OpenAI Platform

Migration Guide Beta

We have changed the way that tools and files work in the Assistants API between the `v1` and `v2` versions of the beta. Both versions of the beta continue to be accessible via the API today, but we recommend migrating to the newest version of our APIs as soon as feasible. We will deprecate `v1` of the beta by the end of 2024.

If you do not use tools or files with the Assistants API today, there should be no changes required for you to migrate from the `v1` version to the `v2` version of the beta. Simply pass the `v2` beta version header and/or move to the latest version of our Node and Python SDKs!

What has changed

The `v2` version of the Assistants API contains the following changes:

1. **Tool rename:** The `retrieval` tool has been renamed to the `file_search` tool
2. **Files belong to tools:** Files are now associated with tools instead of Assistants and Messages. This means that:
    - `AssistantFile` and `MessageFile` objects no longer exist.
    - Instead of `AssistantFile` and `MessageFile`, files are attached to Assistants and **Threads** using the new `tool_resources` object.
        - The `tool_resources` for the code interpreter tool are a list of `file_ids`.
        - The `tool_resources` for the `file_search` tool are a new object called a `vector_stores`.
    - Messages now have an `attachments`, rather than a `file_ids` parameter. Message attachments are helpers that add the files to a Thread's `tool_resources`.

V1 Assistant

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

V2 Assistant

```
 {
   "id": "asst_abc123",
   "object": "assistant",
   "created_at": 1698984975,
   "name": "Math Tutor",
   "description": null,
   "model": "gpt-4-turbo",
   "instructions": "You are a personal math tutor. When asked a question, write
+  "tools": [
+    {
+      "type": "code_interpreter"
+    },
+    {
+      "type": "file_search"
+    }
+  ],
+  "tool_resources": {
+    "file_search": {
+      "vector_store_ids": ["vs_abc"]
+    },
+    "code_interpreter": {
+      "file_ids": ["file-123", "file-456"]
+    }
+  }
 }
```

Assistants have `tools` and `tool_resources` instead of `file_ids`. The `retrieval` tool is now the `file_search` tool. The `tool_resource` for the `file_search` tool is a `vector_store`.

V1 Thread

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

V2 Thread

```
  {
    "id": "thread_abc123",
    "object": "thread",
    "created_at": 1699012949,
+   "metadata": {},
+   "tools": [
+     {
+       "type": "file_search"
+     },
+     {
+       "type": "code_interpreter"
+     }
+   ],
+   "tool_resources": {
+     "file_search": {
+       "vector_store_ids": ["vs_abc"]
+     },
+     "code_interpreter": {
+       "file_ids": ["file-123", "file-456"]
+     }
+   }
  }
```

Threads can bring their own `tool_resources` into a conversation.

V1 Message

```
{
  "id": "msg_abc123",
  "object": "thread.message",
  "created_at": 1698983503,
  "thread_id": "thread_abc123",
  "role": "assistant",
  "content": [
    {
      "type": "text",
      "text": {
        "value": "Hi! How can I help you today?",
        "annotations": []
      }
    }
  ],
  "assistant_id": "asst_abc123",
  "run_id": "run_abc123",
  "metadata": {},
-   "file_ids": []
- }
```

V2 Message

```
  {
    "id": "msg_abc123",
    "object": "thread.message",
    "created_at": 1698983503,
    "thread_id": "thread_abc123",
    "role": "assistant",
    "content": [
      {
        "type": "text",
        "text": {
          "value": "Hi! How can I help you today?",
          "annotations": []
        }
      }
    ],
    "assistant_id": "asst_abc123",
    "run_id": "run_abc123",
    "metadata": {},
+   "attachments": [
+     {
+       "file_id": "file-123",
+       "tools": [
+         { "type": "file_search" },
+         { "type": "code_interpreter" }
+       ]
+     }
+   ]
  }
```

Messages have `attachments` instead of `file_ids`. `attachments` are helpers that add files to the Thread’s `tool_resources`.

All `v1` endpoints and objects for the Assistants API can be found under the Legacy section of the API reference.

Accessing v1 data in v2

To make your migration simple between our `v1` and `v2` APIs, we automatically map `AssistantFiles` and `MessageFiles` to the appropriate `tool_resources` based on the tools that are enabled in Assistants or Runs these files are a part of.

	V1 VERSION	V2 VERSION
AssistantFiles for `code_interpreter`	`file_ids` on Assistant	Files in an Assistant's `tool_resources.code_interpreter`
AssistantFiles for `retrieval`	`file_ids` on Assistant	Files in a vector_store attached to an Assistant (`tool_resources.file_search`)
MessageFiles for `code_interpreter`	`file_ids` on Message	Files in an Thread's `tool_resources.code_interpreter`
MessageFiles for `retrieval`	`file_ids` on Message	Files in a vector_store attached to a Thread (`tool_resources.file_search`)

It's important to note that while `file_ids` from `v1` are mapped to `tool_resources` in `v2`, the inverse is not true. Changes you make to `tool_resources` in `v2` will not be reflected as `file_ids` in `v1`.

Because Assistant Files and Message Files are already mapped to the appropriate `tool_resources` in `v2`, when you're ready to migrate to `v2` you shouldn't have to worry about a data migration. Instead, you only need to:

1. Update your integration to reflect the new API and objects. You may need to do things like:
   - Migrate to creating `vector_stores` and using `file_search`, if you were using the `retrieval` tool. Importantly, since these operations are asynchronous, you'll want to ensure files are successfully ingested by the `vector_stores` before creating run.
   - Migrate to adding files to `tool_resources.code_interpreter` instead of an Assistant or Message's files, if you were using the `code_interpreter` tool.
   - Migrate to using Message `attachments` instead of `file_ids`.
2. Upgrade to the latest version of our SDKs

Changing beta versions

Without SDKs

Both beta versions can be accessed by passing the right API version header in your API requests:

1. `v1`: `OpenAI-Beta: assistants=v1`
2. `v2`: `OpenAI-Beta: assistants=v2`

v2

```
curl "https://api.openai.com/v1/assistants" \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "OpenAI-Beta: assistants=v2" \
  -d '{
    "instructions": "You are a personal math tutor. When asked a question, write
    "name": "Math Tutor",
    "tools": [{"type": "code_interpreter"}],
    "model": "gpt-4-turbo"
  }'
```

With SDKs

Versions of our SDKs that are released after the release of the `v2` beta will have the `openai.beta` namespace point to the `v2` version of the API by default. You can still access the `v1` version of the API by using an older version of the SDK (1.20.0 or earlier for python, 4.36.0 or earlier for node) or by overriding the version header.

To install an older version of the SDK, you can use the following commands:

Installing older versions of the SDK python

```
pip install openai==1.20.0
```

You can also override this header in a newer SDK version, but we don't recommend this approach since the object types in these newer SDK versions will be different from the `v1` objects.

Accessing the `v1` API version in new SDKs python

```python
from openai import OpenAI

client = OpenAI(default_headers={"OpenAI-Beta": "assistants=v1"})
```

Billing

All vector stores created before the release of the `v2` API (April 17, 2024) will be free to use until the end of 2024. This implies that any vector stores that were created as a result of us mapping your `v1` data to `v2`, before the `v2` launch will be free. After the end of 2024, they'll be billed at whatever the fees for vector stores are at that point. See our pricing page for the latest pricing information.

Any vector store that is created before the release of the `v2` API (April 17, 2024) but not used in a single Run between that release date and the end of 2024 will be deleted. This is to avoid us starting to bill you for something you created during the beta but never used.

Vector stores created after the release of the `v2` API will be billed at current rates as specified on the pricing page.

Deleting files

Deleting Assistant Files / Message Files via the `v1` API also removes them from the `v2` API. However, the inverse is not true - deletions in the `v2` version of the API do not propogate to `v1`. If you created a file on `v1` and would like to "fully" delete a file from your account on both `v1` and `v2` you should:

- delete Assistant Files / Message Files you create using `v1` APIs using the `v1` endpoints, or
- delete the underlying file object — this ensures it is fully removed from all objects in all versions of the API.

Playground

The default playground experience has been migrated to use the `v2` version of the API (you will still have a read-only view of the `v1` version of objects, but will not be able to edit them). Any changes you make to tools and files via the Playground will only be accessible in the `v2` version of the API.

In order to make changes to files in the `v1` version of the API, you will need to use the API directly.