tts-1",
  voice="alloy",
  input="The quick brown fox jumped over the lazy dog."
)
response.stream_to_file(speech_file_path)
```

Create transcription

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

Default Word timestamps Segment timestamps

Transcribes audio into the input language.

Request body

file file Required

Example request python

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

```

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 transcript 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**.

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.

```
4 audio_file = open("speech.mp3", "rb")
5 transcript = client.audio.transcriptions.create(
6   model="whisper-1",
7   file=audio_file
8 )
```

Response

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

Example request python

```
1 from openai import OpenAI
2 client = OpenAI()
3
4 audio_file = open("speech.mp3", "rb")
5 transcript = client.audio.translations.create(
6   model="whisper-1",
7   file=audio_file
8 )
```

Response

```
1 {
2   "text": "Hello, my name is Wolfgang and I come from Germany.
```

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 transcript 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.

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'
}
```


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. F
  "segments": [
    {
      "id": 0,
      "seek": 0,
      "start": 0.0,
      "end": 3.319999933242798,
      "text": " The beach was a popular spot on a hot summer da
      "tokens": [
        50364, 440, 7534, 390, 257, 3743, 4008, 322, 257, 2368,
      ],
      "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.

Request body

messages array Required
A list of messages comprising the conversation so far. **Example Python code**.
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.

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 integer or null Optional
The maximum number of **tokens** that can be generated in the chat completion.

Default | Image input | Streaming | Functions | l

Example request gpt-4o python

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

completion = client.chat.completions.create(
  model="gpt-4o",
  messages=[
    {"role": "system", "content": "You are a helpful assistant."},
    {"role": "user", "content": "Hello!"}
  ]
)

print(completion.choices[0].message)
```

Response

```
{
  "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,
```

The total length of input tokens and generated tokens is limited by the model's context length. **Example Python code** for counting tokens.

n integer or null Optional Defaults to 1
How many chat completion choices to generate for each input message. Note that you will be charged based on the number of generated tokens across all of the choices. Keep `n` as `1` to minimize costs.

presence_penalty number or null Optional Defaults to 0
Number between -2.0 and 2.0. Positive values penalize new tokens based on whether they appear in the text so far, increasing the model's likelihood to talk about new topics.

See more information about frequency and presence penalties.

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

seed integer or null Optional
This feature is in Beta. If specified, our system will make a best effort to sample deterministically, such that repeated requests with the same `seed` and parameters should return the same result. Determinism is not guaranteed, and you should refer to the `system_fingerprint` response parameter to monitor changes in the backend.

service_tier string or null Optional Defaults to null
Specifies the latency tier to use for processing the request. This parameter is relevant for customers subscribed to the scale tier service:

> If set to 'auto', the system will utilize scale tier credits until they are exhausted.
>
> If set to 'default', the request will be processed using the default service tier with a lower uptime SLA and no latency guarentee.

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.

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 chat completion object


```
{
  "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
  }
}
```

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

usage object
Usage statistics for the completion request.
Show properties

The chat completion chunk object

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

The chat completion chunk object

```
{"id":"chatcmpl-123","object":"chat.completion.chunk","created":

{"id":"chatcmpl-123","object":"chat.completion.chunk","created":

....

{"id":"chatcmpl-123","object":"chat.completion.chunk","created":
```

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

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

Example request python

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**.

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

client.embeddings.create(
  model="text-embedding-ada-002",
  input="The food was delicious and the waiter...",
  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
  }
}
```

Returns

A list of **embedding** objects.

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 18 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-3.5-turbo: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

Default | Epochs | Validation file | W&B Integration

Example request python

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

client.fine_tuning.jobs.create(
  training_file="file-abc123",
  model="gpt-3.5-turbo"
)
```

Response

```
{
  "object": "fine_tuning.job",
  "id": "ftjob-abc123",
  "model": "gpt-3.5-turbo-0125",
  "created_at": 1614807352,
  "fine_tuned_model": null,
  "organization_id": "org-123",
  "result_files": [],
  "status": "queued",
  "validation_file": null,
  "training_file": "file-abc123",
}
```

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.

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 python

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

client.fine_tuning.jobs.list()
```

Response

```
{
  "object": "list",
  "data": [
    {
      "object": "fine_tuning.job.event",
      "id": "ft-event-TjX0lMfOniCZX64t9PUQT5hn",
      "created_at": 1689813489,
      "level": "warn",
      "message": "Fine tuning process stopping due to job cance
      "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

Example request python

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

client.fine_tuning.jobs.list_events(
  fine_tuning_job_id="ftjob-abc123",
  limit=2
)
```

Response

```
{
  "object": "list",
  "data": [
    {
      "object": "fine_tuning.job.event",
```

Number of events to retrieve.

Returns

A list of fine-tuning event objects.


```
{
  "id": "ft-event-ddTJfwuMVpfLXseO0Am0Gqjm",
  "created_at": 1692407401,
  "level": "info",
  "message": "Fine tuning job successfully completed",
  "data": null,
  "type": "message"
},
{
  "object": "fine_tuning.job.event",
  "id": "ft-event-tyiGuB72evQncpH87xe505Sv",
  "created_at": 1692407400,
  "level": "info",
  "message": "New fine-tuned model created: ft:gpt-3.5-tu
  "data": null,
  "type": "message"
```


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/chec
  -H "Authorization: Bearer $OPENAI_API_KEY"
```

Response


```
{
  "object": "list"
  "data": [
    {
      "object": "fine_tuning.job.checkpoint",
      "id": "ftckpt_zc4Q7MP6XxulcVzj4MZdwsAB",
      "created_at": 1519129973,
      "fine_tuned_model_checkpoint": "ft:gpt-3.5-turbo-0125:m
      "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": 1519129833,
      "fine_tuned_model_checkpoint": "ft:gpt-3.5-turbo-0125:m
      "metrics": {
        "full_valid_loss": 0.167,
        "full_valid_mean_token_accuracy": 0.781
      },
```


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

Example request python

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

client.fine_tuning.jobs.retrieve("ftjob-abc123")
```

fine_tuning_job_id string Required
The ID of the fine-tuning job.

Returns

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

Response

```
{
  "object": "fine_tuning.job",
  "id": "ftjob-abc123",
  "model": "davinci-002",
  "created_at": 1692661014,
  "finished_at": 1692661190,
  "fine_tuned_model": "ft:davinci-002:my-org:custom_suffix:7
  "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 python

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

client.fine_tuning.jobs.cancel("ftjob-abc123")
```

Response

```
{
  "object": "fine_tuning.job",
  "id": "ftjob-abc123",
  "model": "gpt-3.5-turbo-0125",
  "created_at": 1689376978,
  "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

Training format for chat models

```
{
  "messages": [
```

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

```
{ "role": "user", "content": "What is the weather in San Fra
{
  "role": "assistant",
  "tool_calls": [
    {
      "id": "call_id",
      "type": "function",
      "function": {
        "name": "get_current_weather",
        "arguments": "{\"location\": \"San Francisco, USA\", \'
      }
    }
  ]
}
],
"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 Franc
          },
          "format": { "type": "string", "enum": ["celsius", "fahren
        },
        "required": ["location", "format"]
      }
    }
  }
]
}
```

Training format for completions models

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

Training format for completions models

prompt string
The input prompt for this training example.

completion string
The desired completion for this training example.

```
{
  "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.

The fine-tuning job object

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

created_at integer

```
{
  "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:7
```

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


```
{
  "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
}
```

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 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-3.5-turbo-0125:my-org
  "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 python

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

client.batches.create(
  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": {
```

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 python

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

client.batches.retrieve("batch_abc123")
```

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
```

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 python

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

client.batches.cancel("batch_abc123")
```

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
  },
```

List batch

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

List your organization's batches.

Query parameters

Example request python

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

client.batches.list()
```

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.

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,
```


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 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 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 maxium of 512 characters long.

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

The request input object

{"custom_id": "request-1", "method": "POST", "url": "/v1/chat/compl

supported.

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": {"

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.

Example request python

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

client.files.create(
  file=open("mydata.jsonl", "rb"),
  purpose="fine-tune"
)
```

Response

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

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.

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 python

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

client.files.list()
```

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 python

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

client.files.retrieve("file-abc123")
```

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 python

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

client.files.delete("file-abc123")
```

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 python

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

content = client.files.content("file-abc123")
```

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 file object

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

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`.

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.

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
}
```

Returns

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

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="aHR0cHM6Ly9hcGkub3BlbmFpLmNvbS92MS91cGx
```

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

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": {
```

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.

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.

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",
```

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.

```
17   }
18 }
```

The upload part object

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

The upload part object

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

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`.

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.

Example request python

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

client.images.generate(
  model="dall-e-3",
  prompt="A cute baby sea otter",
  n=1,
  size="1024x1024"
)
```

Response

n integer or null Optional Defaults to 1
The number of images to generate. Must be between 1 and 10. For `dall-e-3`, only `n=1` is supported.

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

response_format string or null Optional Defaults to url
The format in which the generated images are returned. Must be one of `url` or `b64_json`. URLs are only valid for 60 minutes after the image has been generated.

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

style string or null Optional Defaults to vivid
The style of the generated images. Must be one of `vivid` or `natural`. Vivid causes the model to lean towards generating hyper-real and dramatic images. Natural causes the model to produce more natural, less hyper-real looking images. This param is only supported for `dall-e-3`.

user string Optional
A unique identifier representing your end-user, which can help OpenAI to monitor and detect abuse. **Learn more**.

Returns

Returns a list of **image** objects.

```
{
  "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

Example request python

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

client.images.edit(
  image=open("otter.png", "rb"),
  mask=open("mask.png", "rb"),
  prompt="A cute baby sea otter wearing a beret",
  n=2,
  size="1024x1024"
)
```

Response

```
{
  "created": 1589478378,
  "data": [
```

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 1024x1024
The size of the generated images. Must be one of `256x256`, `512x512`, or `1024x1024`.

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.

```
4    {
5      "url": "https://..."
6    },
7    {
8      "url": "https://..."
9    }
10  ]
11 }
```

Create image variation

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

Creates a variation of a given image.

Request body

image file Required
The image to use as the basis for the variation(s). Must be a valid PNG file, less than 4MB, and square.

model string Optional Defaults to dall-e-2
The model to use for image generation. Only `dall-e-2` is supported at this time.

n integer or null Optional Defaults to 1
The number of images to generate. Must be between 1 and 10. For `dall-e-3`, only `n=1` is supported.

response_format string or null Optional Defaults to url
The format in which the generated images are returned. Must be one of `url` or `b64_json`. URLs are only valid for 60 minutes after the image has been generated.

Example request python

```
1  from openai import OpenAI
2  client = OpenAI()
3
4  response = client.images.create_variation(
5    image=open("image_edit_original.png", "rb"),
6    n=2,
7    size="1024x1024"
8  )
```

Response

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

size string or null Optional Defaults to 1024x1024
The size of the generated images. Must be one of `256x256`, `512x512`, or `1024x1024`.

user string Optional
A unique identifier representing your end-user, which can help OpenAI to monitor and detect abuse. **Learn more**.

Returns

Returns a list of **image** objects.

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 python

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

client.models.list()
```

Response

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

```
    "object": "model",
    "created": 1686935002,
    "owned_by": "organization-owner",
  },
  {
    "id": "model-id-2",
    "object": "model",
    "created": 1686935002,
    "owned_by": "openai"
  },
],
"object": "list"
```

Retrieve model

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

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

Path parameters

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

Returns

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

Example req… gpt-3.5-turbo-instruct python

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

client.models.retrieve("gpt-3.5-turbo-instruct")
```

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 python

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

client.models.delete("ft:gpt-3.5-turbo:acemeco:suffix:abc123")
```

Response

```
{
  "id": "ft:gpt-3.5-turbo: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 model object

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

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.

Moderations

Given some input text, outputs if the model classifies it as potentially harmful across several categories.

Related guide: **Moderations**

Create moderation

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

Classifies if text is potentially harmful.

Request body

input string or array Required
The input text to classify

model string Optional Defaults to text-moderation-latest
Two content moderations models are available: `text-moderation-stable` and `text-moderation-latest`.

The default is `text-moderation-latest` which will be automatically upgraded over time. This ensures you are always using our most accurate model. If you use `text-moderation-stable`, we will provide advanced notice before updating the model. Accuracy of `text-moderation-stable` may be slightly lower than for `text-moderation-latest`.

Returns

A **moderation** object.

Example request python

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

moderation = client.moderations.create(input="I want to kill them
print(moderation)
```

Response

```
{
  "id": "modr-XXXXX",
  "model": "text-moderation-005",
  "results": [
    {
      "flagged": true,
      "categories": {
        "sexual": false,
        "hate": false,
        "harassment": false,
        "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": 1.2282071e-06,
        "hate": 0.010696256,
```

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

The moderation object

```
{
  "id": "modr-XXXXX",
  "model": "text-moderation-005",
  "results": [
    {
      "flagged": true,
      "categories": {
```

A list of moderation objects.

Show properties

```
      "sexual": false,
      "hate": false,
      "harassment": false,
      "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": 1.2282071e-06,
      "hate": 0.010696256,
      "harassment": 0.29842457,
      "self-harm": 1.5236925e-08,
      "sexual/minors": 5.7246268e-08,
      "hate/threatening": 0.0060676364,
      "violence/graphic": 4.435014e-06,
      "self-harm/intent": 8.098441e-10,
      "self-harm/instructions": 2.8498655e-11,
      "harassment/threatening": 0.63055265,
      "violence": 0.99011886,
    }
  }
 ]
}
```

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

Code Interpreter Files

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 []

Example request python

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

my_assistant = client.beta.assistants.create(
    instructions="You are a personal math tutor. When asked a q
    name="Math Tutor",
    tools=[{"type": "code_interpreter"}],
    model="gpt-4-turbo",
)
print(my_assistant)
```

Response

```
{
  "id": "asst_abc123",
  "object": "assistant",
  "created_at": 1698984975,
  "name": "Math Tutor",
  "description": null,
  "model": "gpt-4-turbo",
```

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 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.

List assistants Beta

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

Returns a list of assistants.

```
8   "instructions": "You are a personal math tutor. When asked
9   "tools": [
10    {
11      "type": "code_interpreter"
12    }
13  ],
14  "metadata": {},
```

Example request python

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.

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

my_assistants = client.beta.assistants.list(
    order="desc",
    limit="20",
)
print(my_assistants.data)
```

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 m
      "tools": [],
      "tool_resources": {},
      "metadata": {},
      "top_p": 1.0,
      "temperature": 1.0,
      "response_format": "auto"
    },
    {
```


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 python

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

my_assistant = client.beta.assistants.retrieve("asst_abc123")
print(my_assistant)
```

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
  "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 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

Example request python

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

my_updated_assistant = client.beta.assistants.update(
  "asst_abc123",
  instructions="You are an HR bot, and you have access to files
  name="HR Helper",
  tools=[{"type": "file_search"}],
  model="gpt-4-turbo"
)

print(my_updated_assistant)
```

Response

```
{
  "id": "asst_123",
  "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
  "tools": [
    {
      "type": "file_search"
    }
  ],
  "tool_resources": {
```

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.

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 python

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

response = client.beta.assistants.delete("asst_abc123")
print(response)
```

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 assistant object

```
{
  "id": "asst_abc123",
  "object": "assistant",
```

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 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.


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

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

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 maxium of 512 characters long.

Returns

A **thread** object.

Empty Messages

Example request python

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

empty_thread = client.beta.threads.create()
print(empty_thread)
```

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.

Example request python

Path parameters

thread_id string Required
The ID of the thread to retrieve.

Returns

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

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

my_thread = client.beta.threads.retrieve("thread_abc123")
print(my_thread)
```

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 maxium of 512 characters long.

Returns

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

Example request python

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

my_updated_thread = client.beta.threads.update(
  "thread_abc123",
  metadata={
    "modified": "true",
    "user": "abc123"
  }
)
print(my_updated_thread)
```

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 python

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

response = client.beta.threads.delete("thread_abc123")
print(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 maxium 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

Example request python

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

thread_message = client.beta.threads.messages.create(
  "thread_abc123",
  role="user",
```

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 maxium of 512 characters long.

Returns

A **message** object.

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

```
7   content="How does AI work? Explain it in simple terms.",
8 )
9 print(thread_message)
```

Response

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

Example request python

```
1 from openai import OpenAI
2 client = OpenAI()
3
4 thread_messages = client.beta.threads.messages.list("thread_a
5 print(thread_messages.data)
```

Response

```
1  {
2    "object": "list",
3    "data": [
4      {
5        "id": "msg_abc123",
6        "object": "thread.message",
7        "created_at": 1699016383,
8        "assistant_id": null,
9        "thread_id": "thread_abc123",
```

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.

```
10      "run_id": null,
11      "role": "user",
12      "content": [
13        {
14          "type": "text",
15          "text": {
16            "value": "How does AI work? Explain it in simple term
17            "annotations": []
18          }
19        }
20      ],
21      "attachments": [],
```

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 python

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

message = client.beta.threads.messages.retrieve(
  message_id="msg_abc123",
  thread_id="thread_abc123",
)
print(message)
```

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": [],
```

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 maxium of 512 characters long.

Returns

The modified **message** object.

Example request python

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

message = client.beta.threads.messages.update(
  message_id="msg_abc12",
  thread_id="thread_abc123",
  metadata={
    "modified": "true",
    "user": "abc123",
  },
)
print(message)
```

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": []
      }
```

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 python

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

deleted_message = client.beta.threads.messages.delete(
  message_id="msg_abc12",
  thread_id="thread_abc123",
)
print(deleted_message)
```

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

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": {}
}
```

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.

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.

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.

Default Streaming Streaming with Functions

Example request python

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

run = client.beta.threads.runs.create(
  thread_id="thread_abc123",
  assistant_id="asst_abc123"
)

print(run)
```

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": [
```

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 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.

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

Default Streaming Streaming with Functions

Example request python

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

run = client.beta.threads.create_and_run(
  assistant_id="asst_abc123",
  thread={
    "messages": [
      {"role": "user", "content": "Explain deep learning to a 5 ye
    ]
  }
)

print(run)
```

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,
```

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 `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 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.

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

Example request python

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

runs = client.beta.threads.runs.list(
  "thread_abc123"
)

print(runs)
```

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,
```

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.

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 python

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

run = client.beta.threads.runs.retrieve(
  thread_id="thread_abc123",
  run_id="run_abc123"
)

print(run)
```

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"
```

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

Example request python

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

run = client.beta.threads.runs.update(
  thread_id="thread_abc123",
  run_id="run_abc123",
  metadata={"user_id": "user_abc123"},
)
```

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.

```
9
10 print(run)
```

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": [
    {
```

Submit tool outputs to run Beta

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

Default Streaming

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

Example request python

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

run = client.beta.threads.runs.submit_tool_outputs(
  thread_id="thread_123",
  run_id="run_123",
  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,
```

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

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 python

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

run = client.beta.threads.runs.cancel(
  thread_id="thread_abc123",
  run_id="run_abc123"
)

print(run)
```

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": "file_search"
    }
```

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 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-4-turbo",
  "instructions": null,
  "tools": [{"type": "file_search"}, {"type": "code_interpreter"}],
  "metadata": {},
  "incomplete_details": null,
```

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 maxium 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


```
"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
}
```

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 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

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.

Returns

A list of **run step** objects.

Example request python

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

run_steps = client.beta.threads.runs.steps.list(
    thread_id="thread_abc123",
    run_id="run_abc123"
)

print(run_steps)
```

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",
```

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.

Example request python

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

run_step = client.beta.threads.runs.steps.retrieve(
    thread_id="thread_abc123",
    run_id="run_abc123",
    step_id="step_abc123"
```

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.

```
)

print(run_step)
```

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"
```

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 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
  }
}
```

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 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

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

Example request python

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

vector_store = client.beta.vector_stores.create(
  name="Support FAQ"
)
print(vector_store)
```

Response

```
{
  "id": "vs_abc123",
  "object": "vector_store",
  "created_at": 1699061776,
  "name": "Support FAQ",
  "bytes": 139920,
```

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 maxium of 512 characters long.

Returns

A **vector store** object.

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

List vector stores Beta

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

Returns a list of vector stores.

Query parameters

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

order string Optional Defaults to desc
Sort order by the `created_at` timestamp of the objects. `asc` for ascending order and `desc` for descending order.

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

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

Returns

A list of **vector store** objects.

Example request python

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

vector_stores = client.beta.vector_stores.list()
print(vector_stores)
```

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,
```

Retrieve vector store Beta

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

Retrieves a vector store.

Example request python

Path parameters

vector_store_id string Required
The ID of the vector store to retrieve.

Returns

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

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

vector_store = client.beta.vector_stores.retrieve(
  vector_store_id="vs_abc123"
)
print(vector_store)
```

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 maxium of 512 characters long.

Returns

The modified **vector store** object.

Example request python

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

vector_store = client.beta.vector_stores.update(
  vector_store_id="vs_abc123",
  name="Support FAQ"
)
print(vector_store)
```

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

Example request python

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

deleted_vector_store = client.beta.vector_stores.delete(
```

vector_store_id string Required
The ID of the vector store to delete.

Returns

Deletion status

```
vector_store_id="vs_abc123"
)
print(deleted_vector_store)
```

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

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
}
```

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

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 python

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

vector_store_file = client.beta.vector_stores.files.create(
  vector_store_id="vs_abc123",
  file_id="file-abc123"
)
print(vector_store_file)
```

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

Example request python

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

vector_store_files = client.beta.vector_stores.files.list(
  vector_store_id="vs_abc123"
)
print(vector_store_files)
```

Response

```
{
  "object": "list",
  "data": [
    {
      "id": "file-abc123",
```

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.

```
      "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 python

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

vector_store_file = client.beta.vector_stores.files.retrieve(
  vector_store_id="vs_abc123",
  file_id="file-abc123"
)
print(vector_store_file)
```

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

Example request python

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

```

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

```
deleted_vector_store_file = client.beta.vector_stores.files.dele
    vector_store_id="vs_abc123",
    file_id="file-abc123"
)
print(deleted_vector_store_file)
```

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 python

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

vector_store_file_batch = client.beta.vector_stores.file_batches.
  vector_store_id="vs_abc123",
  file_ids=["file-abc123", "file-abc456"]
)
print(vector_store_file_batch)
```

Response

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

Retrieve vector store file batch 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 python

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

vector_store_file_batch = client.beta.vector_stores.file_batches.
  vector_store_id="vs_abc123",
  batch_id="vsfb_abc123"
)
print(vector_store_file_batch)
```

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,
```

```
13  }
```

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 python

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

deleted_vector_store_file_batch = client.beta.vector_stores.file_
    vector_store_id="vs_abc123",
    file_batch_id="vsfb_abc123"
)
print(deleted_vector_store_file_batch)
```

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_id}/files

Returns a list of vector store files in a batch.

Path parameters

vector_store_id string Required
The ID of the vector store that the files belong to.

batch_id string Required
The ID of the file batch that the files belong to.

Query parameters

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

order string Optional Defaults to desc
Sort order by the `created_at` timestamp of the objects. `asc` for ascending order and `desc` for descending order.

Example request python

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

vector_store_files = client.beta.vector_stores.file_batches.list_fil
  vector_store_id="vs_abc123",
  batch_id="vsfb_abc123"
)
print(vector_store_files)
```

Response

```
{
  "object": "list",
  "data": [
    {
      "id": "file-abc123",
      "object": "vector_store.file",
      "created_at": 1699061776,
      "vector_store_id": "vs_abc123"
    },
    {
      "id": "file-abc456",
```

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`.

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

Returns

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

The vector store files batch object Beta

A batch of files attached to a vector store.

The vector store files batch object

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

```
{
  "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.

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

No streaming Streaming

Example req… gpt-3.5-turbo-instruct python

```
1 from openai import OpenAI
2 client = OpenAI()
3
4 client.completions.create(
```

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.

```
5  model="gpt-3.5-turbo-instruct",
6  prompt="Say this is a test",
7  max_tokens=7,
8  temperature=0
9  )
```

Response gpt-3.5-turbo-instruct

```
1  {
2    "id": "cmpl-uqkvlQyYK7bGYrRHQ0eXlWi7",
3    "object": "text_completion",
4    "created": 1589478378,
5    "model": "gpt-3.5-turbo-instruct",
6    "system_fingerprint": "fp_44709d6fcb",
7    "choices": [
8      {
9        "text": "\n\nThis is indeed a test",
10       "index": 0,
11       "logprobs": null,
12       "finish_reason": "length"
13     }
14   ],
15   "usage": {
16     "prompt_tokens": 5,
17     "completion_tokens": 7,
```

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.

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-uqkvlQyYK7bGYrRHQ0eXlWi7",
  "object": "text_completion",
  "created": 1589478378,
  "model": "gpt-4-turbo",
  "choices": [
    {
      "text": "\n\nThis is indeed a test",
      "index": 0,
      "logprobs": null,
      "finish_reason": "length"
    }
  ],
  "usage": {
    "prompt_tokens": 5,
    "completion_tokens": 7,
    "total_tokens": 12
  }
}
```

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.

Code Interpreter Files

Example request python

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

my_assistant = client.beta.assistants.create(
    instructions="You are a personal math tutor. When asked a q
    name="Math Tutor",
    tools=[{"type": "code_interpreter"}],
    model="gpt-4-turbo",
)
print(my_assistant)
```

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
  "tools": [
    {
      "type": "code_interpreter"
    }
  ],
  "file_ids": [],
  "metadata": {},
```

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.

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.

Example request python

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

assistant_file = client.beta.assistants.files.create(
  assistant_id="asst_abc123",
  file_id="file-abc123"
)
print(assistant_file)
```

Response

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

Returns

An **assistant file** object.

List assistants (v1) Legacy

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

Returns a list of assistants.

Query parameters

Example request python

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

```

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.

```python
my_assistants = client.beta.assistants.list(
    order="desc",
    limit="20",
)
print(my_assistants.data)
```

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


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.

Example request python

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

assistant_files = client.beta.assistants.files.list(
  assistant_id="asst_abc123"
)
print(assistant_files)
```

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"
    }
  ],
```

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.

```
  "first_id": "file-abc123",
  "last_id": "file-abc456",
  "has_more": false
}
```

Returns

A list of **assistant file** objects.

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 python

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

my_assistant = client.beta.assistants.retrieve("asst_abc123")
print(my_assistant)
```

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 file
  "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

Example request python

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

assistant_file = client.beta.assistants.files.retrieve(
  assistant_id="asst_abc123",
  file_id="file-abc123"
)
print(assistant_file)
```

The ID of the file we're getting.

Returns

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

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

Response

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

Example request python

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

my_updated_assistant = client.beta.assistants.update(
  "asst_abc123",
  instructions="You are an HR bot, and you have access to files
  name="HR Helper",
  tools=[{"type": "retrieval"}],
  model="gpt-4-turbo",
  file_ids=["file-abc123", "file-abc456"],
)

print(my_updated_assistant)
```

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
  "tools": [
    {
      "type": "retrieval"
    }
  ],
```

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.

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 python

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

response = client.beta.assistants.delete("asst_abc123")
print(response)
```

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 python

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

deleted_assistant_file = client.beta.assistants.files.delete(
    assistant_id="asst_abc123",
    file_id="file-abc123"
)
print(deleted_assistant_file)
```

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`.

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

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 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 file object (v1)


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

The assistant ID that the file is attached to.

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 python

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

empty_thread = client.beta.threads.create()
print(empty_thread)
```

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 python

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

my_thread = client.beta.threads.retrieve("thread_abc123")
print(my_thread)
```

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

Example request python

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

my_updated_thread = client.beta.threads.update(
```

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.

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

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.

```
5     "thread_abc123",
6     metadata={
7       "modified": "true",
8       "user": "abc123"
9     }
10  )
11  print(my_updated_thread)
```

Response

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

Example request python

```
1  from openai import OpenAI
2  client = OpenAI()
3
4  response = client.beta.threads.delete("thread_abc123")
5  print(response)
```

Response

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

The thread object (v1)

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

Messages (v1) Legacy

Create messages within threads

Related guide: **Assistants**

Create message (v1) Legacy

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

Create a message.

Path parameters

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

Request body

role string Required
The role of the entity that is creating the message. Allowed values include:

- `user`: Indicates the message is sent by an actual user and should be used in most cases to represent user-generated messages.
- `assistant`: Indicates the message is generated by the assistant. Use this value to insert messages from the assistant into the conversation.

content string Required
The content of the message.

file_ids array Optional Defaults to []
A list of **File** IDs that the message should use. There can be a maximum of 10 files attached to a message. Useful for tools like `retrieval` and `code_interpreter` that can access and use files.

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

Returns

A **message** object.

Example request python

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

thread_message = client.beta.threads.messages.create(
  "thread_abc123",
  role="user",
  content="How does AI work? Explain it in simple terms.",
)
print(thread_message)
```

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,
```

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

Example request python

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

```

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.

```
4 thread_messages = client.beta.threads.messages.list("thread
```

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 term
            "annotations": []
          }
        }
      ],
      "file_ids": [],
      "assistant_id": null,
      "run_id": null,
```


Returns

A list of **message** objects.

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

Example request python

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

message_files = client.beta.threads.messages.files.list(
  thread_id="thread_abc123",
  message_id="msg_abc123"
)
print(message_files)
```

Response


```
{
  "object": "list",
  "data": [
    {
      "id": "file-abc123",
```

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.

```
    "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 python

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

message = client.beta.threads.messages.retrieve(
  message_id="msg_abc123",
  thread_id="thread_abc123",
)
print(message)
```

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,
```

```
18  "run_id": null,
```

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 python

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

message_files = client.beta.threads.messages.files.retrieve(
    thread_id="thread_abc123",
    message_id="msg_abc123",
    file_id="file-abc123"
)
print(message_files)
```

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 python

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

message = client.beta.threads.messages.update(
  message_id="msg_abc12",
  thread_id="thread_abc123",
  metadata={
    "modified": "true",
    "user": "abc123",
  },
)
print(message)
```

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": []
      }
    }
```

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.

```
15  ],
```

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": {}
}
```

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 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

Default Streaming Streaming with Functions

Example request python

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

run = client.beta.threads.runs.create(
  thread_id="thread_abc123",
  assistant_id="asst_abc123"
)

print(run)
```

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,
```

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


```
13  "last_error": null,
14  "model": "gpt-4-turbo",
15  "instructions": null,
16  "incomplete_details": null,
17  "tools": [
```

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.

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

Default Streaming Streaming with Functions

Example request python

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

run = client.beta.threads.create_and_run(
  assistant_id="asst_abc123",
  thread={
    "messages": [
      {"role": "user", "content": "Explain deep learning to a 5 ye
    ]
  }
)

print(run)
```

Response

```
{
  "id": "run_abc123",
  "object": "thread.run",
  "created_at": 1699076792,
  "assistant_id": "asst_abc123",
```

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


```
"thread_id": "thread_abc123",
"status": "queued",
"started_at": null,
"expires_at": 1699077392,
"cancelled_at": null,
"failed_at": null,
"completed_at": null,
"last_error": null,
```

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.

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.

Example request python

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

runs = client.beta.threads.runs.list(
  "thread_abc123"
)

print(runs)
```

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": [
```

Returns

A list of **run** objects.

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.

Example request python

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

run_steps = client.beta.threads.runs.steps.list(
    thread_id="thread_abc123",
    run_id="run_abc123"
)

print(run_steps)
```

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",
```

Returns

A list of **run step** objects.

Retrieve run (v1) Legacy

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

Retrieves a run.

Path parameters

Example request python

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

```

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.

```
run = client.beta.threads.runs.retrieve(
  thread_id="thread_abc123",
  run_id="run_abc123"
)

print(run)
```

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"
```


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 python

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

run_step = client.beta.threads.runs.steps.retrieve(
    thread_id="thread_abc123",
    run_id="run_abc123",
    step_id="step_abc123"
)

print(run_step)
```

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": {
```

```
16    "type": "message_creation",
17    "message_creation": {
```

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 python

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

run = client.beta.threads.runs.update(
  thread_id="thread_abc123",
  run_id="run_abc123",
  metadata={"user_id": "user_abc123"},
)

print(run)
```

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": [
    {
```

Submit tool outputs to run (v1) Legacy

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

Default Streaming

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.

Example request python

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

run = client.beta.threads.runs.submit_tool_outputs(
  thread_id="thread_123",
  run_id="run_123",
  tool_outputs=[
    {
      "tool_call_id": "call_001",
      "output": "70 degrees and sunny."
    }
  ]
)
```

Response

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.

```
{
  "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
```

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 python

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

run = client.beta.threads.runs.cancel(
  thread_id="thread_abc123",
  run_id="run_abc123"
)

print(run)
```

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"
    }
```

The run object (v1) Legacy

Represents an execution run on a **thread**.

id string
The identifier, which can be referenced in API endpoints.

The run object (v1)

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

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


```
{
  "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 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 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.

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
  }
}
```

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

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": [] }
        }
      ]
    }
```

```
16   }
17 }
```

Assistant stream events (v1) Legacy

Represents an event emitted when streaming a Run.

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

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

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

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

thread.created `data` is a thread

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

thread.run.created `data` is a run

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

thread.run.queued `data` is a run

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

thread.run.in_progress `data` is a run

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

thread.run.requires_action `data` is a run

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

thread.run.completed `data` is a run

Occurs when a **run** is completed.

thread.run.failed `data` is a run

Occurs when a **run** fails.

thread.run.cancelling `data` is a run

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

thread.run.cancelled `data` is a run

Occurs when a **run** is cancelled.

thread.run.expired `data` is a run

Occurs when a **run** expires.

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

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

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

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

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

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

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

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

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

Occurs when a **run step** fails.

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

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

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

Occurs when a **run step** expires.

thread.message.created `data` is a message

Occurs when a **message** is created.

thread.message.in_progress `data` is a message

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

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

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

thread.message.completed `data` is a message

Occurs when a **message** is completed.

thread.message.incomplete `data` is a message

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

error `data` is an error

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