

## Introduction

This API reference describes the RESTful, streaming, and realtime APIs you can use to interact with the OpenAI platform. REST APIs are usable via HTTP in any environment that supports HTTP requests. Language-specific SDKs are listed [on the libraries page](#).

---

## Authentication

The OpenAI API uses API keys for authentication. Create, manage, and learn more about API keys in your [organization settings](#).

**Remember that your API key is a secret!** Do not share it with others or expose it in any client-side code (browsers, apps). API keys should be securely loaded from an environment variable or key management service on the server.

API keys should be provided via [HTTP Bearer authentication](#).

```
Authorization: Bearer OPENAI_API_KEY
```



If you belong to multiple organizations or access projects through a legacy user API key, pass a header to specify which organization and project to use for an API request:

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



Usage from these API requests counts as usage for the specified organization and project. Organization IDs can be found on your [organization settings](#) page. Project IDs can be found on your [general settings](#) page by selecting the specific project.

---

# Debugging requests

In addition to [error codes](#) returned from API responses, you can inspect HTTP response headers containing the unique ID of a particular API request or information about rate limiting applied to your requests. Below is an incomplete list of HTTP headers returned with API responses:

## API meta information

- `openai-organization` : The [organization](#) associated with the request
- `openai-processing-ms` : Time taken processing your API request
- `openai-version` : REST API version used for this request (currently `2020-10-01` )
- `x-request-id` : Unique identifier for this API request (used in troubleshooting)

## Rate limiting information

- `x-ratelimit-limit-requests`
- `x-ratelimit-limit-tokens`
- `x-ratelimit-remaining-requests`
- `x-ratelimit-remaining-tokens`
- `x-ratelimit-reset-requests`
- `x-ratelimit-reset-tokens`

OpenAI recommends logging request IDs in production deployments for more efficient troubleshooting with our [support team](#), should the need arise. Our [official SDKs](#) provide a property on top-level response objects containing the value of the `x-request-id` header.

---

# Backward compatibility

OpenAI is committed to providing stability to API users by avoiding breaking changes in major API versions whenever reasonably possible. This includes:

The REST API (currently `v1` )

Our first-party [SDKs](#) (released SDKs adhere to [semantic versioning](#))

[Model](#) families (like `gpt-4o` or `o4-mini` )

**Model prompting behavior between snapshots is subject to change.** Model outputs are by their nature variable, so expect changes in prompting and model behavior between snapshots. For example, if you moved from `gpt-4o-2024-05-13` to `gpt-4o-2024-08-06` , the same `system` or `user` messages could function differently between versions. The best way to ensure consistent prompting behavior and model output is to use pinned model versions, and to implement [evals](#) for your applications.

### Backwards-compatible API changes:

- Adding new resources (URLs) to the REST API and SDKs

- Adding new optional API parameters

- Adding new properties to JSON response objects or event data

- Changing the order of properties in a JSON response object

- Changing the length or format of opaque strings, like resource identifiers and UUIDs

- Adding new event types (in either streaming or the Realtime API)

See the [changelog](#) for a list of backwards-compatible changes and rare breaking changes.

---

## Responses

OpenAI's most advanced interface for generating model responses. Supports text and image inputs, and text outputs. Create stateful interactions with the model, using the output of previous responses as input. Extend the model's capabilities with built-in tools for file search, web search, computer use, and more. Allow the model access to external systems and data using function calling.

Related guides:

- [Quickstart](#)

Emitted when the transcription is complete. Contains the complete transcription text. Only emitted when you [create a transcription](#) with the `Stream` parameter set to `true`.

---

### **logprobs** array

The log probabilities of the individual tokens in the transcription. Only included if you [create a transcription](#) with the `include[]` parameter set to `logprobs`.

▼ Show properties

---

### **text** string

The text that was transcribed.

---

### **type** string

The type of the event. Always `transcript.text.done`.

---

OBJECT Stream Event (transcript.text.done)



```
1 {  
2   "type": "transcript.text.done",  
3   "text": "I see skies of blue and clouds of w  
4 }
```

## 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. [Learn more](#).

## Request body

**prompt** string Required

A text description of the desired image(s). The maximum length is 32000 characters for

`gpt-image-1`, 1000 characters for `dall-e-2` and 4000 characters for `dall-e-3`.

**background** string or null

Optional Defaults to auto

Allows to set transparency for the background of the generated image(s). This parameter is only supported for `gpt-image-1`. Must be one of `transparent`, `opaque` or `auto` (default value). When `auto` is used, the model will automatically determine the best background for the image.

If `transparent`, the output format needs to support transparency, so it should be set to either `png` (default value) or `webp`.

**model** string Optional

Defaults to dall-e-2

The model to use for image

Example request

curl 

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

Response 

```
1 {
2   "created": 1713833628,
3   "data": [
4     {
5       "b64_json": "...
6     }
7   ],
8   "usage": {
9     "total_tokens": 100,
10    "input_tokens": 50,
11    "output_tokens": 50,
12    "input_tokens_details": {
13      "text_tokens": 10,
14      "image_tokens": 40
15    }
16  }
17 }
```

generation. One of `dall-e-2`, `dall-e-3`, or `gpt-image-1`. Defaults to `dall-e-2` unless a parameter specific to `gpt-image-1` is used.

---

**moderation** string or null

Optional Defaults to auto

Control the content-moderation level for images generated by

`gpt-image-1`. Must be either `low` for less restrictive filtering or `auto` (default value).

---

**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.

---

**output\_compression**

integer or null Optional

Defaults to 100

The compression level (0-100%) for the generated images. This parameter is only supported for `gpt-image-1` with the `webp` or `jpeg` output formats, and defaults to 100.

---

**output\_format** string or null

Optional Defaults to png

The format in which the generated images are returned.

This parameter is only supported for `gpt-image-1`. Must be one of `png`, `jpeg`, or `webp`.

---

**quality** string or null Optional

Defaults to auto

The quality of the image that will be generated.

`auto` (default value) will automatically select the best quality for the given model.

`high`, `medium` and `low` are supported for `gpt-image-1`.

`hd` and `standard` are supported for `dall-e-3`.

`standard` is the only option for `dall-e-2`.

---

**response\_format** string or null

Optional Defaults to url

The format in which generated images with `dall-e-2` and `dall-e-3` are returned. Must be one of `url` or `b64_json`.

URLs are only valid for 60 minutes after the image has been generated. This parameter isn't supported for

`gpt-image-1` which will always return base64-encoded images.

---

**size** string or null Optional

Defaults to auto

The size of the generated images. Must be one of

`1024x1024`, `1536x1024` (landscape), `1024x1536` (portrait), or `auto` (default value) for `gpt-image-1`, one of

`256x256` , `512x512` , or  
`1024x1024` for `dall-e-2` ,  
and one of `1024x1024` ,  
`1792x1024` , or `1024x1792`  
for `dall-e-3` .

---

**style** string or null Optional

Defaults to `vivid`

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

---

**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.

---

# Create image edit

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

Example request

curl 

```
1 curl -s -D >(grep -i x-request-id >&2) \  
2 -o >(jq -r '.data[0].b64_json' | base64 --d
```



Creates an edited or extended image given one or more source images and a prompt. This endpoint only supports `gpt-image-1` and `dall-e-2`.

## Request body

**image** string or array **Required**  
The image(s) to edit. Must be a supported image file or an array of images. For `gpt-image-1`, each image should be a `png`, `webp`, or `jpg` file less than 25MB. For `dall-e-2`, you can only provide one image, and it should be a square `png` file less than 4MB.

**prompt** string **Required**  
A text description of the desired image(s). The maximum length is 1000 characters for `dall-e-2`, and 32000 characters for `gpt-image-1`.

**mask** file **Optional**  
An additional image whose fully transparent areas (e.g. where alpha is zero) indicate where `image` should be edited. If there are multiple images provided, the mask will be applied on the first image. Must be a valid PNG file, less than 4MB, and have the same dimensions as `image`.

```
3 -X POST "https://api.openai.com/v1/images/e
4 -H "Authorization: Bearer $OPENAI_API_KEY"
5 -F "model=gpt-image-1" \
6 -F "image[]=@body-lotion.png" \
7 -F "image[]=@bath-bomb.png" \
8 -F "image[]=@incense-kit.png" \
9 -F "image[]=@soap.png" \
10 -F 'prompt=Create a lovely gift basket with
```

## Response



```
1 {
2   "created": 1713833628,
3   "data": [
4     {
5       "b64_json": "...",
6     }
7   ],
8   "usage": {
9     "total_tokens": 100,
10    "input_tokens": 50,
11    "output_tokens": 50,
12    "input_tokens_details": {
13      "text_tokens": 10,
14      "image_tokens": 40
15    }
16  }
17 }
```

---

**model** string Optional

Defaults to dall-e-2

The model to use for image generation. Only `dall-e-2` and `gpt-image-1` are supported. Defaults to `dall-e-2` unless a parameter specific to `gpt-image-1` is used.

---

**n** integer or null Optional

Defaults to 1

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

---

**quality** string or null Optional

Defaults to auto

The quality of the image that will be generated. `high`, `medium` and `low` are only supported for `gpt-image-1`. `dall-e-2` only supports `standard` quality. Defaults to `auto`.

---

**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. This parameter is only supported for `dall-e-2`, as `gpt-image-1` will always return base64-encoded images.

---

**size** string or null Optional

Defaults to 1024×1024

The size of the generated images. Must be one of `1024x1024` , `1536x1024` (landscape), `1024x1536` (portrait), or `auto` (default value) for `gpt-image-1` , and one of `256x256` , `512x512` , or `1024x1024` for `dall-e-2` .

---

**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.

---

# Create image variation

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

Creates a variation of a given image. This endpoint only supports `dall-e-2` .

## Request body

---

**image** file Required

The image to use as the basis for

Example request

curl ↕ 

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

Response



```
1 {
2   "created": 1589478378,
3   "data": [
4     {
```

the variation(s). Must be a valid PNG file, less than 4MB, and square.

---

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

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

---

**n** integer or null Optional  
Defaults to 1

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

---

**response\_format** string or null  
Optional Defaults to url

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

---

**size** string or null Optional  
Defaults to 1024×1024

The size of the generated images. Must be one of `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](#).

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

## Returns

Returns a list of [image](#) objects.

## The image generation response

The response from the image generation endpoint.

**created** integer

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

**data** array

The list of generated images.

▼ Show properties

**usage** object

For `gpt-image-1` only, the token usage information for the image generation.

▼ Show properties

OBJECT The image generation response



```
1  {
2    "created": 1713833628,
3    "data": [
4      {
5        "b64_json": "...",
6      }
7    ],
8    "usage": {
9      "total_tokens": 100,
10     "input_tokens": 50,
11     "output_tokens": 50,
12     "input_tokens_details": {
13       "text_tokens": 10,
14       "image_tokens": 40
15     }
16   }
17 }
```

## Embeddings

Get a vector representation of a given input that can be easily consumed by machine learning models and algorithms. Related guide: [Embeddings](#)