

Introduction

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

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

```
pip install openai
```

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

```
npm install openai
```

Authentication

The OpenAI API uses API keys for authentication. Visit your [API Keys](#) page to retrieve the API key you'll use in your requests.

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
```

Requesting organization

For users who belong to multiple organizations, you can pass a header to specify which organization is used for an API request. Usage from these API requests will count against the specified organization's subscription quota.

Example curl command:

```
1 curl https://api.openai.com/v1/models \  
2   -H "Authorization: Bearer $OPENAI_API_KEY" \  
3   -H "OpenAI-Organization: org-IxK4tGmaXSg4vMrDb10RYkR2"
```

Example with the `openai` Python package:

```
1 import os  
2 import openai  
3 openai.organization = "org-IxK4tGmaXSg4vMrDb10RYkR2"  
4 openai.api_key = os.getenv("OPENAI_API_KEY")  
5 openai.Model.list()
```

Example with the `openai` Node.js package:

```
1 import { Configuration, OpenAIApi } from "openai";  
2 const configuration = new Configuration({  
3   organization: "org-IxK4tGmaXSg4vMrDb10RYkR2",  
4   apiKey: process.env.OPENAI_API_KEY,  
5 });  
6 const openai = new OpenAIApi(configuration);  
7 const response = await openai.listEngines();
```

Organization IDs can be found on your [Organization settings](#) page.

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.

```
1 curl https://api.openai.com/v1/chat/completions \  
2   -H "Content-Type: application/json" \  
3   -H "Authorization: Bearer $OPENAI_API_KEY" \  
4   -d '{  
5     "model": "gpt-3.5-turbo",
```

```
6     "messages": [{"role": "user", "content": "Say this is a test!"}],
7     "temperature": 0.7
8 }
```

This request queries the `gpt-3.5-turbo` model to complete the text starting with a prompt of "Say *this is a test*". You should get a response back that resembles the following:

```
1  {
2    "id": "chatcmpl-abc123",
3    "object": "chat.completion",
4    "created": 1677858242,
5    "model": "gpt-3.5-turbo-0301",
6    "usage": {
7      "prompt_tokens": 13,
8      "completion_tokens": 7,
9      "total_tokens": 20
10   },
11   "choices": [
12     {
13       "message": {
14         "role": "assistant",
15         "content": "\n\nThis is a test!"
16       },
17       "finish_reason": "stop",
18       "index": 0
19     }
20   ]
21 }
```

Now you've generated your first chat completion. We can see the `finish_reason` is `stop` which means the API returned the full completion generated by the model. In the above request, we only generated a single message but you can set the `n` parameter to generate multiple messages choices. In this example, `gpt-3.5-turbo` is being used for more of a traditional **text completion task**. The model is also optimized for **chat applications** as well.

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.

Example request

python  Copy

```
1 import os
2 import openai
3 openai.api_key = os.getenv("OPENAI_API_KEY")
4 openai.Model.list()
```

Response

 Copy

```
1  {
2    "data": [
3      {
4        "id": "model-id-0",
5        "object": "model",
6        "owned_by": "organization-owner",
7        "permission": [...]
8      },
9      {
10       "id": "model-id-1",
11       "object": "model",
12       "owned_by": "organization-owner",
13       "permission": [...]
14     },
15     {
16       "id": "model-id-2",
17       "object": "model",
18       "owned_by": "openai",
19       "permission": [...]
20     },
21   ],
22   "object": "list"
23 }
```

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

Example request

text-davinci-003 ▾ python ▾  Copy

```
1 import os
2 import openai
3 openai.api_key = os.getenv("OPENAI_API_KEY")
4 openai.Model.retrieve("text-davinci-003")
```

Response

text-davinci-003 ▾  Copy

```
1 {
2   "id": "text-davinci-003",
3   "object": "model",
4   "owned_by": "openai",
5   "permission": [...]
6 }
```

Completions

Given a prompt, the model will return one or more predicted completions, and can also return the probabilities of alternative tokens at each position.

Create completion

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

Creates a completion for the provided prompt and parameters.

Request body

model string **Required**

ID of the model to use. You can use the [List models](#) API to see all of your available models, or see our [Model overview](#) for descriptions of them.

prompt string or array Optional Defaults to <|endoftext|>

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.

suffix string Optional Defaults to null

The suffix that comes after a completion of inserted text.

max_tokens integer Optional Defaults to 16

The maximum number of **tokens** to generate in the completion.

The token count of your prompt plus `max_tokens` cannot exceed the model's context length. Most models have a context length of 2048 tokens (except for the newest models, which support 4096).

temperature number 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 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.

`n` integer 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`.

`stream` boolean 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.

`logprobs` integer Optional Defaults to null

Include the log probabilities on the `logprobs` most likely tokens, as well the chosen tokens. For example, if `logprobs` is 5, the API will return a list of the 5 most likely tokens. The API will always return the `logprob` of the sampled token, so there may be up to `logprobs+1` elements in the response.

The maximum value for `logprobs` is 5. If you need more than this, please contact us through our [Help center](#) and describe your use case.

`echo` boolean Optional Defaults to false

Echo back the prompt in addition to the completion

`stop` string or array 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.

`presence_penalty` number 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.](#)

`frequency_penalty` number 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.](#)

`best_of` integer 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`.

`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](#) (which works for both GPT-2 and GPT-3) 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.

`user` string Optional

A unique identifier representing your end-user, which can help OpenAI to monitor and detect abuse.

[Learn more.](#)

Example request

text-davinci-003 ▾ python ▾ Copy

```
1 import os
2 import openai
3 openai.api_key = os.getenv("OPENAI_API_KEY")
4 openai.Completion.create(
5     model="text-davinci-003",
6     prompt="Say this is a test",
7     max_tokens=7,
8     temperature=0
9 )
```

Parameters

text-davinci-003 ▾ Copy

```
1 {
2     "model": "text-davinci-003",
3     "prompt": "Say this is a test",
4     "max_tokens": 7,
```



```
5     "temperature": 0,  
6     "top_p": 1,  
7     "n": 1,  
8     "stream": false,  
9     "logprobs": null,  
10    "stop": "\n"  
11 }
```

Response

text-davinci-003   Copy

```
1  {  
2    "id": "cmpl-uqkvlQyYK7bGYrRHQ0eXlWi7",  
3    "object": "text_completion",  
4    "created": 1589478378,  
5    "model": "text-davinci-003",  
6    "choices": [  
7      {  
8        "text": "\n\nThis is indeed a test",  
9        "index": 0,  
10       "logprobs": null,  
11       "finish_reason": "length"  
12     }  
13   ],  
14   "usage": {  
15     "prompt_tokens": 5,  
16     "completion_tokens": 7,  
17     "total_tokens": 12  
18   }  
19 }
```

Chat

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

Create chat completion Beta

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

Creates a model response for the given chat conversation.

Request body

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.

messages array **Required**

A list of messages describing the conversation so far.

role string **Required**

The role of the author of this message. One of `system`, `user`, or `assistant`.

content string **Required**

The contents of the message.

name string **Optional**

The name of the author of this message. May contain a-z, A-Z, 0-9, and underscores, with a maximum length of 64 characters.

temperature number **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 **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.

n integer **Optional** Defaults to 1

How many chat completion choices to generate for each input message.

stream boolean **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. See the OpenAI Cookbook for [example code](#).

stop string or array Optional Defaults to null

Up to 4 sequences where the API will stop generating further tokens.

max_tokens integer Optional Defaults to inf

The maximum number of **tokens** to generate in the chat completion.

The total length of input tokens and generated tokens is limited by the model's context length.

presence_penalty number 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.](#)

frequency_penalty number 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.

user string Optional

A unique identifier representing your end-user, which can help OpenAI to monitor and detect abuse.

[Learn more.](#)

Example request

python  Copy

```
1 import os
2 import openai
3 openai.api_key = os.getenv("OPENAI_API_KEY")
4
5 completion = openai.ChatCompletion.create(
6     model="gpt-3.5-turbo",
```

```
7     messages=[
8         {"role": "user", "content": "Hello!"}
9     ]
10 )
11
12 print(completion.choices[0].message)
```

Parameters

[Copy](#)

```
1 {
2     "model": "gpt-3.5-turbo",
3     "messages": [{"role": "user", "content": "Hello!"}]
4 }
```

Response

[Copy](#)

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

Edits

Given a prompt and an instruction, the model will return an edited version of the prompt.

Create edit

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

Creates a new edit for the provided input, instruction, and parameters.

Request body

model string **Required**

ID of the model to use. You can use the `text-davinci-edit-001` or `code-davinci-edit-001` model with this endpoint.

input string Optional Defaults to ""

The input text to use as a starting point for the edit.

instruction string **Required**

The instruction that tells the model how to edit the prompt.

n integer Optional Defaults to 1

How many edits to generate for the input and instruction.

temperature number 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 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.

Example request

text-davinci-edit-001 ▾ python ▾  Copy

```
1 import os
2 import openai
```

```
3 openai.api_key = os.getenv("OPENAI_API_KEY")
4 openai.Edit.create(
5     model="text-davinci-edit-001",
6     input="What day of the wek is it?",
7     instruction="Fix the spelling mistakes"
8 )
```

Parameters

text-davinci-edit-001   Copy

```
1 {
2     "model": "text-davinci-edit-001",
3     "input": "What day of the wek is it?",
4     "instruction": "Fix the spelling mistakes",
5 }
```

Response

 Copy

```
1 {
2     "object": "edit",
3     "created": 1589478378,
4     "choices": [
5         {
6             "text": "What day of the week is it?",
7             "index": 0,
8         }
9     ],
10    "usage": {
11        "prompt_tokens": 25,
12        "completion_tokens": 32,
13        "total_tokens": 57
14    }
15 }
```

Images

Given a prompt and/or an input image, the model will generate a new image.

Related guide: [Image generation](#)

Create image Beta

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.

n integer Optional Defaults to 1

The number of images to generate. Must be between 1 and 10.

size string Optional Defaults to 1024x1024

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

response_format string Optional Defaults to url

The format in which the generated images are returned. Must be one of `url` or `b64_json`.

user string Optional

A unique identifier representing your end-user, which can help OpenAI to monitor and detect abuse.

[Learn more.](#)

Example request

python  Copy

```
1 import os
2 import openai
3 openai.api_key = os.getenv("OPENAI_API_KEY")
4 openai.Image.create(
5     prompt="A cute baby sea otter",
6     n=2,
7     size="1024x1024"
8 )
```

Parameters

 Copy

```
1 {
2   "prompt": "A cute baby sea otter",
3   "n": 2,
4   "size": "1024x1024"
5 }
```

Response

 Copy

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

Create image edit Beta

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



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

Request body

image string 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.

mask string Optional

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

prompt string Required

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

n integer Optional Defaults to 1

The number of images to generate. Must be between 1 and 10.

size string Optional Defaults to 1024x1024

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

response_format string Optional Defaults to url

The format in which the generated images are returned. Must be one of `url` or `b64_json` .

user string Optional

A unique identifier representing your end-user, which can help OpenAI to monitor and detect abuse.

[Learn more.](#)

Example request

python  Copy

```
1 import os
2 import openai
3 openai.api_key = os.getenv("OPENAI_API_KEY")
4 openai.Image.create_edit(
5     image=open("otter.png", "rb"),
6     mask=open("mask.png", "rb"),
7     prompt="A cute baby sea otter wearing a beret",
8     n=2,
9     size="1024x1024"
10 )
```

Response

 Copy

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

Create image variation Beta

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

Creates a variation of a given image.

Request body

image string Required

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

n integer Optional Defaults to 1

The number of images to generate. Must be between 1 and 10.

size string Optional Defaults to 1024x1024

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

response_format string Optional Defaults to url

The format in which the generated images are returned. Must be one of `url` or `b64_json` .

user string Optional

A unique identifier representing your end-user, which can help OpenAI to monitor and detect abuse.

[Learn more.](#)

Example request

python  Copy

```
1 import os
2 import openai
3 openai.api_key = os.getenv("OPENAI_API_KEY")
4 openai.Image.create_variation(
5     image=open("otter.png", "rb"),
6     n=2,
7     size="1024x1024"
8 )
```

Response

 Copy

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

Embeddings

Get a vector representation of a given input that can be easily consumed by machine learning models and algorithms.

Related guide: [Embeddings](#)

Create embeddings

POST <https://api.openai.com/v1/embeddings>

Creates an embedding vector representing the input text.

Request body

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.

input string or array **Required**

Input text to get embeddings for, encoded as a string or array of tokens. To get embeddings for multiple inputs in a single request, pass an array of strings or array of token arrays. Each input must not exceed 8192 tokens in length.

user string Optional

A unique identifier representing your end-user, which can help OpenAI to monitor and detect abuse.

[Learn more.](#)

Example request

python  Copy

```
1 import os
2 import openai
3 openai.api_key = os.getenv("OPENAI_API_KEY")
4 openai.Embedding.create(
5     model="text-embedding-ada-002",
6     input="The food was delicious and the waiter..."
7 )
```

Parameters

 Copy

```
1 {
2     "model": "text-embedding-ada-002",
3     "input": "The food was delicious and the waiter..."
4 }
```

Response

 Copy

```
1 {
2     "object": "list",
3     "data": [
4         {
5             "object": "embedding",
6             "embedding": [
7                 0.0023064255,
8                 -0.009327292,
9                 .... (1536 floats total for ada-002)
10                -0.0028842222,
11            ],
12            "index": 0
13        }
14    ],
15    "model": "text-embedding-ada-002",
16    "usage": {
17        "prompt_tokens": 8,
18        "total_tokens": 8
19    }
20 }
```

Audio

Learn how to turn audio into text.

Related guide: [Speech to text](#)

Create transcription Beta

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

Transcribes audio into the input language.

Request body

file string Required

The audio file to transcribe, in one of these formats: mp3, mp4, mpeg, mpga, m4a, wav, or webm.

model string Required

ID of the model to use. Only `whisper-1` is currently available.

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.

language string Optional

The language of the input audio. Supplying the input language in **ISO-639-1** format will improve accuracy and latency.

Example request

python  Copy

```
1 import os
2 import openai
3 openai.api_key = os.getenv("OPENAI_API_KEY")
4 audio_file = open("audio.mp3", "rb")
5 transcript = openai.Audio.transcribe("whisper-1", audio_file)
```

Parameters

 Copy

```
1 {
2   "file": "audio.mp3",
3   "model": "whisper-1"
4 }
```

Response

 Copy

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

Create translation Beta

POST <https://api.openai.com/v1/audio/translations>

Translates audio into English.

Request body

file string Required

The audio file to translate, in one of these formats: mp3, mp4, mpeg, mpga, m4a, wav, or webm.

model string RequiredID of the model to use. Only `whisper-1` is currently available.**prompt** string Optional

An optional text to guide the model's style or continue a previous audio segment. The **prompt** should be in English.

response_format string Optional Defaults to json

The format of the 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.

Example request

python  Copy

```
1 import os
2 import openai
3 openai.api_key = os.getenv("OPENAI_API_KEY")
4 audio_file = open("german.m4a", "rb")
5 transcript = openai.Audio.translate("whisper-1", audio_file)
```

Parameters

 Copy

```
1 {
2   "file": "german.m4a",
3   "model": "whisper-1"
4 }
```

Response

 Copy

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

Files

Files are used to upload documents that can be used with features like **Fine-tuning**.

List files

GET <https://api.openai.com/v1/files>

Returns a list of files that belong to the user's organization.

Example request

python  Copy

```
1 import os
2 import openai
3 openai.api_key = os.getenv("OPENAI_API_KEY")
4 openai.File.list()
```

Response

 Copy

```
1 {
2   "data": [
3     {
4       "id": "file-ccdDZrC3iZVNiQVeEA6Z66wf",
5       "object": "file",
6       "bytes": 175,
7       "created_at": 1613677385,
8       "filename": "train.jsonl",
9       "purpose": "search"
10    },
11    {
12      "id": "file-XjGxS3KTG0uNmNOK362iJua3",
13      "object": "file",
14      "bytes": 140,
15      "created_at": 1613779121,
16      "filename": "puppy.jsonl",
17      "purpose": "search"
18    }
19  ],
20  "object": "list"
21 }
```

Upload file

POST <https://api.openai.com/v1/files>

Upload a file that contains document(s) to be used across various endpoints/features. Currently, the size of all the files uploaded by one organization can be up to 1 GB. Please contact us if you need to increase the storage limit.

Request body

file string **Required**

Name of the **JSON Lines** file to be uploaded.

If the `purpose` is set to "fine-tune", each line is a JSON record with "prompt" and "completion" fields representing your **training examples**.

purpose string **Required**

The intended purpose of the uploaded documents.

Use "fine-tune" for **Fine-tuning**. This allows us to validate the format of the uploaded file.

Example request

python  Copy

```
1 import os
2 import openai
3 openai.api_key = os.getenv("OPENAI_API_KEY")
4 openai.File.create(
5     file=open("mydata.jsonl", "rb"),
6     purpose='fine-tune'
7 )
```

Response

 Copy

```
1 {
2     "id": "file-XjGxS3KTG0uNmN0K362iJua3",
3     "object": "file",
4     "bytes": 140,
5     "created_at": 1613779121,
6     "filename": "mydata.jsonl",
7     "purpose": "fine-tune"
8 }
```

Delete file

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

Delete a file.

Path parameters

`file_id` string **Required**

The ID of the file to use for this request

Example request

python  Copy

```
1 import os
2 import openai
3 openai.api_key = os.getenv("OPENAI_API_KEY")
4 openai.File.delete("file-XjGxS3KTG0uNmNOK362iJua3")
```

Response

 Copy

```
1 {
2   "id": "file-XjGxS3KTG0uNmNOK362iJua3",
3   "object": "file",
4   "deleted": true
5 }
```

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

Example request

python  Copy

```
1 import os
2 import openai
3 openai.api_key = os.getenv("OPENAI_API_KEY")
4 openai.File.retrieve("file-XjGxS3KTG0uNmNOK362iJua3")
```

Response

 Copy

```
1 {
2   "id": "file-XjGxS3KTG0uNmNOK362iJua3",
3   "object": "file",
4   "bytes": 140,
5   "created_at": 1613779657,
6   "filename": "mydata.jsonl",
7   "purpose": "fine-tune"
8 }
```

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

Example request

python  Copy

```
1 import os
2 import openai
3 openai.api_key = os.getenv("OPENAI_API_KEY")
4 content = openai.File.download("file-XjGxS3KTG0uNmNOK362iJua3")
```

Fine-tunes

Manage fine-tuning jobs to tailor a model to your specific training data.

Related guide: [Fine-tune models](#)

Create fine-tune

POST <https://api.openai.com/v1/fine-tunes>

Creates a job that fine-tunes a specified 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

`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, where each training example is a JSON object with the keys "prompt" and "completion". Additionally, you must upload your file with the purpose `fine-tune`.

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

`validation_file` string **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](#). Your train and validation data should be mutually exclusive.

Your dataset must be formatted as a JSONL file, where each validation example is a JSON object with the keys "prompt" and "completion". Additionally, you must upload your file with the purpose `fine-tune`.

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

model string Optional Defaults to curie

The name of the base model to fine-tune. You can select one of "ada", "babbage", "curie", "davinci", or a fine-tuned model created after 2022-04-21. To learn more about these models, see the [Models](#) documentation.

n_epochs integer Optional Defaults to 4

The number of epochs to train the model for. An epoch refers to one full cycle through the training dataset.

batch_size integer Optional Defaults to null

The batch size to use for training. The batch size is the number of training examples used to train a single forward and backward pass.

By default, the batch size will be dynamically configured to be ~0.2% of the number of examples in the training set, capped at 256 - in general, we've found that larger batch sizes tend to work better for larger datasets.

learning_rate_multiplier number Optional Defaults to null

The learning rate multiplier to use for training. The fine-tuning learning rate is the original learning rate used for pretraining multiplied by this value.

By default, the learning rate multiplier is the 0.05, 0.1, or 0.2 depending on final `batch_size` (larger learning rates tend to perform better with larger batch sizes). We recommend experimenting with values in the range 0.02 to 0.2 to see what produces the best results.

prompt_loss_weight number Optional Defaults to 0.01

The weight to use for loss on the prompt tokens. This controls how much the model tries to learn to generate the prompt (as compared to the completion which always has a weight of 1.0), and can add a stabilizing effect to training when completions are short.

If prompts are extremely long (relative to completions), it may make sense to reduce this weight so as to avoid over-prioritizing learning the prompt.

compute_classification_metrics boolean Optional Defaults to false

If set, we calculate classification-specific metrics such as accuracy and F-1 score using the validation set at the end of every epoch. These metrics can be viewed in the [results file](#).

In order to compute classification metrics, you must provide a `validation_file`. Additionally, you must specify `classification_n_classes` for multiclass classification or `classification_positive_class` for binary classification.

classification_n_classes integer Optional Defaults to null

The number of classes in a classification task.

This parameter is required for multiclass classification.

classification_positive_class string Optional Defaults to null

The positive class in binary classification.

This parameter is needed to generate precision, recall, and F1 metrics when doing binary classification.

classification_betas array Optional Defaults to null

If this is provided, we calculate F-beta scores at the specified beta values. The F-beta score is a generalization of F-1 score. This is only used for binary classification.

With a beta of 1 (i.e. the F-1 score), precision and recall are given the same weight. A larger beta score puts more weight on recall and less on precision. A smaller beta score puts more weight on precision and less on recall.

suffix string Optional Defaults to null

A string of up to 40 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 `ada:ft-your-org:custom-model-name-2022-02-15-04-21-04`.

Example request

python  Copy

```
1 import os
2 import openai
3 openai.api_key = os.getenv("OPENAI_API_KEY")
4 openai.FineTune.create(training_file="file-XGinujblHPwGLSztz8cPS8XY")
```

Response

 Copy

```
1 {
2   "id": "ft-AF1WoRqd3aJAHsqc9NY7iL8F",
3   "object": "fine-tune",
4   "model": "curie",
5   "created_at": 1614807352,
6   "events": [
7     {
8       "object": "fine-tune-event",
9       "created_at": 1614807352,
```

```
10     "level": "info",
11     "message": "Job enqueued. Waiting for jobs ahead to complete. Queue number: 6",
12 }
13 ],
14 "fine_tuned_model": null,
15 "hyperparams": {
16     "batch_size": 4,
17     "learning_rate_multiplier": 0.1,
18     "n_epochs": 4,
19     "prompt_loss_weight": 0.1,
20 },
21 "organization_id": "org-...",
22 "result_files": [],
23 "status": "pending",
24 "validation_files": [],
25 "training_files": [
26     {
27         "id": "file-XGinujblHPwGLSztz8cPS8XY",
28         "object": "file",
29         "bytes": 1547276,
30         "created_at": 1610062281,
31         "filename": "my-data-train.jsonl",
32         "purpose": "fine-tune-train"
33     }
34 ],
35 "updated_at": 1614807352,
36 }
```

List fine-tunes

GET <https://api.openai.com/v1/fine-tunes>

Example request

python  Copy

```
1 import os
2 import openai
3 openai.api_key = os.getenv("OPENAI_API_KEY")
4 openai.FineTune.list()
```



Response

```
1  {
2    "object": "list",
3    "data": [
4      {
5        "id": "ft-AF1WoRqd3aJAHsqc9NY7iL8F",
6        "object": "fine-tune",
7        "model": "curie",
8        "created_at": 1614807352,
9        "fine_tuned_model": null,
10       "hyperparams": { ... },
11       "organization_id": "org-...",
12       "result_files": [],
13       "status": "pending",
14       "validation_files": [],
15       "training_files": [ { ... } ],
16       "updated_at": 1614807352,
17     },
18     { ... },
19     { ... }
20   ]
21 }
```

Retrieve fine-tune

GET https://api.openai.com/v1/fine-tunes/{fine_tune_id}

Gets info about the fine-tune job.

[Learn more about Fine-tuning](#)

Path parameters

`fine_tune_id` string **Required**

The ID of the fine-tune job

Example request

python Copy


```
1 import os
2 import openai
3 openai.api_key = os.getenv("OPENAI_API_KEY")
4 openai.FineTune.retrieve(id="ft-AF1WoRqd3aJAHsqc9NY7iL8F")
```

Response

 Copy

```
1  {
2    "id": "ft-AF1WoRqd3aJAHsqc9NY7iL8F",
3    "object": "fine-tune",
4    "model": "curie",
5    "created_at": 1614807352,
6    "events": [
7      {
8        "object": "fine-tune-event",
9        "created_at": 1614807352,
10       "level": "info",
11       "message": "Job enqueued. Waiting for jobs ahead to complete. Queue number: 6",
12     },
13     {
14       "object": "fine-tune-event",
15       "created_at": 1614807356,
16       "level": "info",
17       "message": "Job started."
18     },
19     {
20       "object": "fine-tune-event",
21       "created_at": 1614807861,
22       "level": "info",
23       "message": "Uploaded snapshot: curie:ft-acmeco-2021-03-03-21-44-20."
24     },
25     {
26       "object": "fine-tune-event",
27       "created_at": 1614807864,
28       "level": "info",
29       "message": "Uploaded result files: file-QQm6ZpqdNwAaVC3aSz5sWwLT."
30     },
31     {
32       "object": "fine-tune-event",
33       "created_at": 1614807864,
34       "level": "info",
35       "message": "Job succeeded."
36     }
37   ],
38   "fine_tuned_model": "curie:ft-acmeco-2021-03-03-21-44-20",
39   "hyperparams": {
40     "batch_size": 4,
```

```
41     "learning_rate_multiplier": 0.1,
42     "n_epochs": 4,
43     "prompt_loss_weight": 0.1,
44 },
45 "organization_id": "org-...",
46 "result_files": [
47   {
48     "id": "file-QQm6ZpqdNwAaVC3aSz5sWwLT",
49     "object": "file",
50     "bytes": 81509,
51     "created_at": 1614807863,
52     "filename": "compiled_results.csv",
53     "purpose": "fine-tune-results"
54   }
55 ],
56 "status": "succeeded",
57 "validation_files": [],
58 "training_files": [
59   {
60     "id": "file-XGinujblHPwGLSztz8cPS8XY",
61     "object": "file",
62     "bytes": 1547276,
63     "created_at": 1610062281,
64     "filename": "my-data-train.jsonl",
65     "purpose": "fine-tune-train"
66   }
67 ],
68 "updated_at": 1614807865,
69 }
```

Cancel fine-tune

POST https://api.openai.com/v1/fine-tunes/{fine_tune_id}/cancel

Immediately cancel a fine-tune job.

Path parameters

`fine_tune_id` string **Required**

The ID of the fine-tune job to cancel

Example request

python  Copy

```
1 import os
2 import openai
3 openai.api_key = os.getenv("OPENAI_API_KEY")
4 openai.FineTune.cancel(id="ft-AF1WoRqd3aJAHsqc9NY7iL8F")
```

Response

 Copy

```
1 {
2   "id": "ft-xhrpBbvVUzYGo8oU01FY4nI7",
3   "object": "fine-tune",
4   "model": "curie",
5   "created_at": 1614807770,
6   "events": [ { ... } ],
7   "fine_tuned_model": null,
8   "hyperparams": { ... },
9   "organization_id": "org-...",
10  "result_files": [],
11  "status": "cancelled",
12  "validation_files": [],
13  "training_files": [
14    {
15      "id": "file-XGinujblHPwGLSztz8cPS8XY",
16      "object": "file",
17      "bytes": 1547276,
18      "created_at": 1610062281,
19      "filename": "my-data-train.jsonl",
20      "purpose": "fine-tune-train"
21    }
22  ],
23  "updated_at": 1614807789,
24 }
```

List fine-tune events

GET https://api.openai.com/v1/fine-tunes/{fine_tune_id}/events

Get fine-grained status updates for a fine-tune job.

Path parameters

`fine_tune_id` string **Required**

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

Query parameters

`stream` boolean Optional Defaults to false

Whether to stream events for the fine-tune job. If set to true, events will be sent as data-only **server-sent events** as they become available. The stream will terminate with a `data: [DONE]` message when the job is finished (succeeded, cancelled, or failed).

If set to false, only events generated so far will be returned.

Example request

python  Copy

```
1 import os
2 import openai
3 openai.api_key = os.getenv("OPENAI_API_KEY")
4 openai.FineTune.list_events(id="ft-AF1WoRqd3aJAHsqc9NY7iL8F")
```

Response

 Copy

```
1 {
2   "object": "list",
3   "data": [
4     {
5       "object": "fine-tune-event",
6       "created_at": 1614807352,
7       "level": "info",
8       "message": "Job queued. Waiting for jobs ahead to complete. Queue number: 6",
9     },
10    {
11      "object": "fine-tune-event",
12      "created_at": 1614807356,
13      "level": "info",
14      "message": "Job started."
15    },
16    {
17      "object": "fine-tune-event",
18      "created_at": 1614807861,
19      "level": "info",
20      "message": "Uploaded snapshot: curie:ft-acmeco-2021-03-03-21-44-20."
21    },
22    {
```

```
23     "object": "fine-tune-event",
24     "created_at": 1614807864,
25     "level": "info",
26     "message": "Uploaded result files: file-QQm6ZpqdNwAaVC3aSz5sWwLT."
27 },
28 {
29     "object": "fine-tune-event",
30     "created_at": 1614807864,
31     "level": "info",
32     "message": "Job succeeded."
33 }
34 ]
35 }
```

Delete fine-tune model

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

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

Path parameters

`model` string **Required**

The model to delete

Example request

python  Copy

```
1 import os
2 import openai
3 openai.api_key = os.getenv("OPENAI_API_KEY")
4 openai.Model.delete("curie:ft-acmeco-2021-03-03-21-44-20")
```

Response

 Copy

```
1 {
2     "id": "curie:ft-acmeco-2021-03-03-21-44-20",
3     "object": "model",
4
5 }
```

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

Moderations

Given a input text, outputs if the model classifies it as violating OpenAI's content policy.

Related guide: [Moderations](#)

Create moderation

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

Classifies if text violates OpenAI's Content Policy

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`.

Example request

python  Copy

```
1 import os  
2 import openai  
3 openai.api_key = os.getenv("OPENAI_API_KEY")  
4 openai.Moderation.create(  

```

```
5  input="I want to kill them.",
6  )
```

Parameters

[Copy](#)

```
1  {
2    "input": "I want to kill them."
3  }
```

Response

[Copy](#)

```
1  {
2    "id": "modr-5MWoL0",
3    "model": "text-moderation-001",
4    "results": [
5      {
6        "categories": {
7          "hate": false,
8          "hate/threatening": true,
9          "self-harm": false,
10         "sexual": false,
11         "sexual/minors": false,
12         "violence": true,
13         "violence/graphic": false
14       },
15       "category_scores": {
16         "hate": 0.22714105248451233,
17         "hate/threatening": 0.4132447838783264,
18         "self-harm": 0.005232391878962517,
19         "sexual": 0.01407341007143259,
20         "sexual/minors": 0.0038522258400917053,
21         "violence": 0.9223177433013916,
22         "violence/graphic": 0.036865197122097015
23       },
24       "flagged": true
25     }
26   ]
27 }
```

Engines

 The Engines endpoints are deprecated.

Please use their replacement, [Models](#), instead. [Learn more](#).

These endpoints describe and provide access to the various engines available in the API.

List engines Deprecated

GET <https://api.openai.com/v1/engines>

Lists the currently available (non-finetuned) models, and provides basic information about each one such as the owner and availability.

Example request

python  Copy

```
1 import os
2 import openai
3 openai.api_key = os.getenv("OPENAI_API_KEY")
4 openai.Engine.list()
```

Response

 Copy

```
1  {
2    "data": [
3      {
4        "id": "engine-id-0",
5        "object": "engine",
6        "owner": "organization-owner",
7        "ready": true
8      },
9      {
10       "id": "engine-id-2",
11       "object": "engine",
12       "owner": "organization-owner",
13       "ready": true
14     },
15     {
16       "id": "engine-id-3",
17       "object": "engine",
18       "owner": "openai",
19       "ready": false
20     },
21   ],
```



```
22     "object": "list"
23 }
```

Retrieve engine Deprecated

GET https://api.openai.com/v1/engines/{engine_id}

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

Path parameters

engine_id string Required

The ID of the engine to use for this request

Example request

text-davinci-003 ▾ python ▾  Copy

```
1 import os
2 import openai
3 openai.api_key = os.getenv("OPENAI_API_KEY")
4 openai.Engine.retrieve("text-davinci-003")
```

Response

text-davinci-003 ▾  Copy

```
1 {
2   "id": "text-davinci-003",
3   "object": "engine",
4   "owner": "openai",
5   "ready": true
6 }
```

Parameter details

Frequency and presence penalties