Assistants Beta

Build assistants that can call models and use tools to perform tasks.

Get started with the Assistants API

Create assistant Beta

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

Create an assistant with a model and instructions.

Request body

model Required
ID of the model to use. You can use the **List models** API to see all of your available models, or see our **Model overview** for descriptions of them.

name string or null Optional
The name of the assistant. The maximum length is 256 characters.

description string or null Optional
The description of the assistant. The maximum length is 512 characters.

instructions string or null Optional
The system instructions that the assistant uses. The maximum length is 32768 characters.

tools array Optional Defaults to []
A list of tool enabled on the assistant. There can be a maximum of 128 tools per assistant. Tools can be of types code_interpreter , retrieval , or function .
Show possible types

file_ids array Optional Defaults to []
A list of **file** IDs attached to this assistant. There can be a maximum of 20 files attached to the assistant. Files are ordered by their creation date in ascending order.

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

Returns

Code Interpreter Files

Example request node.js

```
import OpenAI from "openai";

const openai = new OpenAI();

async function main() {
  const myAssistant = await openai.beta.assistants.create({
    instructions:
      "You are a personal math tutor. When asked a question, w
    name: "Math Tutor",
    tools: [{ type: "code_interpreter" }],
    model: "gpt-4",
  });

  console.log(myAssistant);
}

main();
```

Response

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

An **assistant** object.

Create assistant file Beta

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

Create an assistant file by attaching a **File** to an **assistant**.

Path parameters

assistant_id string Required
The ID of the assistant for which to create a File.

Request body

file_id string Required
A **File** ID (with purpose="assistants") that the assistant should use. Useful for tools like retrieval and code_interpreter that can access files.

Returns

An **assistant file** object.

Example request node.js

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

async function main() {
  const myAssistantFile = await openai.beta.assistants.files.cr
    "asst_abc123",
    {
      file_id: "file-abc123"
    }
  );
  console.log(myAssistantFile);
}

main();
```

Response

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

List assistants Beta

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

Returns a list of assistants.

Query parameters

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

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

after string Optional
A cursor for use in pagination. after is an object ID that defines your place in the list. For instance, if you make a list request and

Example request node.js

```
import OpenAI from "openai";

const openai = new OpenAI();

async function main() {
  const myAssistants = await openai.beta.assistants.list({
    order: "desc",
    limit: "20",
  });

  console.log(myAssistants.data);
}

main();
```

Response

receive 100 objects, ending with obj_foo, your subsequent call can include after=obj_foo in order to fetch the next page of the list.

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

Returns

A list of **assistant** objects.

```
{
  "object": "list",
  "data": [
    {
      "id": "asst_abc123",
      "object": "assistant",
      "created_at": 1698982736,
      "name": "Coding Tutor",
      "description": null,
      "model": "gpt-4",
      "instructions": "You are a helpful assistant designed to ma
      "tools": [],
      "file_ids": [],
      "metadata": {}
    },
    {
      "id": "asst_abc456",
      "object": "assistant",
      "created_at": 1698982718,
      "name": "My Assistant",
      "description": null,
      "model": "gpt-4",
      "instructions": "You are a helpful assistant designed to ma
      "tools": [],
      "file_ids": [],
      "metadata": {}
    },
    {
      "id": "asst_abc789",
      "object": "assistant",
      "created_at": 1698982643,
      "name": null,
      "description": null,
      "model": "gpt-4",
      "instructions": null,
      "tools": [],
      "file_ids": [],
      "metadata": {}
    }
  ],
  "first_id": "asst_abc123",
  "last_id": "asst_abc789",
  "has_more": false
}
```

List assistant files Beta

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

Returns a list of assistant files.

Path parameters

assistant_id string Required

Example request node.js

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

async function main() {
  const assistantFiles = await openai.beta.assistants.files.list(
    "asst_abc123"
  );
```

The ID of the assistant the file belongs to.

Query parameters

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

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

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

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

Returns

A list of **assistant file** objects.

Retrieve assistant Beta

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

Retrieves an assistant.

Path parameters

assistant_id string Required
The ID of the assistant to retrieve.

Returns

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

```
8     console.log(assistantFiles);
9   }
10
11  main();
```

Response

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

Example request node.js

```
import OpenAI from "openai";

const openai = new OpenAI();

async function main() {
  const myAssistant = await openai.beta.assistants.retrieve(
    "asst_abc123"
  );

  console.log(myAssistant);
}

main();
```

Response

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

Retrieve assistant file Beta

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

Retrieves an AssistantFile.

Path parameters

assistant_id string Required
The ID of the assistant who the file belongs to.

file_id string Required
The ID of the file we're getting.

Returns

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

Example request node.js

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

async function main() {
  const myAssistantFile = await openai.beta.assistants.files.re
    "asst_abc123",
    "file-abc123"
  );
  console.log(myAssistantFile);
}

main();
```

Response

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

Modify assistant Beta

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

Modifies an assistant.

Path parameters

Example request node.js

```
import OpenAI from "openai";

const openai = new OpenAI();

async function main() {
```

assistant_id string Required
The ID of the assistant to modify.

Request body

model Optional
ID of the model to use. You can use the **List models** API to see all of your available models, or see our **Model overview** for descriptions of them.

name string or null Optional
The name of the assistant. The maximum length is 256 characters.

description string or null Optional
The description of the assistant. The maximum length is 512 characters.

instructions string or null Optional
The system instructions that the assistant uses. The maximum length is 32768 characters.

tools array Optional Defaults to []
A list of tool enabled on the assistant. There can be a maximum of 128 tools per assistant. Tools can be of types code_interpreter , retrieval , or function .
Show possible types

file_ids array Optional Defaults to []
A list of **File** IDs attached to this assistant. There can be a maximum of 20 files attached to the assistant. Files are ordered by their creation date in ascending order. If a file was previously attached to the list but does not show up in the list, it will be deleted from the assistant.

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

```
6   const myUpdatedAssistant = await openai.beta.assistants.up
7     "asst_abc123",
8     {
9       instructions:
10        "You are an HR bot, and you have access to files to answ
11      name: "HR Helper",
12      tools: [{ type: "retrieval" }],
13      model: "gpt-4",
14      file_ids: [
15        "file-abc123",
16        "file-abc456",
17      ],
18    }
19  );
20
21  console.log(myUpdatedAssistant);
22 }
23
24 main();
```

Response

```
1  {
2    "id": "asst_abc123",
3    "object": "assistant",
4    "created_at": 1699009709,
5    "name": "HR Helper",
6    "description": null,
7    "model": "gpt-4",
8    "instructions": "You are an HR bot, and you have access to fi
9    "tools": [
10     {
11       "type": "retrieval"
12     }
13   ],
14   "file_ids": [
15     "file-abc123",
16     "file-abc456"
17   ],
18   "metadata": {}
19 }
```

Returns

The modified **assistant** object.

Delete assistant Beta

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

Example request node.js

Delete an assistant.

Path parameters

assistant_id string Required
The ID of the assistant to delete.

Returns

Deletion status

```
import OpenAI from "openai";

const openai = new OpenAI();

async function main() {
  const response = await openai.beta.assistants.del("asst_abc

  console.log(response);
}
main();
```

Response

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

Delete assistant file Beta

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

Delete an assistant file.

Path parameters

assistant_id string Required
The ID of the assistant that the file belongs to.

file_id string Required
The ID of the file to delete.

Returns

Deletion status

Example request node.js

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

async function main() {
  const deletedAssistantFile = await openai.beta.assistants.file
    "asst_abc123",
    "file-abc123"
  );
  console.log(deletedAssistantFile);
}

main();
```

Response

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

The assistant object Beta

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

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

object string

The assistant object

```
{
  "id": "asst_abc123",
  "object": "assistant",
  "created_at": 1698984975,
```

The object type, which is always `assistant`.

created_at integer
The Unix timestamp (in seconds) for when the assistant was created.

name string or null
The name of the assistant. The maximum length is 256 characters.

description string or null
The description of the assistant. The maximum length is 512 characters.

model string
ID of the model to use. You can use the **List models** API to see all of your available models, or see our **Model overview** for descriptions of them.

instructions string or null
The system instructions that the assistant uses. The maximum length is 32768 characters.

tools array
A list of tool enabled on the assistant. There can be a maximum of 128 tools per assistant. Tools can be of types `code_interpreter`, `retrieval`, or `function`.
Show possible types

file_ids array
A list of **file** IDs attached to this assistant. There can be a maximum of 20 files attached to the assistant. Files are ordered by their creation date in ascending order.

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

```
5    "name": "Math Tutor",
6    "description": null,
7    "model": "gpt-4",
8    "instructions": "You are a personal math tutor. When asked a
9    "tools": [
10     {
11       "type": "code_interpreter"
12     }
13   ],
14   "file_ids": [],
15   "metadata": {}
16 }
```

The assistant file object Beta

A list of **Files** attached to an `assistant`.

The assistant file object

```
1  {
2    "id": "file-abc123",
3    "object": "assistant.file",
4    "created_at": 1699055364,
5    "assistant_id": "asst_abc123"
6  }
```

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

object string
The object type, which is always `assistant.file`.

created_at integer

The Unix timestamp (in seconds) for when the assistant file was created.

assistant_id string
The assistant ID that the file is attached to.

Threads Beta

Create threads that assistants can interact with.

Related guide: **Assistants**

Create thread Beta

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

Create a thread.

Request body

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

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

Returns

A **thread** object.

Empty Messages

Example request node.js

```
import OpenAI from "openai";

const openai = new OpenAI();

async function main() {
  const emptyThread = await openai.beta.threads.create();

  console.log(emptyThread);
}

main();
```

Response

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

Retrieve thread Beta

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

Retrieves a thread.

Path parameters

thread_id string Required
The ID of the thread to retrieve.

Returns

Example request node.js

```
import OpenAI from "openai";

const openai = new OpenAI();

async function main() {
  const myThread = await openai.beta.threads.retrieve(
    "thread_abc123"
  );

  console.log(myThread);
}

main();
```

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

Response

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

Modify thread Beta

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

Modifies a thread.

Path parameters

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

Request body

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

Returns

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

Example request node.js

```
import OpenAI from "openai";

const openai = new OpenAI();

async function main() {
  const updatedThread = await openai.beta.threads.update(
    "thread_abc123",
    {
      metadata: { modified: "true", user: "abc123" },
    }
  );

  console.log(updatedThread);
}

main();
```

Response

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

Delete thread Beta

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

Delete a thread.

Path parameters

thread_id string Required

Example request node.js

```
import OpenAI from "openai";

const openai = new OpenAI();

async function main() {
  const response = await openai.beta.threads.del("thread_abc
```

The ID of the thread to delete.

Returns

Deletion status

```
8     console.log(response);
9   }
10  main();
```

Response

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

The thread object Beta

Represents a thread that contains **messages**.

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

object string
The object type, which is always `thread`.

created_at integer
The Unix timestamp (in seconds) for when the thread was created.

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

The thread object

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

Messages Beta

Create messages within threads

Related guide: **Assistants**

Create message Beta

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

Create a message.

Path parameters

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

Request body

role string Required

Example request node.js

```
import OpenAI from "openai";

const openai = new OpenAI();

async function main() {
  const threadMessages = await openai.beta.threads.message
    "thread_abc123",
    { role: "user", content: "How does AI work? Explain it in sim
  );

  console.log(threadMessages);
}


```

The role of the entity that is creating the message. Currently only `user` is supported.

content string Required
The content of the message.

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

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

Returns

A **message** object.

List messages Beta

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

Returns a list of messages for a given thread.

Path parameters

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

Query parameters

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

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

after string Optional
A cursor for use in pagination. `after` is an object ID that defines your place in the list. For instance, if you make a list request and

```
main();
```

Response

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

Example request node.js

```
import OpenAI from "openai";

const openai = new OpenAI();

async function main() {
  const threadMessages = await openai.beta.threads.message
    "thread_abc123"
  );

  console.log(threadMessages.data);
}

main();
```

Response

```
{
  "object": "list",
  "data": [
    {
      "id": "msg_abc123",
      "object": "thread.message",
      "created_at": 1699016383,
      "thread_id": "thread_abc123",
```

receive 100 objects, ending with obj_foo, your subsequent call can include after=obj_foo in order to fetch the next page of the list.

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

Returns

A list of **message** objects.

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

List message files Beta

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

Returns a list of message files.

Path parameters

thread_id string Required
The ID of the thread that the message and files belong to.

Example request node.js

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

async function main() {
  const messageFiles = await openai.beta.threads.messages.
    "thread_abc123",
    "msg_abc123"
  );
  console.log(messageFiles);
```

message_id string Required
The ID of the message that the files belongs to.

Query parameters

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

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

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

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

Returns

A list of **message file** objects.

Retrieve message Beta

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

Retrieve a message.

Path parameters

thread_id string Required
The ID of the **thread** to which this message belongs.

message_id string Required
The ID of the message to retrieve.

Returns

Example request node.js

```node.js
import OpenAI from "openai";

const openai = new OpenAI();

async function main() {
  const message = await openai.beta.threads.messages.retrie
    "thread_abc123",
    "msg_abc123"
  );

  console.log(message);
}

main();
```

Response

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

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

Response

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

Retrieve message file Beta

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

Retrieves a message file.

Path parameters

thread_id string Required
The ID of the thread to which the message and File belong.

message_id string Required
The ID of the message the file belongs to.

file_id string Required
The ID of the file being retrieved.

Returns

The **message file** object.

Example request node.js

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

async function main() {
  const messageFile = await openai.beta.threads.messages.fil
    "thread_abc123",
    "msg_abc123",
    "file-abc123"
  );
  console.log(messageFile);
}

main();
```

Response

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

Modify message Beta

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

Example request node.js

Modifies a message.

Path parameters

thread_id string Required
The ID of the thread to which this message belongs.

message_id string Required
The ID of the message to modify.

Request body

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

Returns

The modified **message** object.

```
import OpenAI from "openai";

const openai = new OpenAI();

async function main() {
  const message = await openai.beta.threads.messages.upda
    "thread_abc123",
    "msg_abc123",
    {
      metadata: {
        modified: "true",
        user: "abc123",
      },
    }
  }'
```

Response

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

The message object Beta

Represents a message within a **thread**.

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

object string
The object type, which is always thread.message .

created_at integer

The message object

```
{
  "id": "msg_abc123",
  "object": "thread.message",
  "created_at": 1698983503,
  "thread_id": "thread_abc123",
  "role": "assistant",
  "content": [
    {
```

The Unix timestamp (in seconds) for when the message was created.

thread_id string
The **thread** ID that this message belongs to.

role string
The entity that produced the message. One of `user` or `assistant`.

content array
The content of the message in array of text and/or images.
Show possible types

assistant_id string or null
If applicable, the ID of the **assistant** that authored this message.

run_id string or null
If applicable, the ID of the **run** associated with the authoring of this message.

file_ids array
A list of **file** IDs that the assistant should use. Useful for tools like retrieval and code_interpreter that can access files. A maximum of 10 files can be attached to a message.

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

```
        "type": "text",
        "text": {
          "value": "Hi! How can I help you today?",
          "annotations": []
        }
      }
    ],
    "file_ids": [],
    "assistant_id": "asst_abc123",
    "run_id": "run_abc123",
    "metadata": {}
  }
```

The message file object Beta

A list of files attached to a `message`.

The message file object

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

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

object string
The object type, which is always `thread.message.file`.

created_at integer
The Unix timestamp (in seconds) for when the message file was created.

message_id string
The ID of the **message** that the **File** is attached to.

Runs Beta

Represents an execution run on a thread.

Related guide: **Assistants**

Create run Beta

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

Create a run.

Path parameters

thread_id string Required
The ID of the thread to run.

Request body

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

model string or null Optional
The ID of the **Model** to be used to execute this run. If a value is provided here, it will override the model associated with the assistant. If not, the model associated with the assistant will be used.

instructions string or null Optional
Overrides the **instructions** of the assistant. This is useful for modifying the behavior on a per-run basis.

additional_instructions string or null Optional
Appends additional instructions at the end of the instructions for the run. This is useful for modifying the behavior on a per-run basis without overriding other instructions.

tools array or null Optional
Override the tools the assistant can use for this run. This is useful for modifying the behavior on a per-run basis.
Show possible types

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

Returns

A **run** object.

Example request node.js

```
import OpenAI from "openai";

const openai = new OpenAI();

async function main() {
  const run = await openai.beta.threads.runs.create(
    "thread_abc123",
    { assistant_id: "asst_abc123" }
  );

  console.log(run);
}

main();
```

Response

```
{
  "id": "run_abc123",
  "object": "thread.run",
  "created_at": 1699063290,
  "assistant_id": "asst_abc123",
  "thread_id": "thread_abc123",
  "status": "queued",
  "started_at": 1699063290,
  "expires_at": null,
  "cancelled_at": null,
  "failed_at": null,
  "completed_at": 1699063291,
  "last_error": null,
  "model": "gpt-4",
  "instructions": null,
  "tools": [
    {
      "type": "code_interpreter"
    }
  ],
  "file_ids": [
    "file-abc123",
    "file-abc456"
  ],
  "metadata": {},
  "usage": null
}
```

Create thread and run Beta

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

Create a thread and run it in one request.

Request body

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

thread object Optional
Show properties

model string or null Optional
The ID of the **Model** to be used to execute this run. If a value is provided here, it will override the model associated with the assistant. If not, the model associated with the assistant will be used.

instructions string or null Optional
Override the default system message of the assistant. This is useful for modifying the behavior on a per-run basis.

tools array or null Optional
Override the tools the assistant can use for this run. This is useful for modifying the behavior on a per-run basis.

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

Returns

A **run** object.

Example request node.js

```node.js
import OpenAI from "openai";

const openai = new OpenAI();

async function main() {
  const run = await openai.beta.threads.createAndRun({
    assistant_id: "asst_abc123",
    thread: {
      messages: [
        { role: "user", content: "Explain deep learning to a 5 year
      ],
    },
  });

  console.log(run);
}

main();
```

Response

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

List runs Beta

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

Returns a list of runs belonging to a thread.

Path parameters

Example request node.js

```node.js
import OpenAI from "openai";

const openai = new OpenAI();

```

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

Query parameters

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

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

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

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

Returns

A list of **run** objects.

```
async function main() {
  const runs = await openai.beta.threads.runs.list(
    "thread_abc123"
  );

  console.log(runs);
}

main();
```

Response


```
{
  "object": "list",
  "data": [
    {
      "id": "run_abc123",
      "object": "thread.run",
      "created_at": 1699075072,
      "assistant_id": "asst_abc123",
      "thread_id": "thread_abc123",
      "status": "completed",
      "started_at": 1699075072,
      "expires_at": null,
      "cancelled_at": null,
      "failed_at": null,
      "completed_at": 1699075073,
      "last_error": null,
      "model": "gpt-3.5-turbo",
      "instructions": null,
      "tools": [
        {
          "type": "code_interpreter"
        }
      ],
      "file_ids": [
        "file-abc123",
        "file-abc456"
      ],
      "metadata": {},
      "usage": {
        "prompt_tokens": 123,
        "completion_tokens": 456,
        "total_tokens": 579
      }
    },
    {
      "id": "run_abc456",
      "object": "thread.run",
      "created_at": 1699063290,
      "assistant_id": "asst_abc123",
      "thread_id": "thread_abc123",
      "status": "completed",
      "started_at": 1699063290,
      "expires_at": null,
      "cancelled_at": null,
```

```
    "failed_at": null,
    "completed_at": 1699063291,
    "last_error": null,
    "model": "gpt-3.5-turbo",
    "instructions": null,
    "tools": [
     {
       "type": "code_interpreter"
     }
    ],
    "file_ids": [
     "file-abc123",
     "file-abc456"
    ],
    "metadata": {},
    "usage": {
     "prompt_tokens": 123,
     "completion_tokens": 456,
     "total_tokens": 579
    }
   }
  ],
  "first_id": "run_abc123",
  "last_id": "run_abc456",
  "has_more": false
}
```

List run steps Beta

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

Returns a list of run steps belonging to a run.

Path parameters

thread_id string Required
The ID of the thread the run and run steps belong to.

run_id string Required
The ID of the run the run steps belong to.

Query parameters

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

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

after string Optional

Example request node.js

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

async function main() {
  const runStep = await openai.beta.threads.runs.steps.list(
    "thread_abc123",
    "run_abc123"
  );
  console.log(runStep);
}

main();
```

Response

```
{
  "object": "list",
  "data": [
    {
      "id": "step_abc123",
      "object": "thread.run.step",
      "created_at": 1699063291,
      "run_id": "run_abc123",
      "assistant_id": "asst_abc123",
```

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

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

Returns

A list of **run step** objects.

```
10      "thread_id": "thread_abc123",
11      "type": "message_creation",
12      "status": "completed",
13      "cancelled_at": null,
14      "completed_at": 1699063291,
15      "expired_at": null,
16      "failed_at": null,
17      "last_error": null,
18      "step_details": {
19        "type": "message_creation",
20        "message_creation": {
21          "message_id": "msg_abc123"
22        }
23      },
24      "usage": {
25        "prompt_tokens": 123,
26        "completion_tokens": 456,
27        "total_tokens": 579
28      }
29    }
30  ],
31  "first_id": "step_abc123",
32  "last_id": "step_abc456",
33  "has_more": false
34 }
```

Retrieve run Beta

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

Retrieves a run.

Path parameters

thread_id string Required
The ID of the **thread** that was run.

run_id string Required
The ID of the run to retrieve.

Returns

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

Example request node.js

```
1  import OpenAI from "openai";
2
3  const openai = new OpenAI();
4
5  async function main() {
6    const run = await openai.beta.threads.runs.retrieve(
7      "thread_abc123",
8      "run_abc123"
9    );
10
11   console.log(run);
12 }
13
14 main();
```

Response

```
1  {
2    "id": "run_abc123",
3    "object": "thread.run",
4    "created_at": 1699075072,
5    "assistant_id": "asst_abc123",
6    "thread_id": "thread_abc123",
7    "status": "completed",
8    "started_at": 1699075072,
```

```
  "expires_at": null,
  "cancelled_at": null,
  "failed_at": null,
  "completed_at": 1699075073,
  "last_error": null,
  "model": "gpt-3.5-turbo",
  "instructions": null,
  "tools": [
    {
      "type": "code_interpreter"
    }
  ],
  "file_ids": [
    "file-abc123",
    "file-abc456"
  ],
  "metadata": {},
  "usage": {
    "prompt_tokens": 123,
    "completion_tokens": 456,
    "total_tokens": 579
  }
}
```

Retrieve run step **Beta**

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

Retrieves a run step.

Path parameters

thread_id string Required
The ID of the thread to which the run and run step belongs.

run_id string Required
The ID of the run to which the run step belongs.

step_id string Required
The ID of the run step to retrieve.

Returns

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

Example request node.js

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

async function main() {
  const runStep = await openai.beta.threads.runs.steps.retriev
    "thread_abc123",
    "run_abc123",
    "step_abc123"
  );
  console.log(runStep);
}

main();
```

Response

```
{
  "id": "step_abc123",
  "object": "thread.run.step",
  "created_at": 1699063291,
  "run_id": "run_abc123",
  "assistant_id": "asst_abc123",
  "thread_id": "thread_abc123",
  "type": "message_creation",
  "status": "completed",
  "cancelled_at": null,
  "completed_at": 1699063291,
```

```
  "expired_at": null,
  "failed_at": null,
  "last_error": null,
  "step_details": {
    "type": "message_creation",
    "message_creation": {
      "message_id": "msg_abc123"
    }
  },
  "usage": {
    "prompt_tokens": 123,
    "completion_tokens": 456,
    "total_tokens": 579
  }
}
```

Modify run **Beta**

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

Modifies a run.

Path parameters

thread_id string Required
The ID of the **thread** that was run.

run_id string Required
The ID of the run to modify.

Request body

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

Returns

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

Example request node.js

```
import OpenAI from "openai";

const openai = new OpenAI();

async function main() {
  const run = await openai.beta.threads.runs.update(
    "thread_abc123",
    "run_abc123",
    {
      metadata: {
        user_id: "user_abc123",
      },
    }
  );

  console.log(run);
}

main();
```

Response

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

```
  "model": "gpt-3.5-turbo",
  "instructions": null,
  "tools": [
    {
      "type": "code_interpreter"
    }
  ],
  "file_ids": [
    "file-abc123",
    "file-abc456"
  ],
  "metadata": {
    "user_id": "user_abc123"
  },
  "usage": {
    "prompt_tokens": 123,
    "completion_tokens": 456,
    "total_tokens": 579
  }
}
```

Submit tool outputs to run Beta

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

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

Path parameters

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

run_id string Required
The ID of the run that requires the tool output submission.

Request body

tool_outputs array Required
A list of tools for which the outputs are being submitted.

Show properties

Returns

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

Example request node.js

```
import OpenAI from "openai";

const openai = new OpenAI();

async function main() {
  const run = await openai.beta.threads.runs.submitToolOutpu
    "thread_abc123",
    "run_abc123",
    {
      tool_outputs: [
        {
          tool_call_id: "call_abc123",
          output: "28C",
        },
      ],
    }
  );

  console.log(run);
}

main();
```

Response

```
{
  "id": "run_abc123",
  "object": "thread.run",
  "created_at": 1699075592,
  "assistant_id": "asst_abc123",
```

```
  "thread_id": "thread_abc123",
  "status": "queued",
  "started_at": 1699075592,
  "expires_at": 1699076192,
  "cancelled_at": null,
  "failed_at": null,
  "completed_at": null,
  "last_error": null,
  "model": "gpt-4",
  "instructions": "You tell the weather.",
  "tools": [
   {
    "type": "function",
    "function": {
     "name": "get_weather",
     "description": "Determine weather in my location",
     "parameters": {
      "type": "object",
      "properties": {
       "location": {
        "type": "string",
        "description": "The city and state e.g. San Francisco,
       },
       "unit": {
        "type": "string",
        "enum": [
         "c",
         "f"
        ]
       }
      },
      "required": [
       "location"
      ]
     }
    }
   }
  ],
  "file_ids": [],
  "metadata": {},
  "usage": null
}
```

Cancel a run Beta

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

Cancels a run that is in_progress .

Path parameters

thread_id string Required
The ID of the thread to which this run belongs.

Example request node.js

```
import OpenAI from "openai";

const openai = new OpenAI();

async function main() {
 const run = await openai.beta.threads.runs.cancel(
  "thread_abc123",
  "run_abc123"
 );
```

run_id string Required
The ID of the run to cancel.

Returns

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

```
10
11    console.log(run);
12  }
13
14  main();
```

Response

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

The run object Beta

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

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

object string
The object type, which is always thread.run .

created_at integer
The Unix timestamp (in seconds) for when the run was created.

thread_id string
The ID of the **thread** that was executed on as a part of this run.

assistant_id string
The ID of the **assistant** used for execution of this run.

status string

The run object

```
{
  "id": "run_abc123",
  "object": "thread.run",
  "created_at": 1698107661,
  "assistant_id": "asst_abc123",
  "thread_id": "thread_abc123",
  "status": "completed",
  "started_at": 1699073476,
  "expires_at": null,
  "cancelled_at": null,
  "failed_at": null,
  "completed_at": 1699073498,
  "last_error": null,
  "model": "gpt-4",
  "instructions": null,
  "tools": [{"type": "retrieval"}, {"type": "code_interpreter"}],
  "file_ids": [],
  "metadata": {},
```

The status of the run, which can be either `queued`, `in_progress`, `requires_action`, `cancelling`, `cancelled`, `failed`, `completed`, or `expired`.

required_action object or null
Details on the action required to continue the run. Will be `null` if no action is required.
Show properties

last_error object or null
The last error associated with this run. Will be `null` if there are no errors.
Show properties

expires_at integer
The Unix timestamp (in seconds) for when the run will expire.

started_at integer or null
The Unix timestamp (in seconds) for when the run was started.

cancelled_at integer or null
The Unix timestamp (in seconds) for when the run was cancelled.

failed_at integer or null
The Unix timestamp (in seconds) for when the run failed.

completed_at integer or null
The Unix timestamp (in seconds) for when the run was completed.

model string
The model that the **assistant** used for this run.

instructions string
The instructions that the **assistant** used for this run.

tools array
The list of tools that the **assistant** used for this run.
Show possible types

file_ids array
The list of **File** IDs the **assistant** used for this run.

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

usage object or null
Usage statistics related to the run. This value will be `null` if the run is not in a terminal state (i.e. `in_progress`, `queued`, etc.).

```
19     "usage": {
20       "prompt_tokens": 123,
21       "completion_tokens": 456,
22       "total_tokens": 579
23     }
24   }
```

Show properties

The run step object Beta

Represents a step in execution of a run.

id string
The identifier of the run step, which can be referenced in API endpoints.

object string
The object type, which is always `thread.run.step`.

created_at integer
The Unix timestamp (in seconds) for when the run step was created.

assistant_id string
The ID of the **assistant** associated with the run step.

thread_id string
The ID of the **thread** that was run.

run_id string
The ID of the **run** that this run step is a part of.

type string
The type of run step, which can be either `message_creation` or `tool_calls`.

status string
The status of the run step, which can be either `in_progress`, `cancelled`, `failed`, `completed`, or `expired`.

step_details object
The details of the run step.
Show possible types

last_error object or null
The last error associated with this run step. Will be `null` if there are no errors.
Show properties

expired_at integer or null
The Unix timestamp (in seconds) for when the run step expired. A step is considered expired if the parent run is expired.

cancelled_at integer or null
The Unix timestamp (in seconds) for when the run step was cancelled.

failed_at integer or null

The run step object


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

The Unix timestamp (in seconds) for when the run step failed.

completed_at integer or null
The Unix timestamp (in seconds) for when the run step completed.

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

usage object or null
Usage statistics related to the run step. This value will be `null` while the run step's status is `in_progress`.
Show properties