OpenAI Platform

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:

```
curl https://api.openai.com/v1/models \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "OpenAI-Organization: org-4VsgZaw4tQ5kPK1gFZlB3cI7" \
  -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
- Text inputs and outputs
- Image inputs
- Structured Outputs
- Function calling
- Conversation state
- Extend the models with tools

Create a model response

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

Text input | Image input | Web search | File search | S

Creates a model response. Provide text or image inputs to generate text or JSON outputs. Have the model call your own custom code or use built-in tools like web search or file search to use your own data as input for the model's response.

Request body

input string or array Required

Text, image, or file inputs to the model, used to generate a response.

Learn more:

- Text inputs and outputs
- Image inputs
- File inputs
- Conversation state
- Function calling

Show possible types

model string Required

Model ID used to generate the response, like `gpt-4o` or `o3`. OpenAI offers a wide range of models with different capabilities, performance characteristics, and price points. Refer to the model guide to browse and compare available models.

include array or null Optional

Specify additional output data to include in the model response. Currently supported values are:

- `file_search_call.results` : Include the search results of the file search tool call.
- `message.input_image.image_url` : Include image urls from the input message.
- `computer_call_output.output.image_url` : Include image urls from the computer call output.

instructions string or null Optional

Inserts a system (or developer) message as the first item in the model's context.

When using along with `previous_response_id` , the instructions from a previous response will not be carried over to the next response. This makes it simple to swap out system (or developer) messages in new responses.

max_output_tokens integer or null Optional

An upper bound for the number of tokens that can be generated for a response, including visible output tokens and reasoning tokens.

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, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

parallel_tool_calls boolean or null Optional Defaults to true

Whether to allow the model to run tool calls in parallel.

previous_response_id string or null Optional

The unique ID of the previous response to the model. Use this to create multi-turn conversations. Learn more about conversation state.

reasoning object or null Optional

o-series models only

Configuration options for reasoning models.

Example request javascript

```javascript
import OpenAI from "openai";

const openai = new OpenAI();

const response = await openai.responses.create(
    model: "gpt-4.1",
    input: "Tell me a three sentence bedtime st
});

console.log(response);
```

Response


```
{
  "id": "resp_67ccd2bed1ec8190b14f964abc054267(
  "object": "response",
  "created_at": 1741476542,
  "status": "completed",
  "error": null,
  "incomplete_details": null,
  "instructions": null,
  "max_output_tokens": null,
  "model": "gpt-4.1-2025-04-14",
  "output": [
    {
      "type": "message",
      "id": "msg_67ccd2bf17f0819081ff3bb2cf6508
      "status": "completed",
      "role": "assistant",
      "content": [
        {
          "type": "output_text",
          "text": "In a peaceful grove beneath
          "annotations": []
        }
      ]
    }
  ],
  "parallel_tool_calls": true,
  "previous_response_id": null,
  "reasoning": {
    "effort": null,
    "summary": null
  },
  "store": true,
  "temperature": 1.0,
  "text": {
    "format": {
      "type": "text"
    }
  },
  "tool_choice": "auto",
  "tools": [],
  "top_p": 1.0,
  "truncation": "disabled",
  "usage": {
    "input_tokens": 36,
    "input_tokens_details": {
      "cached_tokens": 0
    },
    "output_tokens": 87,
    "output_tokens_details": {
      "reasoning_tokens": 0
    },
    "total_tokens": 123
  },
  "user": null,
```

Show properties

service_tier string or null Optional Defaults to auto

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

- If set to 'auto', and the Project is Scale tier enabled, the system will utilize scale tier credits until they are exhausted.
- If set to 'auto', and the Project is not Scale tier enabled, the request will be processed using the default service tier with a lower uptime SLA and no latency guarentee.
- If set to 'default', the request will be processed using the default service tier with a lower uptime SLA and no latency guarentee.
- If set to 'flex', the request will be processed with the Flex Processing service tier. Learn more.
- When not set, the default behavior is 'auto'.

When this parameter is set, the response body will include the `service_tier` utilized.

store boolean or null Optional Defaults to true

Whether to store the generated model response for later retrieval via API.

stream boolean or null Optional Defaults to false

If set to true, the model response data will be streamed to the client as it is generated using server-sent events. See the Streaming section below for more information.

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.

text object Optional

Configuration options for a text response from the model. Can be plain text or structured JSON data. Learn more:

- Text inputs and outputs
- Structured Outputs

Show properties

tool_choice string or object Optional

How the model should select which tool (or tools) to use when generating a response. See the `tools` parameter to see how to specify which tools the model can call.

Show possible types

tools array Optional

An array of tools the model may call while generating a response. You can specify which tool to use by setting the `tool_choice` parameter.

The two categories of tools you can provide the model are:

- **Built-in tools**: Tools that are provided by OpenAI that extend the model's capabilities, like web search or file search. Learn more about built-in tools.
- **Function calls (custom tools)**: Functions that are defined by you, enabling the model to call your own code. Learn more about function calling.

Show possible types

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.

truncation string or null Optional Defaults to disabled


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

The truncation strategy to use for the model response.

- `auto` : If the context of this response and previous ones exceeds the model's context window size, the model will truncate the response to fit the context window by dropping input items in the middle of the conversation.
- `disabled` (default): If a model response will exceed the context window size for a model, the request will fail with a 400 error.

`user` string Optional

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

Returns

Returns a Response object.

Get a model response

`GET https://api.openai.com/v1/responses/{response_id}`

Retrieves a model response with the given ID.

Path parameters

`response_id` string Required

The ID of the response to retrieve.

Query parameters

`include` array Optional

Additional fields to include in the response. See the `include` parameter for Response creation above for more information.

Returns

The Response object matching the specified ID.

Example request javascript

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

const response = await client.responses.retriev
console.log(response);
```

Response


```
{
  "id": "resp_67cb71b351908190a308f3859487620d
  "object": "response",
  "created_at": 1741386163,
  "status": "completed",
  "error": null,
  "incomplete_details": null,
  "instructions": null,
  "max_output_tokens": null,
  "model": "gpt-4o-2024-08-06",
  "output": [
    {
      "type": "message",
      "id": "msg_67cb71b3c2b0819084d481baaaf148
      "status": "completed",
      "role": "assistant",
      "content": [
        {
          "type": "output_text",
          "text": "Silent circuits hum,  \nThou
          "annotations": []
        }
      ]
    }
  ],
  "parallel_tool_calls": true,
  "previous_response_id": null,
  "reasoning": {
    "effort": null,
    "summary": null
  },
  "store": true,
  "temperature": 1.0,
  "text": {
    "format": {
```

```
    "type": "text"
  }
},
"tool_choice": "auto",
"tools": [],
"top_p": 1.0,
"truncation": "disabled",
"usage": {
  "input_tokens": 32,
  "input_tokens_details": {
    "cached_tokens": 0
  },
  "output_tokens": 18,
  "output_tokens_details": {
    "reasoning_tokens": 0
  },
  "total_tokens": 50
},
"user": null,
"metadata": {}
}
```

Delete a model response

`DELETE https://api.openai.com/v1/responses/{response_id}`

Deletes a model response with the given ID.

Path parameters

response_id string Required
The ID of the response to delete.

Returns

A success message.

Example request — javascript

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

const response = await client.responses.del("res
console.log(response);
```

Response

```
{
  "id": "resp_6786a1bec27481909a17d673315b29f6"
  "object": "response",
  "deleted": true
}
```

List input items

`GET https://api.openai.com/v1/responses/{response_id}/input_items`

Returns a list of input items for a given response.

Path parameters

response_id string Required
The ID of the response to retrieve input items for.

Query parameters

after string Optional
An item ID to list items after, used in pagination.

before string Optional
An item ID to list items before, used in pagination.

Example request — javascript

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

const response = await client.responses.inputIt
console.log(response.data);
```

Response

```
{
  "object": "list",
  "data": [
    {
      "id": "msg_abc123",
      "type": "message",
      "role": "user",
      "content": [
        {
          "type": "input_text",
```

include array Optional

Additional fields to include in the response. See the `include` parameter for Response creation above for more information.

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

The order to return the input items in. Default is `asc`.

- `asc`: Return the input items in ascending order.
- `desc`: Return the input items in descending order.

```
          "text": "Tell me a three sentence bed
        }
      ]
    }
  ],
  "first_id": "msg_abc123",
  "last_id": "msg_abc123",
  "has_more": false
}
```

Returns

A list of input item objects.

The response object

created_at number

Unix timestamp (in seconds) of when this Response was created.

error object or null

An error object returned when the model fails to generate a Response.

Show properties

id string

Unique identifier for this Response.

incomplete_details object or null

Details about why the response is incomplete.

Show properties

instructions string or null

Inserts a system (or developer) message as the first item in the model's context.

When using along with `previous_response_id`, the instructions from a previous response will not be carried over to the next response. This makes it simple to swap out system (or developer) messages in new responses.

max_output_tokens integer or null

An upper bound for the number of tokens that can be generated for a response, including visible output tokens and reasoning tokens.

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, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

model string

Model ID used to generate the response, like `gpt-4o` or `o3`. OpenAI offers a wide range of models with different capabilities, performance characteristics, and price points. Refer to the model guide to browse and compare available models.

OBJECT The response object

```
{
  "id": "resp_67ccd3a9da748190baa7f1570fe91ac6
  "object": "response",
  "created_at": 1741476777,
  "status": "completed",
  "error": null,
  "incomplete_details": null,
  "instructions": null,
  "max_output_tokens": null,
  "model": "gpt-4o-2024-08-06",
  "output": [
    {
      "type": "message",
      "id": "msg_67ccd3acc8d48190a77525dc6de64
      "status": "completed",
      "role": "assistant",
      "content": [
        {
          "type": "output_text",
          "text": "The image depicts a scenic
          "annotations": []
        }
      ]
    }
  ],
  "parallel_tool_calls": true,
  "previous_response_id": null,
  "reasoning": {
    "effort": null,
    "summary": null
  },
  "store": true,
  "temperature": 1.0,
  "text": {
    "format": {
      "type": "text"
    }
  },
  "tool_choice": "auto",
  "tools": [],
```

`object` string

The object type of this resource - always set to `response`.

`output` array

An array of content items generated by the model.

- The length and order of items in the `output` array is dependent on the model's response.
- Rather than accessing the first item in the `output` array and assuming it's an `assistant` message with the content generated by the model, you might consider using the `output_text` property where supported in SDKs.

Show possible types

`output_text` string or null SDK Only

SDK-only convenience property that contains the aggregated text output from all `output_text` items in the `output` array, if any are present. Supported in the Python and JavaScript SDKs.

`parallel_tool_calls` boolean

Whether to allow the model to run tool calls in parallel.

`previous_response_id` string or null

The unique ID of the previous response to the model. Use this to create multi-turn conversations. Learn more about conversation state.

`reasoning` object or null

o-series models only

Configuration options for reasoning models.

Show properties

`service_tier` string or null

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

- If set to 'auto', and the Project is Scale tier enabled, the system will utilize scale tier credits until they are exhausted.
- If set to 'auto', and the Project is not Scale tier enabled, the request will be processed using the default service tier with a lower uptime SLA and no latency guarentee.
- If set to 'default', the request will be processed using the default service tier with a lower uptime SLA and no latency guarentee.
- If set to 'flex', the request will be processed with the Flex Processing service tier. Learn more.
- When not set, the default behavior is 'auto'.

When this parameter is set, the response body will include the `service_tier` utilized.

`status` string

The status of the response generation. One of `completed`, `failed`, `in_progress`, or `incomplete`.

`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. We generally recommend altering this or `top_p` but not both.

`text` object

Configuration options for a text response from the model. Can be plain text or structured JSON data. Learn more:

- Text inputs and outputs
- Structured Outputs


```
  "top_p": 1.0,
  "truncation": "disabled",
  "usage": {
    "input_tokens": 328,
    "input_tokens_details": {
      "cached_tokens": 0
    },
    "output_tokens": 52,
    "output_tokens_details": {
      "reasoning_tokens": 0
    },
    "total_tokens": 380
  },
  "user": null,
  "metadata": {}
}
```

Show properties

`tool_choice` string or object

How the model should select which tool (or tools) to use when generating a response. See the `tools` parameter to see how to specify which tools the model can call.

Show possible types

`tools` array

An array of tools the model may call while generating a response. You can specify which tool to use by setting the `tool_choice` parameter.

The two categories of tools you can provide the model are:

- **Built-in tools**: Tools that are provided by OpenAI that extend the model's capabilities, like web search or file search. Learn more about built-in tools.
- **Function calls (custom tools)**: Functions that are defined by you, enabling the model to call your own code. Learn more about function calling.

Show possible types

`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.

`truncation` string or null

The truncation strategy to use for the model response.

- `auto` : If the context of this response and previous ones exceeds the model's context window size, the model will truncate the response to fit the context window by dropping input items in the middle of the conversation.
- `disabled` (default): If a model response will exceed the context window size for a model, the request will fail with a 400 error.

`usage` object

Represents token usage details including input tokens, output tokens, a breakdown of output tokens, and the total tokens used.

Show properties

`user` string

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

The input item list

A list of Response items.

`data` array

A list of items used to generate this response.

Show possible types

`first_id` string

The ID of the first item in the list.

`has_more` boolean

Whether there are more items available.

`last_id` string

The ID of the last item in the list.

OBJECT The input item list


```
{
  "object": "list",
  "data": [
    {
      "id": "msg_abc123",
      "type": "message",
      "role": "user",
      "content": [
        {
          "type": "input_text",
          "text": "Tell me a three sentence bed
        }
      ]
    }
```

object string
The type of object returned, must be `list`.

```
  ],
  "first_id": "msg_abc123",
  "last_id": "msg_abc123",
  "has_more": false
}
```

Streaming

When you create a Response with `stream` set to `true`, the server will emit server-sent events to the client as the Response is generated. This section contains the events that are emitted by the server.

Learn more about streaming responses.

response.created

An event that is emitted when a response is created.

response object
The response that was created.
Show properties

type string
The type of the event. Always `response.created`.

OBJECT response.created

```
{
  "type": "response.created",
  "response": {
    "id": "resp_67ccfcdd16748190a91872c75d3853
    "object": "response",
    "created_at": 1741487325,
    "status": "in_progress",
    "error": null,
    "incomplete_details": null,
    "instructions": null,
    "max_output_tokens": null,
    "model": "gpt-4o-2024-08-06",
    "output": [],
    "parallel_tool_calls": true,
    "previous_response_id": null,
    "reasoning": {
      "effort": null,
      "summary": null
    },
    "store": true,
    "temperature": 1,
    "text": {
      "format": {
        "type": "text"
      }
    },
    "tool_choice": "auto",
    "tools": [],
    "top_p": 1,
    "truncation": "disabled",
    "usage": null,
    "user": null,
    "metadata": {}
  }
}
```

response.in_progress

Emitted when the response is in progress.

response object
The response that is in progress.
Show properties

OBJECT response.in_progress

```
{
  "type": "response.in_progress",
  "response": {
    "id": "resp_67ccfcdd16748190a91872c75d3853
```

type string

The type of the event. Always `response.in_progress` .


```
  "object": "response",
  "created_at": 1741487325,
  "status": "in_progress",
  "error": null,
  "incomplete_details": null,
  "instructions": null,
  "max_output_tokens": null,
  "model": "gpt-4o-2024-08-06",
  "output": [],
  "parallel_tool_calls": true,
  "previous_response_id": null,
  "reasoning": {
    "effort": null,
    "summary": null
  },
  "store": true,
  "temperature": 1,
  "text": {
    "format": {
      "type": "text"
    }
  },
  "tool_choice": "auto",
  "tools": [],
  "top_p": 1,
  "truncation": "disabled",
  "usage": null,
  "user": null,
  "metadata": {}
}
```


response.completed

Emitted when the model response is complete.

response object

Properties of the completed response.

Show properties

type string

The type of the event. Always `response.completed` .

OBJECT response.completed


```
{
  "type": "response.completed",
  "response": {
    "id": "resp_123",
    "object": "response",
    "created_at": 1740855869,
    "status": "completed",
    "error": null,
    "incomplete_details": null,
    "input": [],
    "instructions": null,
    "max_output_tokens": null,
    "model": "gpt-4o-mini-2024-07-18",
    "output": [
      {
        "id": "msg_123",
        "type": "message",
        "role": "assistant",
        "content": [
          {
            "type": "output_text",
            "text": "In a shimmering forest unc
            "annotations": []
          }
        ]
      }
    ],
    "previous_response_id": null,
    "reasoning_effort": null,
    "store": false,
    "temperature": 1,
    "text": {
```

```
      "format": {
        "type": "text"
      }
    },
    "tool_choice": "auto",
    "tools": [],
    "top_p": 1,
    "truncation": "disabled",
    "usage": {
      "input_tokens": 0,
      "output_tokens": 0,
      "output_tokens_details": {
        "reasoning_tokens": 0
      },
      "total_tokens": 0
    },
    "user": null,
    "metadata": {}
  }
}
```


response.failed

An event that is emitted when a response fails.

response object

The response that failed.

Show properties

type string

The type of the event. Always `response.failed` .

OBJECT response.failed


```
{
  "type": "response.failed",
  "response": {
    "id": "resp_123",
    "object": "response",
    "created_at": 1740855869,
    "status": "failed",
    "error": {
      "code": "server_error",
      "message": "The model failed to generate
    },
    "incomplete_details": null,
    "instructions": null,
    "max_output_tokens": null,
    "model": "gpt-4o-mini-2024-07-18",
    "output": [],
    "previous_response_id": null,
    "reasoning_effort": null,
    "store": false,
    "temperature": 1,
    "text": {
      "format": {
        "type": "text"
      }
    },
    "tool_choice": "auto",
    "tools": [],
    "top_p": 1,
    "truncation": "disabled",
    "usage": null,
    "user": null,
    "metadata": {}
  }
}
```


response.incomplete

An event that is emitted when a response finishes as incomplete.

OBJECT response.incomplete

response object

The response that was incomplete.

Show properties

type string

The type of the event. Always `response.incomplete`.

```
{
  "type": "response.incomplete",
  "response": {
    "id": "resp_123",
    "object": "response",
    "created_at": 1740855869,
    "status": "incomplete",
    "error": null,
    "incomplete_details": {
      "reason": "max_tokens"
    },
    "instructions": null,
    "max_output_tokens": null,
    "model": "gpt-4o-mini-2024-07-18",
    "output": [],
    "previous_response_id": null,
    "reasoning_effort": null,
    "store": false,
    "temperature": 1,
    "text": {
      "format": {
        "type": "text"
      }
    },
    "tool_choice": "auto",
    "tools": [],
    "top_p": 1,
    "truncation": "disabled",
    "usage": null,
    "user": null,
    "metadata": {}
  }
}
```

response.output_item.added

Emitted when a new output item is added.

item object

The output item that was added.

Show possible types

output_index integer

The index of the output item that was added.

type string

The type of the event. Always `response.output_item.added`.

OBJECT response.output_item.added

```
{
  "type": "response.output_item.added",
  "output_index": 0,
  "item": {
    "id": "msg_123",
    "status": "in_progress",
    "type": "message",
    "role": "assistant",
    "content": []
  }
}
```

response.output_item.done

Emitted when an output item is marked done.

item object

The output item that was marked done.

Show possible types

output_index integer

The index of the output item that was marked done.

type string

OBJECT response.output_item.done

```
{
  "type": "response.output_item.done",
  "output_index": 0,
  "item": {
    "id": "msg_123",
    "status": "completed",
    "type": "message",
    "role": "assistant",
    "content": [
      {
```

The type of the event. Always `response.output_item.done`.

```
        "type": "output_text",
        "text": "In a shimmering forest under a
        "annotations": []
      }
    ]
  }
}
```

response.content_part.added

Emitted when a new content part is added.

content_index integer
The index of the content part that was added.

item_id string
The ID of the output item that the content part was added to.

output_index integer
The index of the output item that the content part was added to.

part object
The content part that was added.

Show possible types

type string
The type of the event. Always `response.content_part.added`.

OBJECT response.content_part.added

```
{
  "type": "response.content_part.added",
  "item_id": "msg_123",
  "output_index": 0,
  "content_index": 0,
  "part": {
    "type": "output_text",
    "text": "",
    "annotations": []
  }
}
```

response.content_part.done

Emitted when a content part is done.

content_index integer
The index of the content part that is done.

item_id string
The ID of the output item that the content part was added to.

output_index integer
The index of the output item that the content part was added to.

part object
The content part that is done.

Show possible types

type string
The type of the event. Always `response.content_part.done`.

OBJECT response.content_part.done

```
{
  "type": "response.content_part.done",
  "item_id": "msg_123",
  "output_index": 0,
  "content_index": 0,
  "part": {
    "type": "output_text",
    "text": "In a shimmering forest under a sky
    "annotations": []
  }
}
```

response.output_text.delta

Emitted when there is an additional text delta.

content_index integer
The index of the content part that the text delta was added to.

delta string

OBJECT response.output_text.delta

```
{
  "type": "response.output_text.delta",
  "item_id": "msg_123",
  "output_index": 0,
  "content_index": 0,
```

The text delta that was added.

item_id string
The ID of the output item that the text delta was added to.

output_index integer
The index of the output item that the text delta was added to.

type string
The type of the event. Always `response.output_text.delta`.

```
6   "delta": "In"
7 }
```

response.output_text.annotation.added

Emitted when a text annotation is added.

annotation object
Show possible types

annotation_index integer
The index of the annotation that was added.

content_index integer
The index of the content part that the text annotation was added to.

item_id string
The ID of the output item that the text annotation was added to.

output_index integer
The index of the output item that the text annotation was added to.

type string
The type of the event. Always `response.output_text.annotation.added`.

OBJECT response.output_text.annotation.added

```
{
  "type": "response.output_text.annotation.adde
  "item_id": "msg_abc123",
  "output_index": 1,
  "content_index": 0,
  "annotation_index": 0,
  "annotation": {
    "type": "file_citation",
    "index": 390,
    "file_id": "file-4wDz5b167pAf72nx1h9eiN",
    "filename": "dragons.pdf"
  }
}
```

response.output_text.done

Emitted when text content is finalized.

content_index integer
The index of the content part that the text content is finalized.

item_id string
The ID of the output item that the text content is finalized.

output_index integer
The index of the output item that the text content is finalized.

text string
The text content that is finalized.

type string
The type of the event. Always `response.output_text.done`.

OBJECT response.output_text.done

```
{
  "type": "response.output_text.done",
  "item_id": "msg_123",
  "output_index": 0,
  "content_index": 0,
  "text": "In a shimmering forest under a sky fu
}
```

response.refusal.delta

Emitted when there is a partial refusal text.

content_index integer
The index of the content part that the refusal text is added to.

delta string
The refusal text that is added.

item_id string
The ID of the output item that the refusal text is added to.

output_index integer
The index of the output item that the refusal text is added to.

type string
The type of the event. Always `response.refusal.delta` .

OBJECT response.refusal.delta

```
{
  "type": "response.refusal.delta",
  "item_id": "msg_123",
  "output_index": 0,
  "content_index": 0,
  "delta": "refusal text so far"
}
```

response.refusal.done

Emitted when refusal text is finalized.

content_index integer
The index of the content part that the refusal text is finalized.

item_id string
The ID of the output item that the refusal text is finalized.

output_index integer
The index of the output item that the refusal text is finalized.

refusal string
The refusal text that is finalized.

type string
The type of the event. Always `response.refusal.done` .

OBJECT response.refusal.done

```
{
  "type": "response.refusal.done",
  "item_id": "item-abc",
  "output_index": 1,
  "content_index": 2,
  "refusal": "final refusal text"
}
```

response.function_call_arguments.delta

Emitted when there is a partial function-call arguments delta.

delta string
The function-call arguments delta that is added.

item_id string
The ID of the output item that the function-call arguments delta is added to.

output_index integer
The index of the output item that the function-call arguments delta is added to.

type string
The type of the event. Always `response.function_call_arguments.delta` .

OBJECT response.function_call_arguments.delta

```
{
  "type": "response.function_call_arguments.del
  "item_id": "item-abc",
  "output_index": 0,
  "delta": "{ \"arg\":"
}
```

response.function_call_arguments.done

Emitted when function-call arguments are finalized.

arguments string
The function-call arguments.

item_id string
The ID of the item.

output_index integer
The index of the output item.

type string

OBJECT response.function_call_arguments.done

```
{
  "type": "response.function_call_arguments.done
  "item_id": "item-abc",
  "output_index": 1,
  "arguments": "{ \"arg\": 123 }"
}
```

response.file_search_call.in_progress

Emitted when a file search call is initiated.

item_id string
The ID of the output item that the file search call is initiated.

output_index integer
The index of the output item that the file search call is initiated.

type string
The type of the event. Always `response.file_search_call.in_progress`.

OBJECT response.file_search_call.in_progress

```
{
  "type": "response.file_search_call.in_progress
  "output_index": 0,
  "item_id": "fs_123",
}
```

response.file_search_call.searching

Emitted when a file search is currently searching.

item_id string
The ID of the output item that the file search call is initiated.

output_index integer
The index of the output item that the file search call is searching.

type string
The type of the event. Always `response.file_search_call.searching`.

OBJECT response.file_search_call.searching

```
{
  "type": "response.file_search_call.searching"
  "output_index": 0,
  "item_id": "fs_123",
}
```

response.file_search_call.completed

Emitted when a file search call is completed (results found).

item_id string
The ID of the output item that the file search call is initiated.

output_index integer
The index of the output item that the file search call is initiated.

OBJECT response.file_search_call.completed

```
{
  "type": "response.file_search_call.completed"
  "output_index": 0,
  "item_id": "fs_123",
}
```

type string
The type of the event. Always `response.file_search_call.completed` .

response.web_search_call.in_progress

Emitted when a web search call is initiated.

item_id string
Unique ID for the output item associated with the web search call.

output_index integer
The index of the output item that the web search call is associated with.

type string
The type of the event. Always `response.web_search_call.in_progress` .

OBJECT response.web_search_call.in_progress

```
{
  "type": "response.web_search_call.in_progress"
  "output_index": 0,
  "item_id": "ws_123",
}
```

response.web_search_call.searching

Emitted when a web search call is executing.

item_id string
Unique ID for the output item associated with the web search call.

output_index integer
The index of the output item that the web search call is associated with.

type string
The type of the event. Always `response.web_search_call.searching` .

OBJECT response.web_search_call.searching

```
{
  "type": "response.web_search_call.searching",
  "output_index": 0,
  "item_id": "ws_123",
}
```

response.web_search_call.completed

Emitted when a web search call is completed.

item_id string
Unique ID for the output item associated with the web search call.

output_index integer
The index of the output item that the web search call is associated with.

type string
The type of the event. Always `response.web_search_call.completed` .

OBJECT response.web_search_call.completed

```
{
  "type": "response.web_search_call.completed",
  "output_index": 0,
  "item_id": "ws_123",
}
```

error

Emitted when an error occurs.

code string or null
The error code.

OBJECT error

```
{
  "type": "error",
  "code": "ERR_SOMETHING",
```

message string
The error message.

param string or null
The error parameter.

type string
The type of the event. Always `error`.

```
4   "message": "Something went wrong",
5   "param": null
6 }
```

Chat Completions

The Chat Completions API endpoint will generate a model response from a list of messages comprising a conversation.

Related guides:

- Quickstart
- Text inputs and outputs
- Image inputs
- Audio inputs and outputs
- Structured Outputs
- Function calling
- Conversation state

Starting a new project? We recommend trying Responses to take advantage of the latest OpenAI platform features. Compare Chat Completions with Responses.

Create chat completion

`POST https://api.openai.com/v1/chat/completions`

Starting a new project? We recommend trying Responses to take advantage of the latest OpenAI platform features. Compare Chat Completions with Responses.

Creates a model response for the given chat conversation. Learn more in the text generation, vision, and audio guides.
Parameter support can differ depending on the model used to generate the response, particularly for newer reasoning models. Parameters that are only supported for reasoning models are noted below. For the current state of unsupported parameters in reasoning models, refer to the reasoning guide.

Request body

messages array Required
A list of messages comprising the conversation so far. Depending on the model you use, different message types (modalities) are supported, like text, images, and audio.
Show possible types

model string Required

Default | Image input | Streaming | Functions | Logp

Example request gpt-4.1 curl

```
curl https://api.openai.com/v1/chat/completions
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -d '{
    "model": "gpt-4.1",
    "messages": [
      {
        "role": "developer",
        "content": "You are a helpful assistant
      },
      {
        "role": "user",
        "content": "Hello!"
      }
    ]
  }'
```

Response

```
{
  "id": "chatcmpl-B9MBs8CjcvOU2jLn4n570S5qMJKc
  "object": "chat.completion",
  "created": 1741569952,
  "model": "gpt-4.1-2025-04-14",
```

Model ID used to generate the response, like `gpt-4o` or `o3`. OpenAI offers a wide range of models with different capabilities, performance characteristics, and price points. Refer to the model guide to browse and compare available models.

audio object or null Optional

Parameters for audio output. Required when audio output is requested with `modalities: ["audio"]`. Learn more.

Show properties

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.

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

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`.

max_completion_tokens integer or null Optional

An upper bound for the number of tokens that can be generated for a completion, including visible output tokens and reasoning tokens.

max_tokens Deprecated integer or null Optional

The maximum number of tokens that can be generated in the chat completion. This value can be used to control costs for text generated via API.

This value is now deprecated in favor of `max_completion_tokens`, and is not compatible with o-series models.

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, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.


```
"choices": [
  {
    "index": 0,
    "message": {
      "role": "assistant",
      "content": "Hello! How can I assist you
      "refusal": null,
      "annotations": []
    },
    "logprobs": null,
    "finish_reason": "stop"
  }
],
"usage": {
  "prompt_tokens": 19,
  "completion_tokens": 10,
  "total_tokens": 29,
  "prompt_tokens_details": {
    "cached_tokens": 0,
    "audio_tokens": 0
  },
  "completion_tokens_details": {
    "reasoning_tokens": 0,
    "audio_tokens": 0,
    "accepted_prediction_tokens": 0,
    "rejected_prediction_tokens": 0
  }
},
"service_tier": "default"
}
```

modalities array or null Optional

Output types that you would like the model to generate. Most models are capable of generating text, which is the default:

`["text"]`

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

`["text", "audio"]`

---

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.

---

parallel_tool_calls boolean Optional Defaults to true

Whether to enable parallel function calling during tool use.

---

prediction object Optional

Configuration for a Predicted Output, which can greatly improve response times when large parts of the model response are known ahead of time. This is most common when you are regenerating a file with only minor changes to most of the content.

Show possible types

---

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.

---

reasoning_effort string or null Optional Defaults to medium

o-series models only

Constrains effort on reasoning for reasoning models. Currently supported values are `low`, `medium`, and `high`. Reducing reasoning effort can result in faster responses and fewer tokens used on reasoning in a response.

---

response_format object Optional

An object specifying the format that the model must output.

Setting to `{ "type": "json_schema", "json_schema": {...} }` enables Structured Outputs which ensures the model will match your supplied JSON schema. Learn more in the Structured Outputs guide.

Setting to `{ "type": "json_object" }` enables the older JSON mode, which ensures the message the model generates is valid JSON. Using `json_schema` is preferred for models that support it.

Show possible types

---

seed integer or null Optional

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

---

service_tier string or null Optional Defaults to auto

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

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

- If set to 'flex', the request will be processed with the Flex Processing service tier. Learn more.
- When not set, the default behavior is 'auto'.

When this parameter is set, the response body will include the `service_tier` utilized.

`stop` string / array / null Optional Defaults to null
Not supported with latest reasoning models `o3` and `o4-mini`.

Up to 4 sequences where the API will stop generating further tokens. The returned text will not contain the stop sequence.

`store` boolean or null Optional Defaults to false
Whether or not to store the output of this chat completion request for use in our model distillation or evals products.

`stream` boolean or null Optional Defaults to false
If set to true, the model response data will be streamed to the client as it is generated using server-sent events. See the Streaming section below for more information, along with the streaming responses guide for more information on how to handle the streaming events.

`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.

`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

`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

`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.

`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.

`web_search_options` object Optional

This tool searches the web for relevant results to use in a response. Learn more about the web search tool.

Show properties

Returns

Returns a chat completion object, or a streamed sequence of chat completion chunk objects if the request is streamed.

Get chat completion

`GET https://api.openai.com/v1/chat/completions/{completion_id}`

Get a stored chat completion. Only Chat Completions that have been created with the `store` parameter set to `true` will be returned.

Path parameters

`completion_id` string Required

The ID of the chat completion to retrieve.

Returns

The ChatCompletion object matching the specified ID.

Example request curl

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

Response

```
{
  "object": "chat.completion",
  "id": "chatcmpl-abc123",
  "model": "gpt-4o-2024-08-06",
  "created": 1738960610,
  "request_id": "req_ded8ab984ec4bf840f37566c1
  "tool_choice": null,
  "usage": {
    "total_tokens": 31,
    "completion_tokens": 18,
    "prompt_tokens": 13
  },
  "seed": 4944116822809979520,
  "top_p": 1.0,
  "temperature": 1.0,
  "presence_penalty": 0.0,
  "frequency_penalty": 0.0,
  "system_fingerprint": "fp_50cad350e4",
  "input_user": null,
  "service_tier": "default",
  "tools": null,
  "metadata": {},
  "choices": [
    {
      "index": 0,
      "message": {
        "content": "Mind of circuits hum,  \nLe
        "role": "assistant",
        "tool_calls": null,
        "function_call": null
      },
      "finish_reason": "stop",
      "logprobs": null
    }
  ],
  "response_format": null
}
```

Get chat messages

```
GET https://api.openai.com/v1/chat/completions/{completion_id}/messages
```

Get the messages in a stored chat completion. Only Chat Completions that have been created with the `store` parameter set to `true` will be returned.

Path parameters

`completion_id` string Required
The ID of the chat completion to retrieve messages from.

Query parameters

`after` string Optional
Identifier for the last message from the previous pagination request.

`limit` integer Optional Defaults to 20
Number of messages to retrieve.

`order` string Optional Defaults to asc
Sort order for messages by timestamp. Use `asc` for ascending order or `desc` for descending order. Defaults to `asc`.

Returns

A list of messages for the specified chat completion.

Example request curl

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

Response

```
{
  "object": "list",
  "data": [
    {
      "id": "chatcmpl-AyPNinnUqUDYo9SAdA52NobM
      "role": "user",
      "content": "write a haiku about ai",
      "name": null,
      "content_parts": null
    }
  ],
  "first_id": "chatcmpl-AyPNinnUqUDYo9SAdA52Nol
  "last_id": "chatcmpl-AyPNinnUqUDYo9SAdA52NobM
  "has_more": false
}
```

List Chat Completions

```
GET https://api.openai.com/v1/chat/completions
```

List stored Chat Completions. Only Chat Completions that have been stored with the `store` parameter set to `true` will be returned.

Query parameters

`after` string Optional
Identifier for the last chat completion from the previous pagination request.

`limit` integer Optional Defaults to 20
Number of Chat Completions to retrieve.

`metadata` map Optional
A list of metadata keys to filter the Chat Completions by. Example:
`metadata[key1]=value1&metadata[key2]=value2`

`model` string Optional
The model used to generate the Chat Completions.

`order` string Optional Defaults to asc

Example request curl

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

Response

```
{
  "object": "list",
  "data": [
    {
      "object": "chat.completion",
      "id": "chatcmpl-AyPNinnUqUDYo9SAdA52NobM
      "model": "gpt-4.1-2025-04-14",
      "created": 1738960610,
      "request_id": "req_ded8ab984ec4bf840f375
      "tool_choice": null,
      "usage": {
        "total_tokens": 31,
        "completion_tokens": 18,
        "prompt_tokens": 13
      },
      "seed": 4944116822809979520,
```

Sort order for Chat Completions by timestamp. Use `asc` for ascending order or `desc` for descending order. Defaults to `asc`.

Returns

A list of Chat Completions matching the specified filters.


```
    "top_p": 1.0,
    "temperature": 1.0,
    "presence_penalty": 0.0,
    "frequency_penalty": 0.0,
    "system_fingerprint": "fp_50cad350e4",
    "input_user": null,
    "service_tier": "default",
    "tools": null,
    "metadata": {},
    "choices": [
      {
        "index": 0,
        "message": {
          "content": "Mind of circuits hum,
          "role": "assistant",
          "tool_calls": null,
          "function_call": null
        },
        "finish_reason": "stop",
        "logprobs": null
      }
    ],
    "response_format": null
  }
  ],
  "first_id": "chatcmpl-AyPNinnUqUDYo9SAdA52Nol
  "last_id": "chatcmpl-AyPNinnUqUDYo9SAdA52NobM
"has_more": false
}
```


Update chat completion

`POST https://api.openai.com/v1/chat/completions/{completion_id}`

Modify a stored chat completion. Only Chat Completions that have been created with the `store` parameter set to `true` can be modified. Currently, the only supported modification is to update the `metadata` field.

Path parameters

`completion_id` string Required

The ID of the chat completion to update.

Request body

`metadata` map Required

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, and querying for objects via API or the dashboard.

Keys are strings with a maximum length of 64 characters. Values are strings with a maximum length of 512 characters.

Returns

The ChatCompletion object matching the specified ID.

Example request — curl

```
curl -X POST https://api.openai.com/v1/chat/com
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{"metadata": {"foo": "bar"}}'
```

Response


```
{
  "object": "chat.completion",
  "id": "chatcmpl-AyPNinnUqUDYo9SAdA52NobMflmj
  "model": "gpt-4o-2024-08-06",
  "created": 1738960610,
  "request_id": "req_ded8ab984ec4bf840f37566c1
  "tool_choice": null,
  "usage": {
    "total_tokens": 31,
    "completion_tokens": 18,
    "prompt_tokens": 13
  },
  "seed": 4944116822809979520,
  "top_p": 1.0,
  "temperature": 1.0,
  "presence_penalty": 0.0,
  "frequency_penalty": 0.0,
  "system_fingerprint": "fp_50cad350e4",
  "input_user": null,
  "service_tier": "default",
  "tools": null,
  "metadata": {
    "foo": "bar"
  },
  "choices": [
    {
```

```
      "index": 0,
      "message": {
        "content": "Mind of circuits hum,  \nLe
        "role": "assistant",
        "tool_calls": null,
        "function_call": null
      },
      "finish_reason": "stop",
      "logprobs": null
    }
  ],
 "response_format": null
}
```

Delete chat completion

`DELETE https://api.openai.com/v1/chat/completions/{completion_id}`

Delete a stored chat completion. Only Chat Completions that have been created with the `store` parameter set to `true` can be deleted.

Path parameters

`completion_id` string Required
The ID of the chat completion to delete.

Returns

A deletion confirmation object.

Example request — curl

```
curl -X DELETE https://api.openai.com/v1/chat/c
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json"
```

Response

```
{
  "object": "chat.completion.deleted",
  "id": "chatcmpl-AyPNinnUqUDYo9SAdA52NobMflmj2
  "deleted": true
}
```

The chat completion object

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

`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.

`id` string
A unique identifier for the chat completion.

`model` string
The model used for the chat completion.

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

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

- If set to 'auto', and the Project is Scale tier enabled, the system will utilize scale tier credits until they are exhausted.

OBJECT The chat completion object

```
{
  "id": "chatcmpl-B9MHDbslfkBeAs8l4bebGdFOJ6PeG
  "object": "chat.completion",
  "created": 1741570283,
  "model": "gpt-4o-2024-08-06",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "The image shows a wooden bo
        "refusal": null,
        "annotations": []
      },
      "logprobs": null,
      "finish_reason": "stop"
    }
  ],
  "usage": {
    "prompt_tokens": 1117,
    "completion_tokens": 46,
    "total_tokens": 1163,
    "prompt_tokens_details": {
      "cached_tokens": 0,
      "audio_tokens": 0
    },
```

- If set to 'auto', and the Project is not Scale tier enabled, the request will be processed using the default service tier with a lower uptime SLA and no latency guarentee.
- If set to 'default', the request will be processed using the default service tier with a lower uptime SLA and no latency guarentee.
- If set to 'flex', the request will be processed with the Flex Processing service tier. Learn more.
- When not set, the default behavior is 'auto'.

When this parameter is set, the response body will include the `service_tier` utilized.

`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.

`usage` object

Usage statistics for the completion request.

Show properties

```
    "completion_tokens_details": {
      "reasoning_tokens": 0,
      "audio_tokens": 0,
      "accepted_prediction_tokens": 0,
      "rejected_prediction_tokens": 0
    }
  },
  "service_tier": "default",
  "system_fingerprint": "fp_fc9f1d7035"
}
```

The chat completion list object

An object representing a list of Chat Completions.

`data` array

An array of chat completion objects.

Show properties

`first_id` string

The identifier of the first chat completion in the data array.

`has_more` boolean

Indicates whether there are more Chat Completions available.

`last_id` string

The identifier of the last chat completion in the data array.

`object` string

The type of this object. It is always set to "list".

OBJECT The chat completion list object

```
{
  "object": "list",
  "data": [
    {
      "object": "chat.completion",
      "id": "chatcmpl-AyPNinnUqUDYo9SAdA52NobM
      "model": "gpt-4o-2024-08-06",
      "created": 1738960610,
      "request_id": "req_ded8ab984ec4bf840f375
      "tool_choice": null,
      "usage": {
        "total_tokens": 31,
        "completion_tokens": 18,
        "prompt_tokens": 13
      },
      "seed": 4944116822809979520,
      "top_p": 1.0,
      "temperature": 1.0,
      "presence_penalty": 0.0,
      "frequency_penalty": 0.0,
      "system_fingerprint": "fp_50cad350e4",
      "input_user": null,
      "service_tier": "default",
      "tools": null,
      "metadata": {},
      "choices": [
        {
          "index": 0,
          "message": {
            "content": "Mind of circuits hum,
            "role": "assistant",
            "tool_calls": null,
            "function_call": null
          },
          "finish_reason": "stop",
          "logprobs": null
        }
      ],
      "response_format": null
```

```
    }
  ],
  "first_id": "chatcmpl-AyPNinnUqUDYo9SAdA52Nol
  "last_id": "chatcmpl-AyPNinnUqUDYo9SAdA52NobM
  "has_more": false
}
```


The chat completion message list object

An object representing a list of chat completion messages.

data array

An array of chat completion message objects.

Show properties

first_id string

The identifier of the first chat message in the data array.

has_more boolean

Indicates whether there are more chat messages available.

last_id string

The identifier of the last chat message in the data array.

object string

The type of this object. It is always set to "list".

OBJECT The chat completion message list obje…


```
{
  "object": "list",
  "data": [
    {
      "id": "chatcmpl-AyPNinnUqUDYo9SAdA52NobM
      "role": "user",
      "content": "write a haiku about ai",
      "name": null,
      "content_parts": null
    }
  ],
  "first_id": "chatcmpl-AyPNinnUqUDYo9SAdA52Nol
  "last_id": "chatcmpl-AyPNinnUqUDYo9SAdA52NobM
  "has_more": false
}
```


Streaming

Stream Chat Completions in real time. Receive chunks of completions returned from the model using server-sent events. Learn more.

The chat completion chunk object

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

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.

id string

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

model string

The model to generate the completion.

OBJECT The chat completion chunk object


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

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

....

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

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

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

- If set to 'auto', and the Project is Scale tier enabled, the system will utilize scale tier credits until they are exhausted.
- If set to 'auto', and the Project is not Scale tier enabled, the request will be processed using the default service tier with a lower uptime SLA and no latency guarentee.
- If set to 'default', the request will be processed using the default service tier with a lower uptime SLA and no latency guarentee.
- If set to 'flex', the request will be processed with the Flex Processing service tier. Learn more.
- When not set, the default behavior is 'auto'.

When this parameter is set, the response body will include the `service_tier` utilized.

`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.

`usage` object or null
Usage statistics for the completion request.

Show properties

Realtime Beta

Communicate with a GPT-4o class model in real time using WebRTC or WebSockets. Supports text and audio inputs and ouputs, along with audio transcriptions. Learn more about the Realtime API.

Session tokens

REST API endpoint to generate ephemeral session tokens for use in client-side applications.

Create session

`POST https://api.openai.com/v1/realtime/sessions`

Create an ephemeral API token for use in client-side applications with the Realtime API. Can be configured with the same session parameters as the `session.update` client event.
It responds with a session object, plus a `client_secret` key which contains a usable ephemeral API token that can be used to authenticate browser clients for the Realtime API.

Request body

`input_audio_format` string Optional Defaults to pcm16

Example request curl

```
curl -X POST https://api.openai.com/v1/realtime/
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "model": "gpt-4o-realtime-preview",
    "modalities": ["audio", "text"],
    "instructions": "You are a friendly assistar
  }'
```

Response