Tutorial for Azure OpenAI Service

Index

1. Use Azure AI Studio 2

1.1 Create a Project 2

Step 1: Click the “New project” button to create a project. 2

Step 2: Fill in the project name, and then click “Next” 3

Step 3: Create or select a “hub” for this project. You can skip this step by directly click “Next” button. 3

Step 4: Review and finish the creation of this project. Click the “Create a project” button. 4

Result: Home page of the newly created project. After the above steps, you will be redirected to the home page of that project. 4

1.2 Deploy a Model 5

Step 1: Click “Chat” tab in the menu of the project home page. 5

Step 2: Click “Create a deployment” button. 5

Step 3: Select a model to deploy. 6

Step 4: Click “Deploy” button. 6

Result: Now you can chat with GPT. 7

1.3 Manage Quota 8

1.3.1 Manage quota of your projects. 8

1.3.2 Request Quota 9

2. Use Python API 12

Step 1. Get endpoint, key and model version. 13

Step 2. Install Python package for OpenAI 14

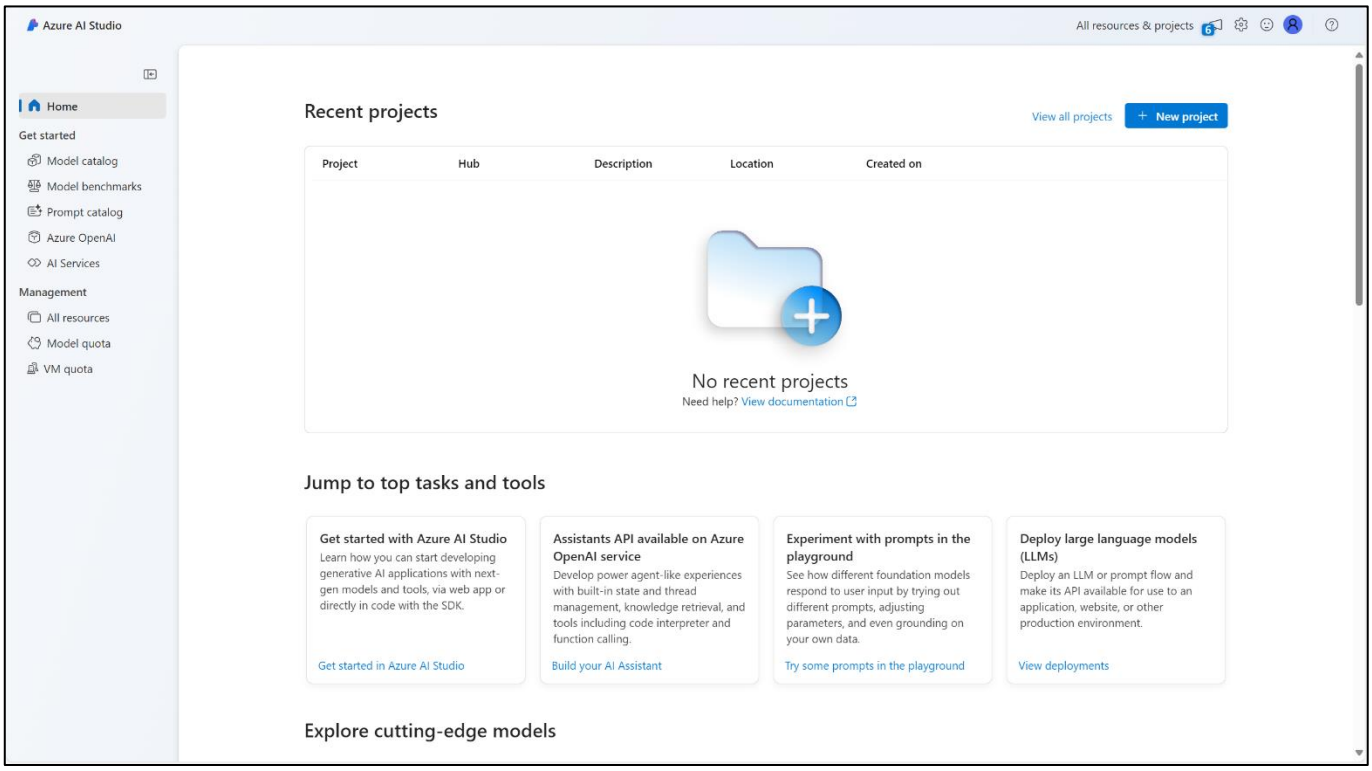
Step 3. Configure the Python code for Azure OpenAI. 14

1. Use Azure AI Studio

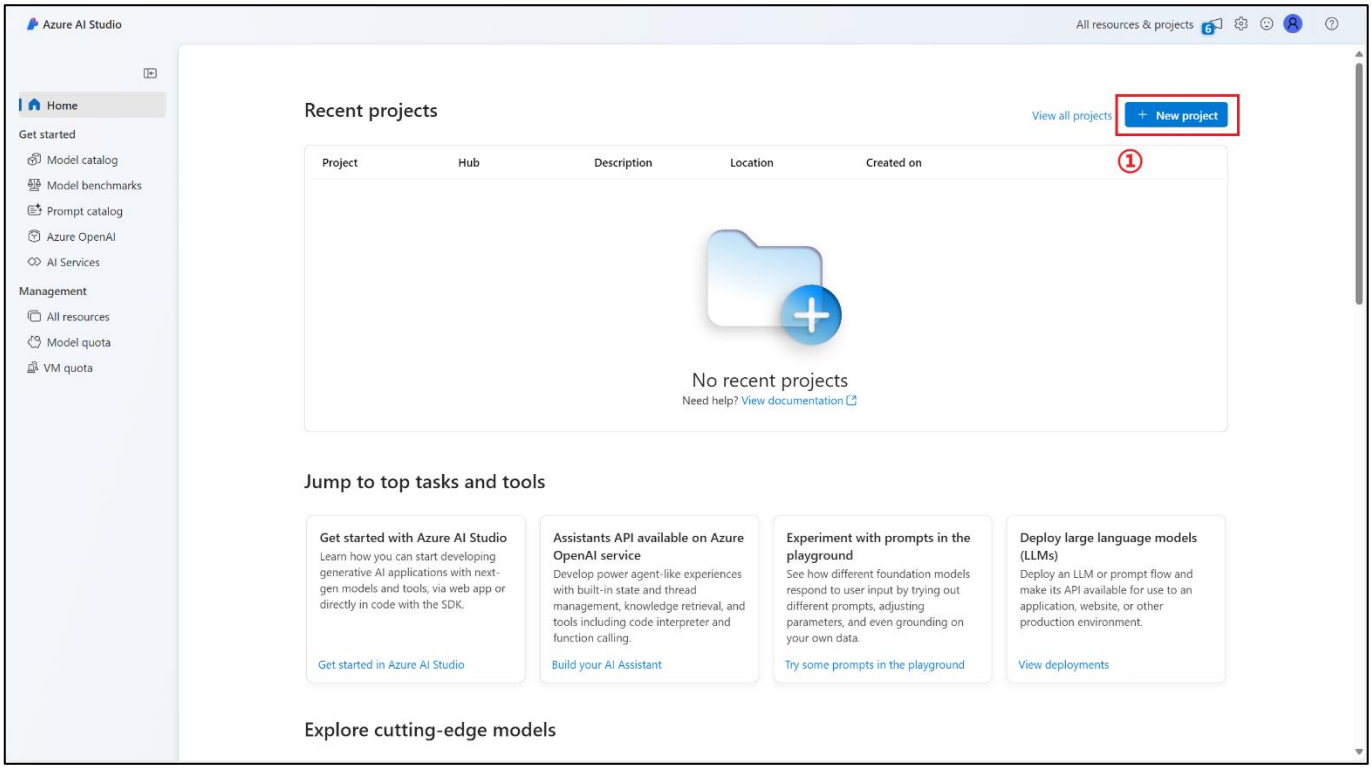
Azure AI Studio (https://ai.azure.com/) is a convenient portal site for managing Azure AI resources. In this chapter, we will introduce the necessary steps to set up a GPT model in Azure AI Studio. This chapter consists of the following operations: (1) creating a project; (2) deploying a model; (3) managing quota of projects.

1.1 Create a Project

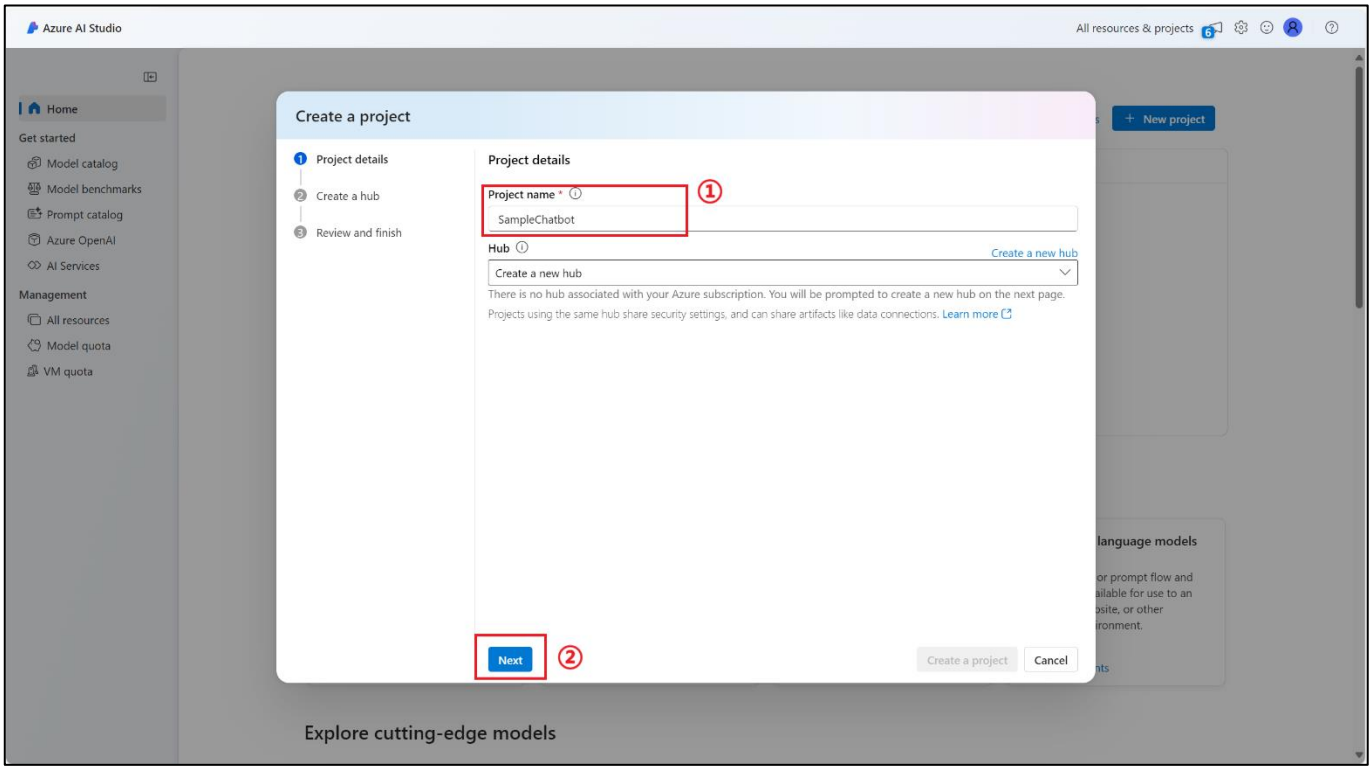
After logging into your UTAC account in Azure AI Studio, the Studio will look like this:



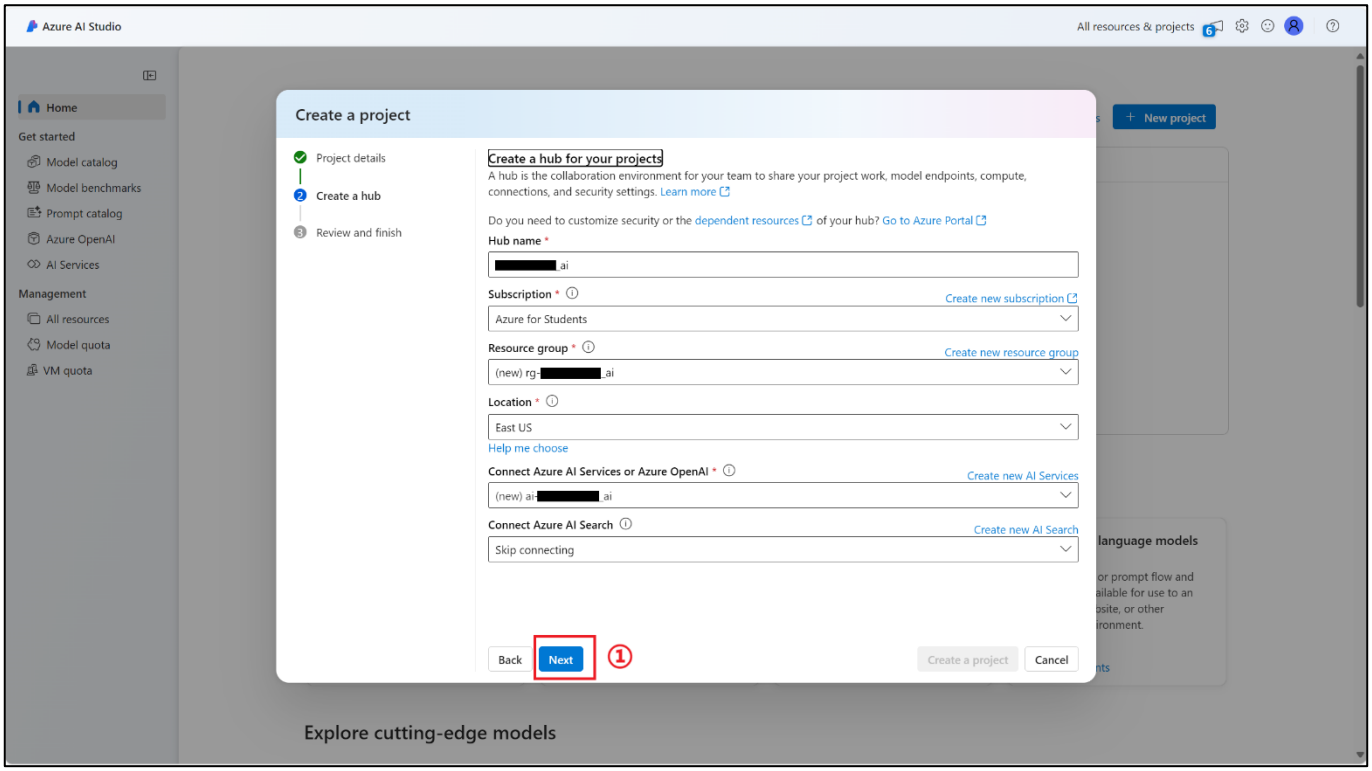
Step 1: Click the “New project” button to create a project.



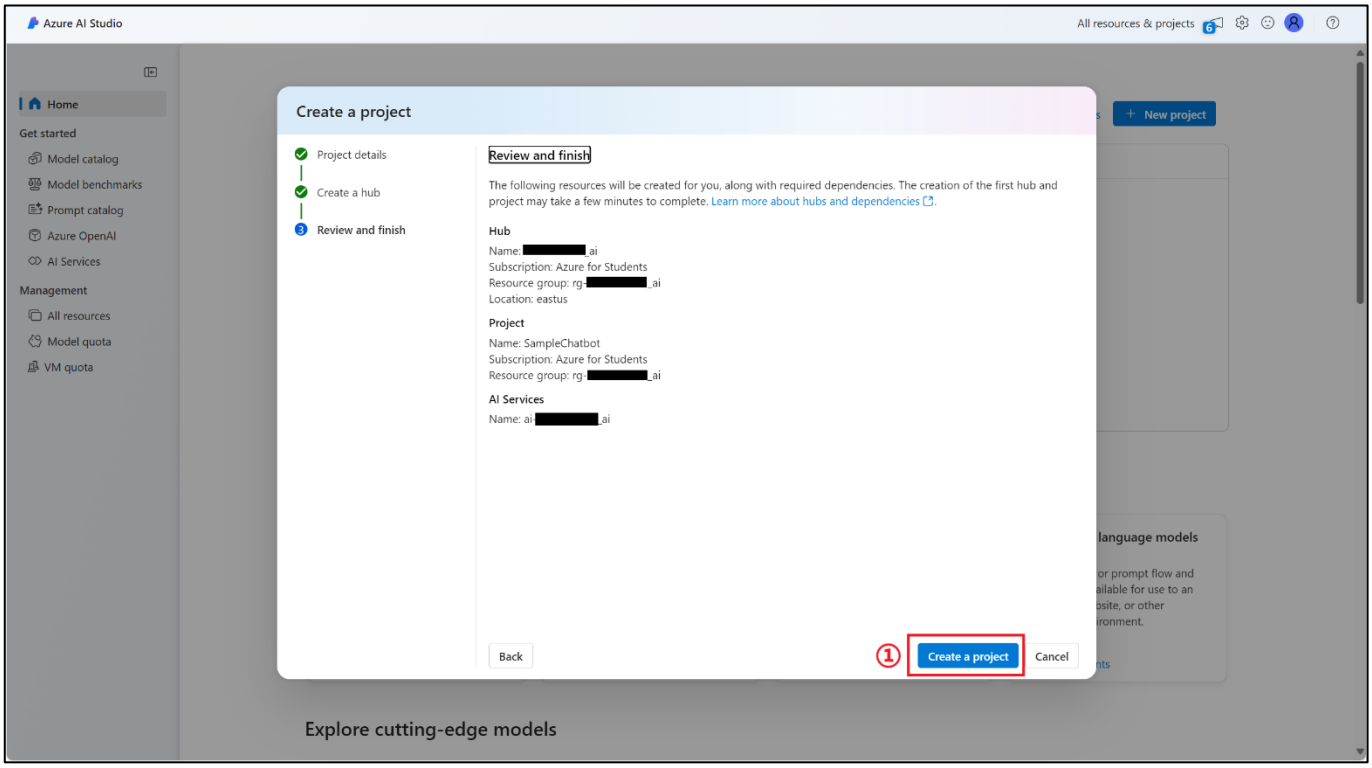
Step 2: Fill in the project name, and then click “Next”.



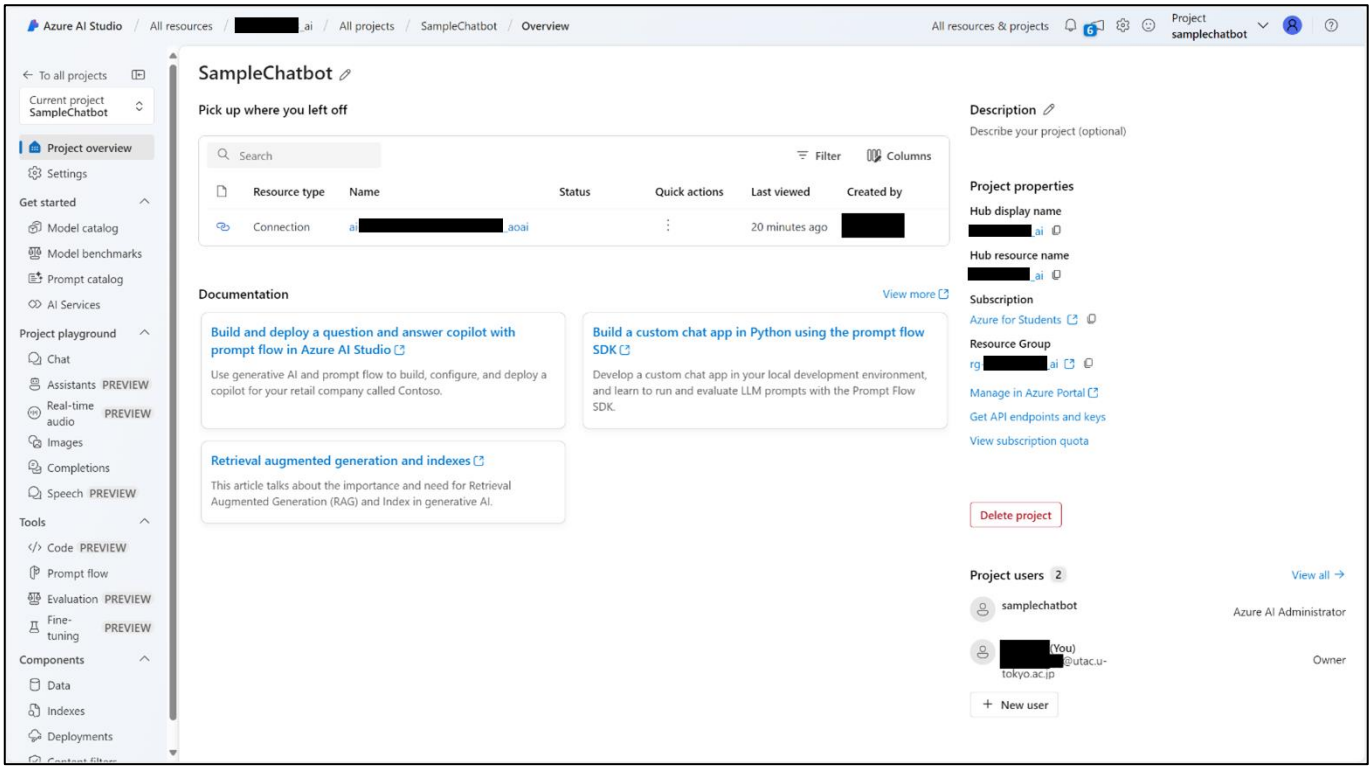
Step 3: Create or select a “hub” for this project. You can skip this step by directly click “Next” button.



Step 4: Review and finish the creation of this project. Click the “Create a project” button.



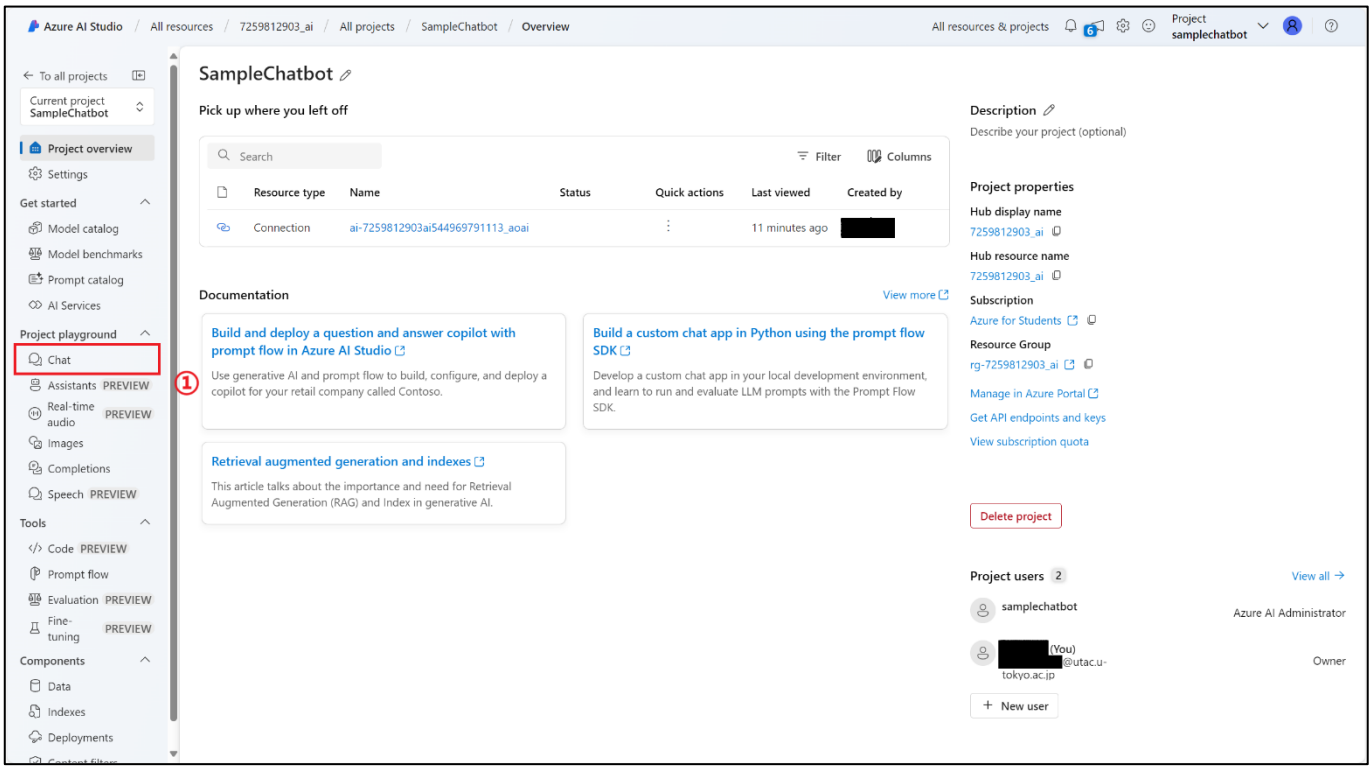
Result: Home page of the newly created project. After the above steps, you will be redirected to the home page of that project.



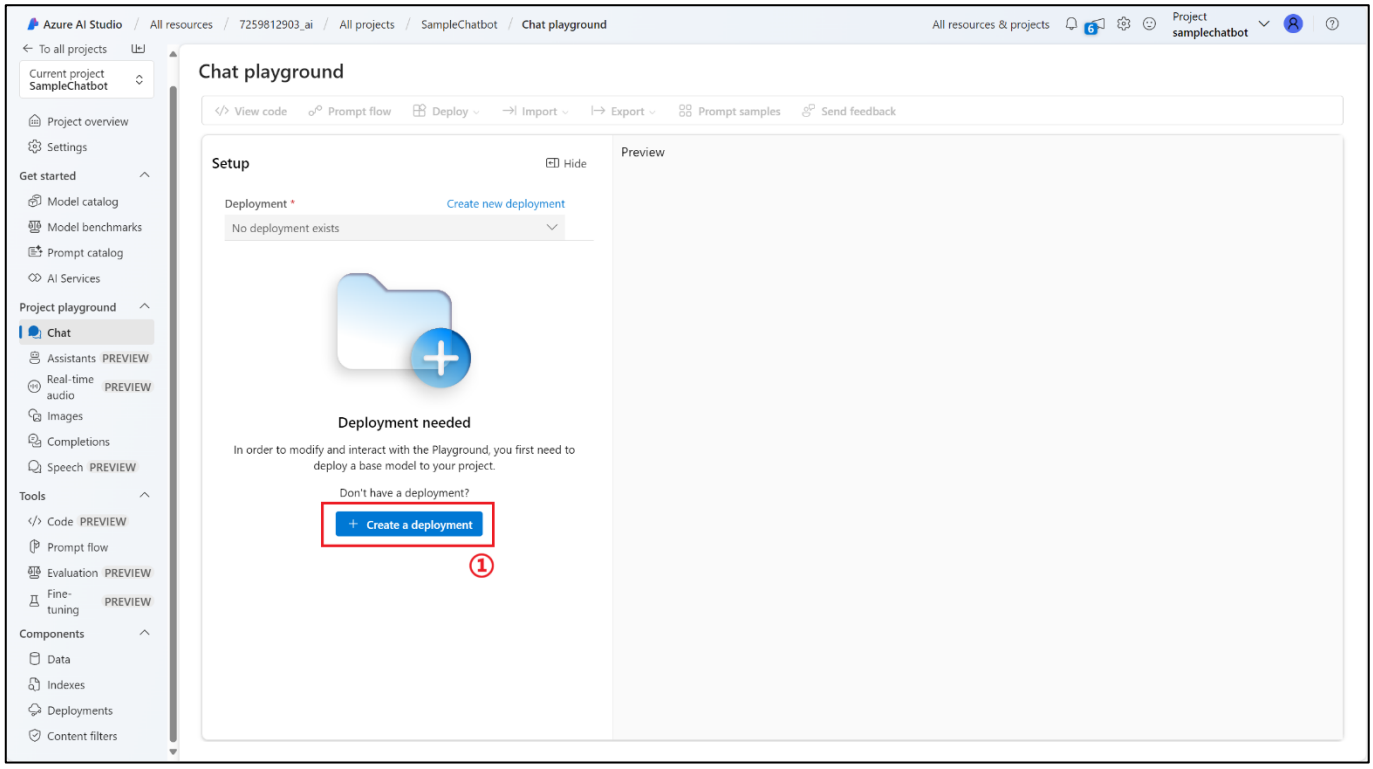
This home page is like OpenAI Playground, you can try OpenAI services, manage fine-tuning tasks, etc.

1.2 Deploy a Model

Step 1: Click “Chat” tab in the menu of the project home page.

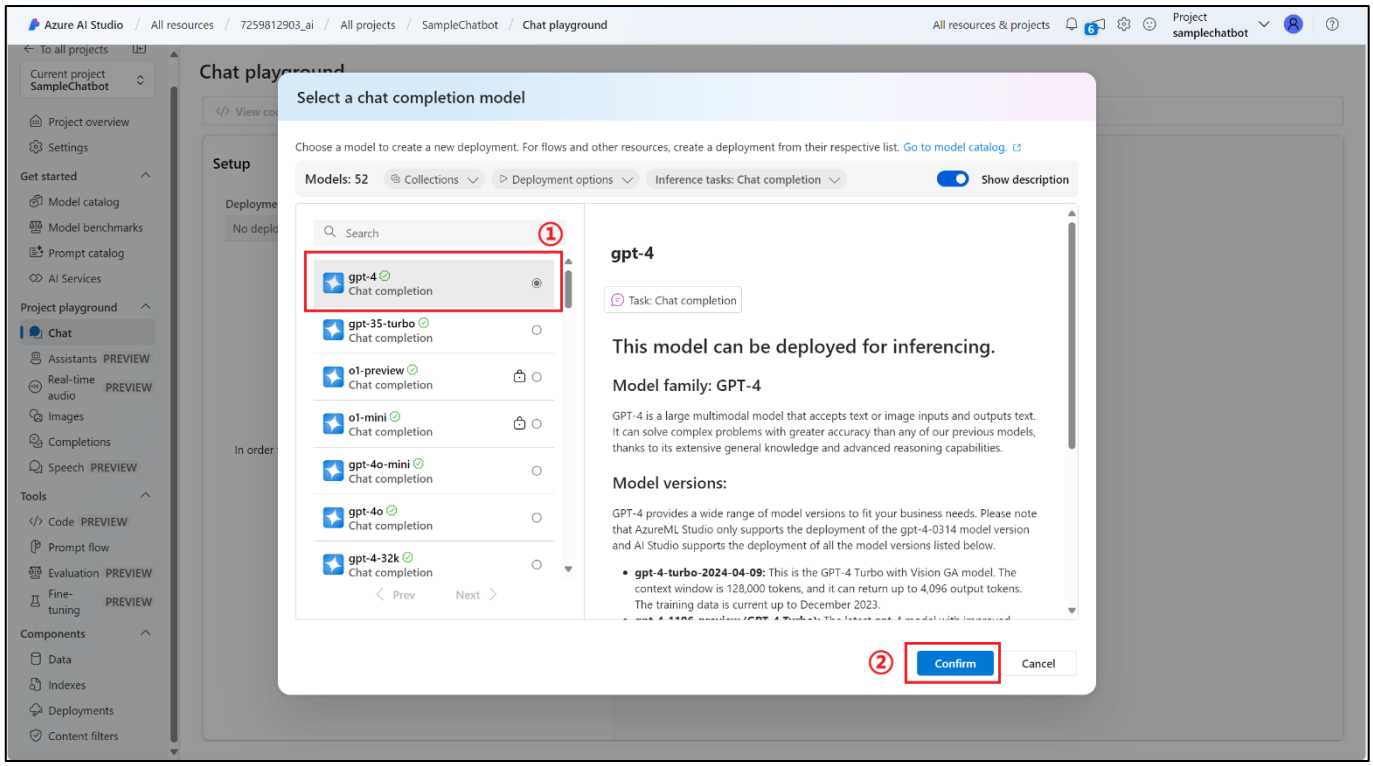


Step 2: Click “Create a deployment” button.

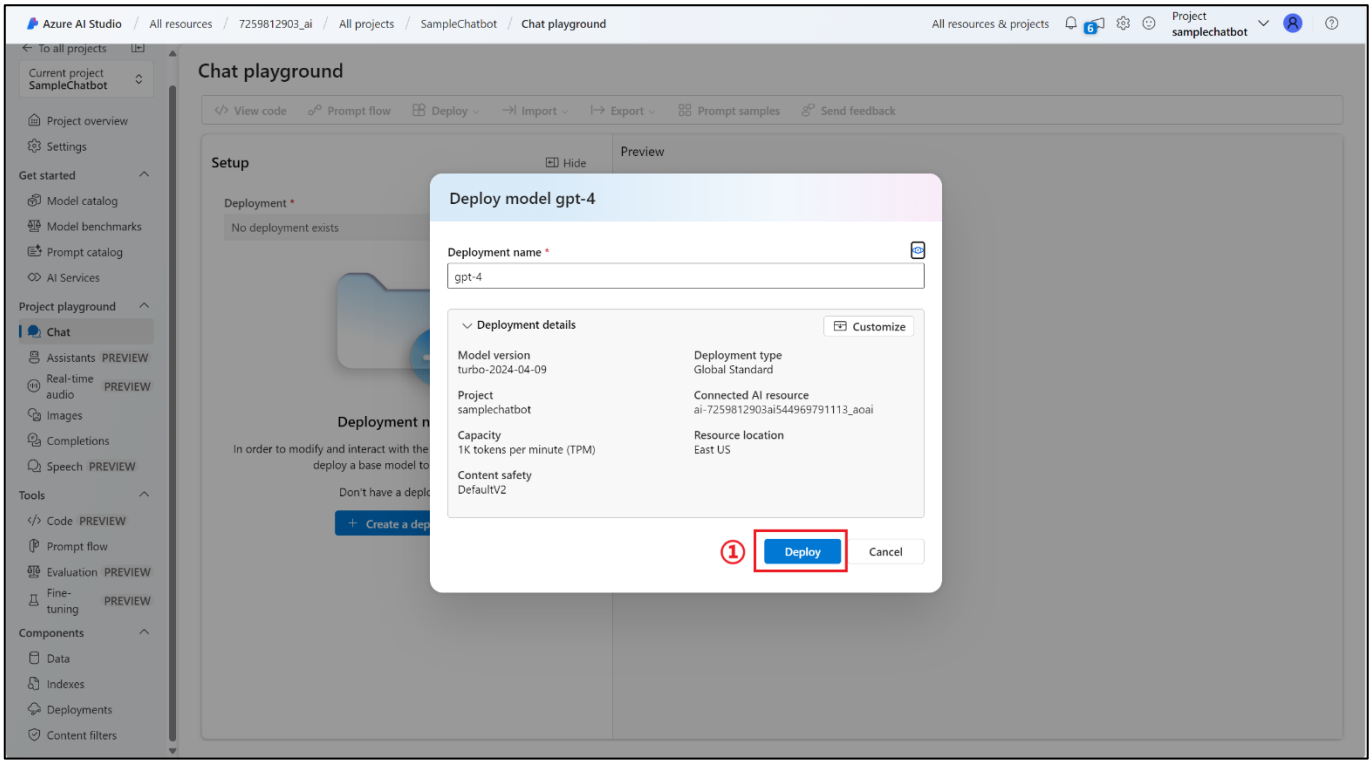


Step 3: Select a model to deploy.

In addition to the GPT models, Azure also provides many other models covering tasks such as translation.

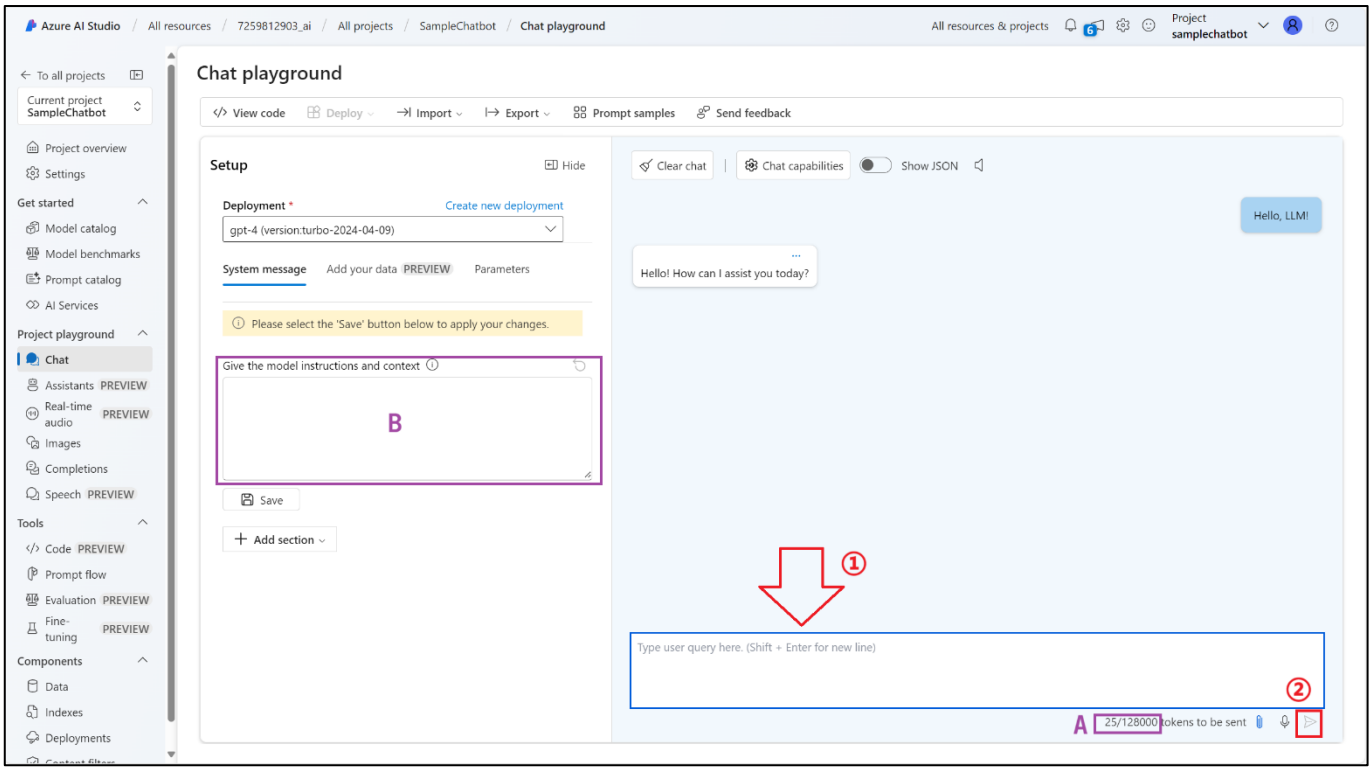


Step 4: Click “Deploy” button.



Notice: You must wait for about 5 minutes before the model is ready for use.

Result: Now you can chat with GPT.



Type your message in textbox ① and click the triangle ② (or press “Enter” key) to send it to the model. Label A represents the token of this message and the token limit for this model. In this picture, sending this message uses 25 tokens, and the overall token limit is 128000 tokens. You can enter the “system prompt” (or instructions for GPT) in textbox B.

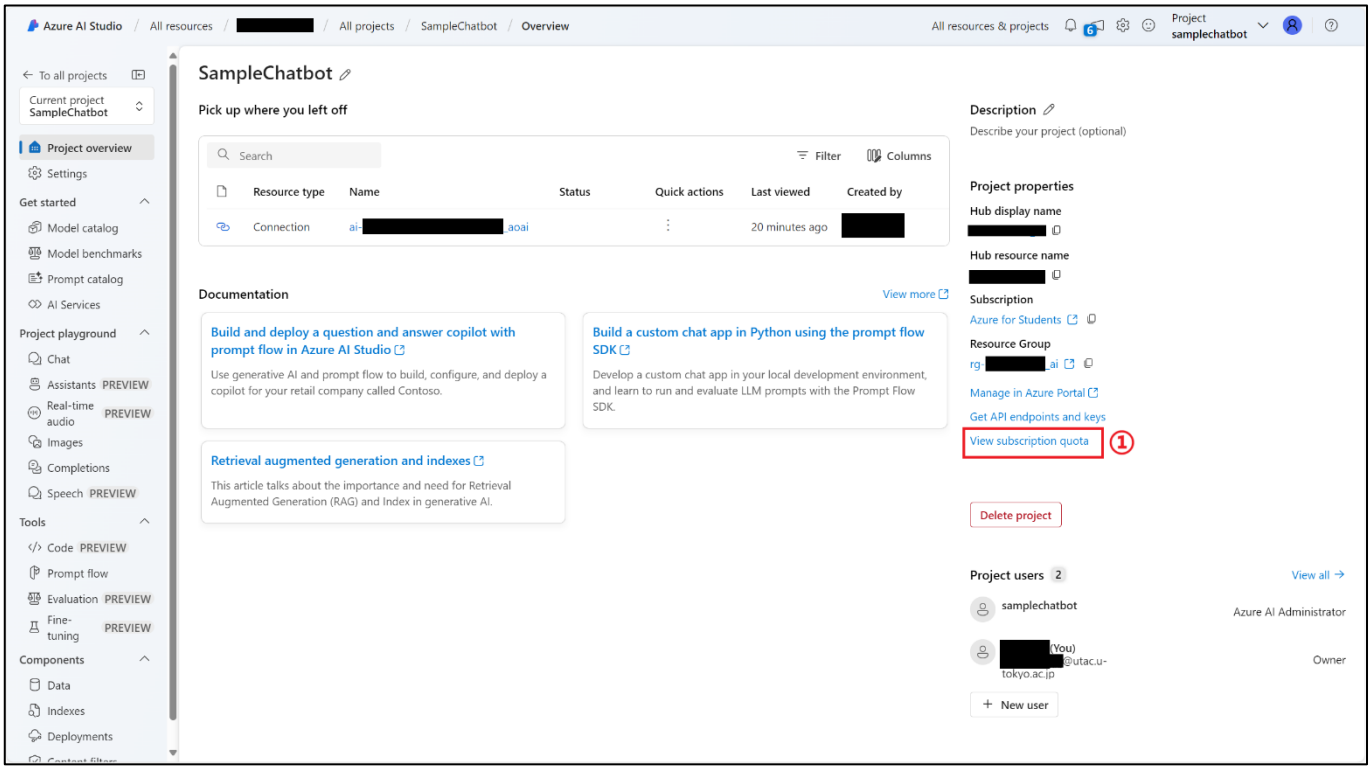
Notice: Every time you send a message to GPT, the whole message history is sent along with the new message. That's the reason why there's no text in the message in this screenshot, but it still costs 25 tokens. Token consumption is calculated on a cumulative basis, for example, if you have messages and replies consisting of correspondingly 2, 5, 1, 7, 11, 1 token(s), then when you send a new message consisting of 9 tokens, this request to send actually consists of 2+5+1+7+11+1+9=36 tokens; so far, you have consumed (2) + (2+5) + (2+5+1) + (2+5+1+7) + (2+5+1+7+11) + (2+5+1+7+11+1) + (2+5+1+7+11+1+9)=121 tokens instead of 2+5+1+7+11+1+9=36 tokens. Unlike the ChatGPT application provided by OpenAI, Azure OpenAI Service is charged by the token, so please pay attention to how the tokens are consumed.

1.3 Manage Quota

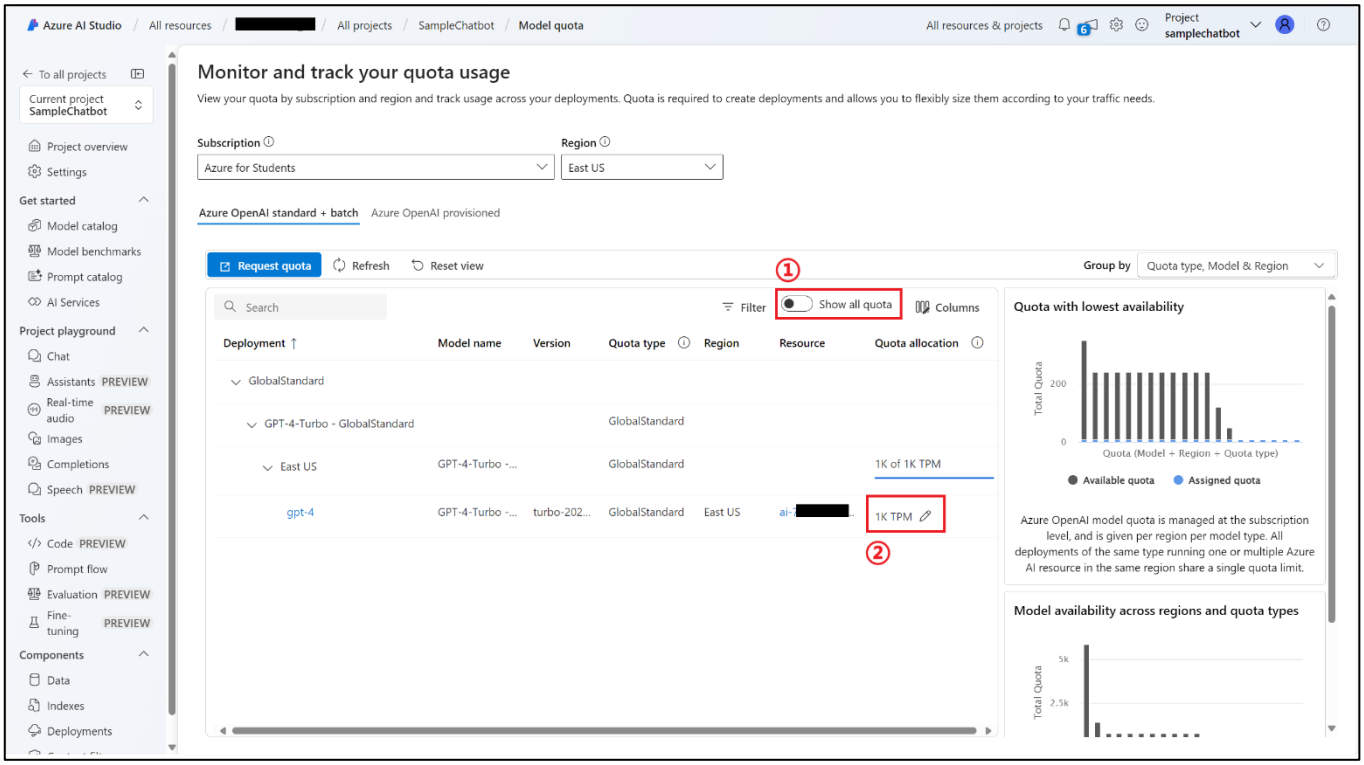
"Quota" is the maximum number of tokens that these models can process in a short period of time, and it represents the rate limit in the Azure OpenAI Service. You may experience errors if you exceed this limit, at which point, you can wait for the tokens per minute to "cool down", or request an increase in the quota for your current project.

1.3.1 Manage quota of your projects.

By clicking "View subscription quota" link on the project home page, you can check the quota for projects. You can also find this link on the home page of Azure AI Studio.

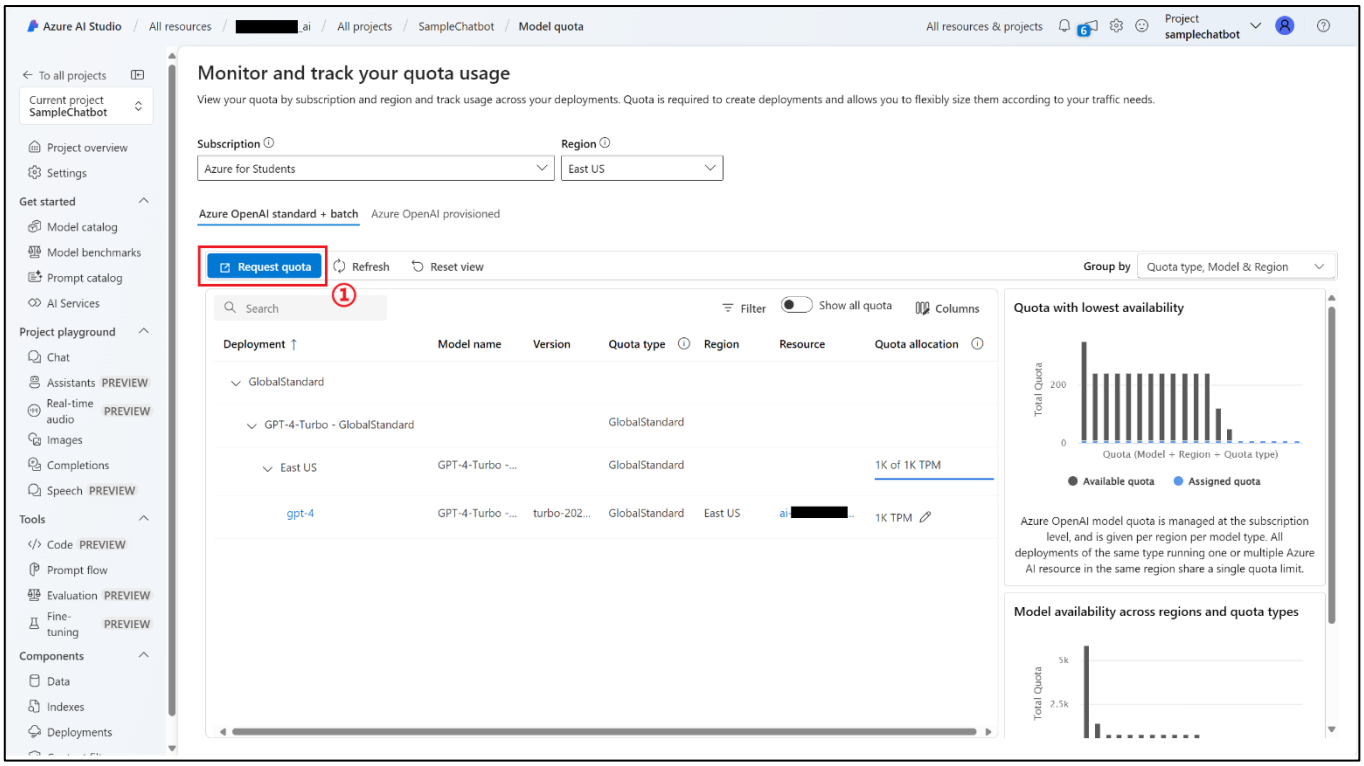


By default, the quota usage view shows the quota of all models in all geographical locations, even if you are not using those models in those locations. Set the "Show all quota" switch to off to only display the models in use. You can then click on the pencil icon to change the quota of these models.



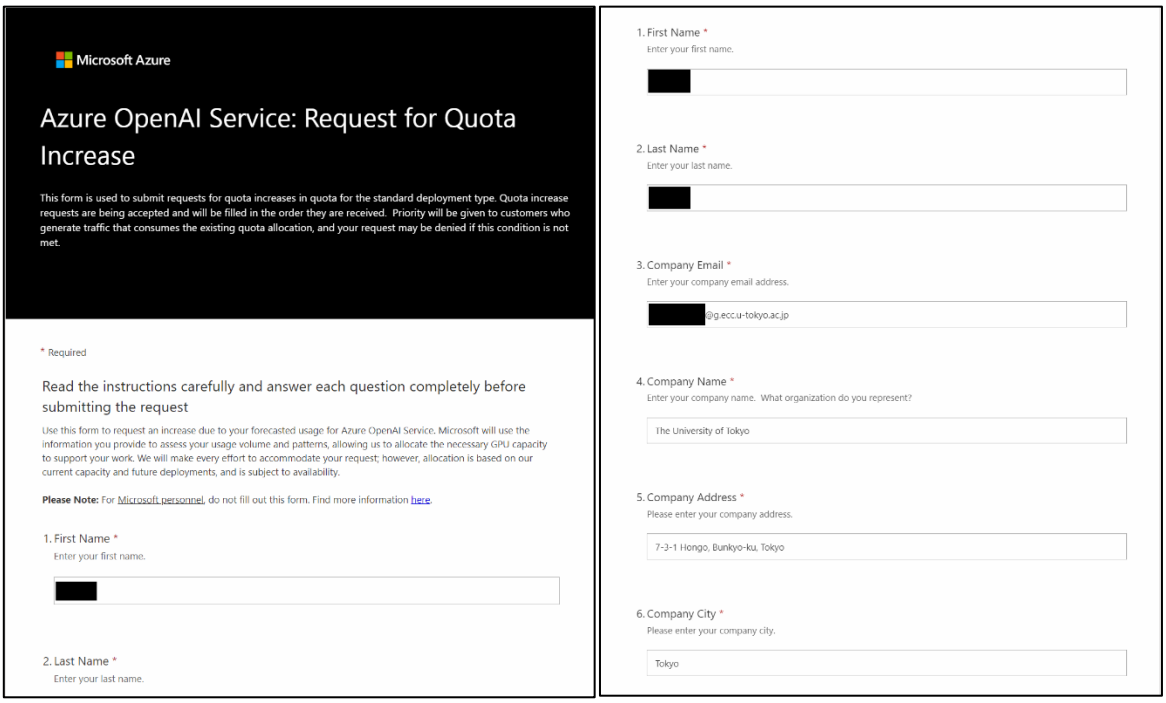
1.3.2 Request Quota

By clicking “Request quota” button, you can request Microsoft Azure Team to increase the quota of your subscription.



Notice: Forms are reviewed manually by Azure Team, so this process may take several days. There is no guarantee that requests will always be approved. Please ensure that the information you provide is accurate and valid. Sometimes the team may send you emails asking for additional information or documents, so please check your emails frequently.

The form looks like this. Fill in your details and those of The University of Tokyo as instructed.



Microsoft Azure

Azure OpenAI Service: Request for Quota Increase

This form is used to submit requests for quota increases in quota for the standard deployment type. Quota increase requests are being accepted and will be filled in the order they are received. Priority will be given to customers who generate traffic that consumes the existing quota allocation, and your request may be denied if this condition is not met.

* Required

Read the instructions carefully and answer each question completely before submitting the request

Use this form to request an increase due to your forecasted usage for Azure OpenAI Service. Microsoft will use the information you provide to assess your usage volume and patterns, allowing us to allocate the necessary GPU capacity to support your work. We will make every effort to accommodate your request; however, allocation is based on our current capacity and future deployments, and is subject to availability.

Please Note: For Microsoft personnel, do not fill out this form. Find more information here.

1. First Name *
Enter your first name.
[redacted]

2. Last Name *
Enter your last name.

1. First Name *
Enter your first name.
[redacted]

2. Last Name *
Enter your last name.
[redacted]

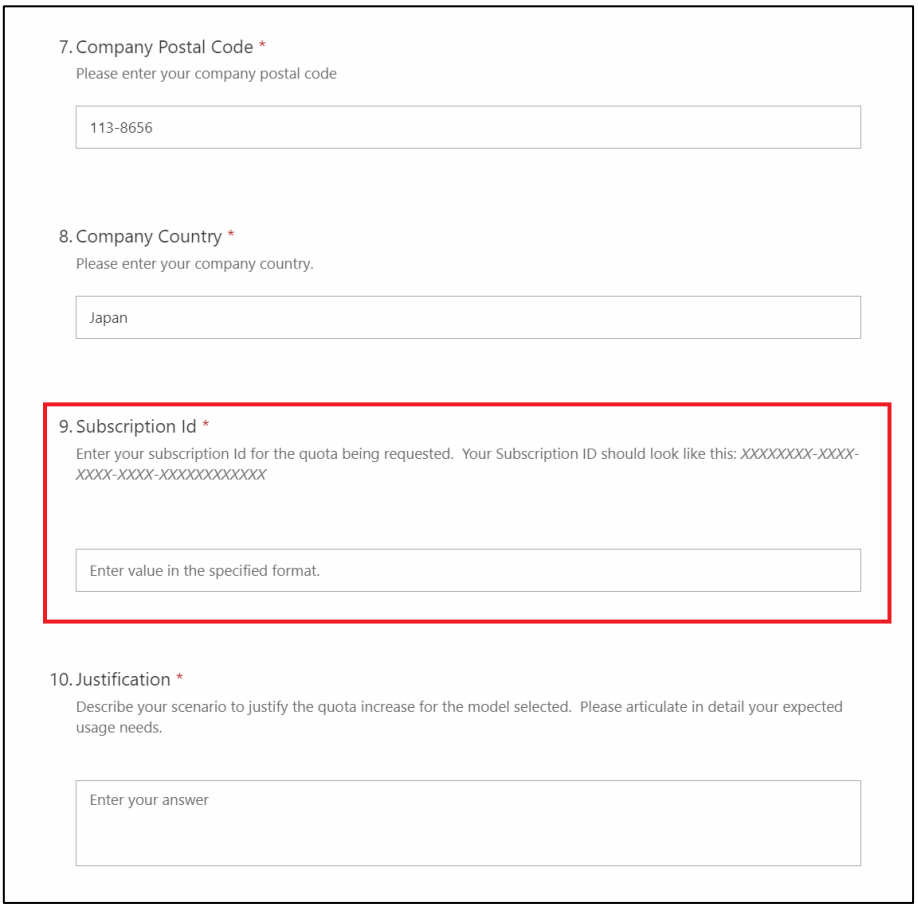
3. Company Email *
Enter your company email address.
[redacted]@g.ecc.u-tokyo.ac.jp

4. Company Name *
Enter your company name. What organization do you represent?
The University of Tokyo

5. Company Address *
Please enter your company address.
7-3-1 Hongo, Bunkyo-ku, Tokyo

6. Company City *
Please enter your company city.
Tokyo

In the form, there is a field named “Subscription Id” as follows:



7. Company Postal Code *
Please enter your company postal code
113-8656

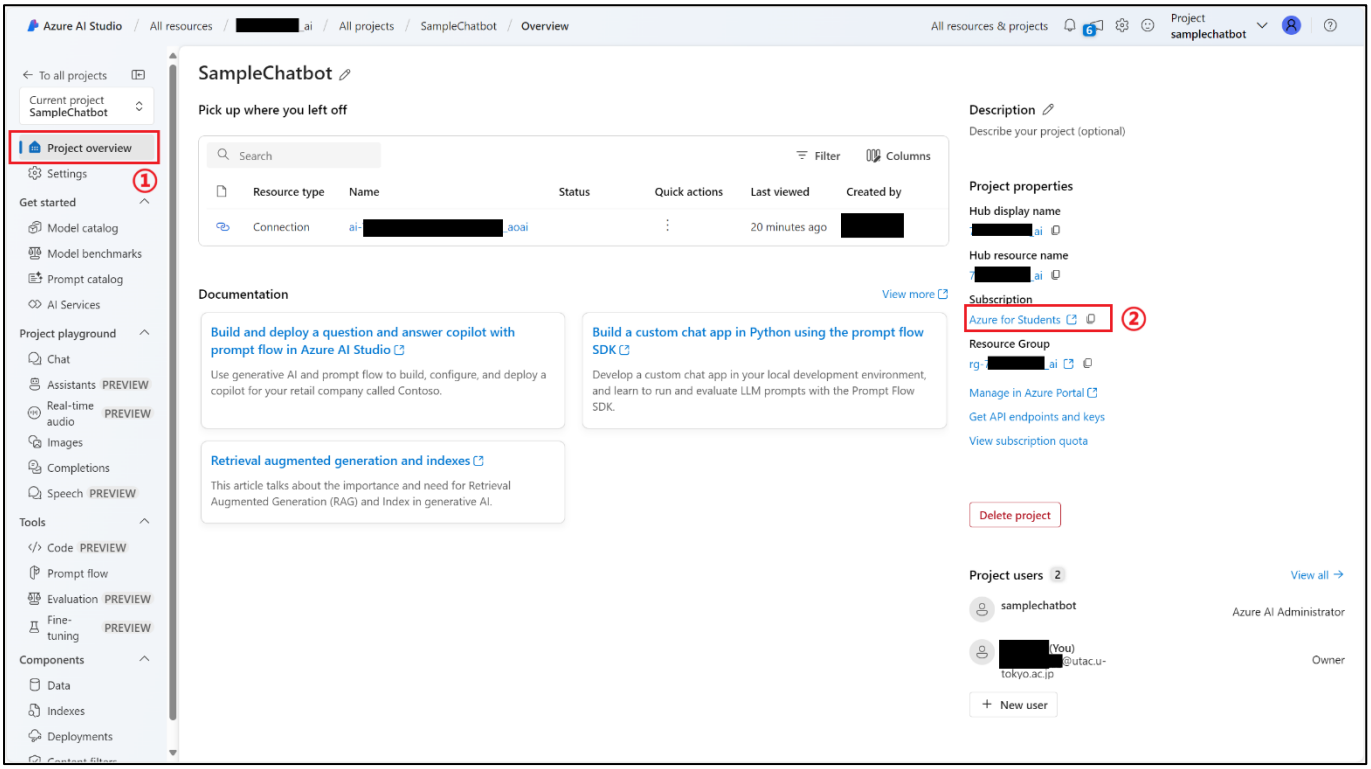
8. Company Country *
Please enter your company country.
Japan

9. Subscription Id *
Enter your subscription Id for the quota being requested. Your Subscription ID should look like this: *XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX*
Enter value in the specified format.

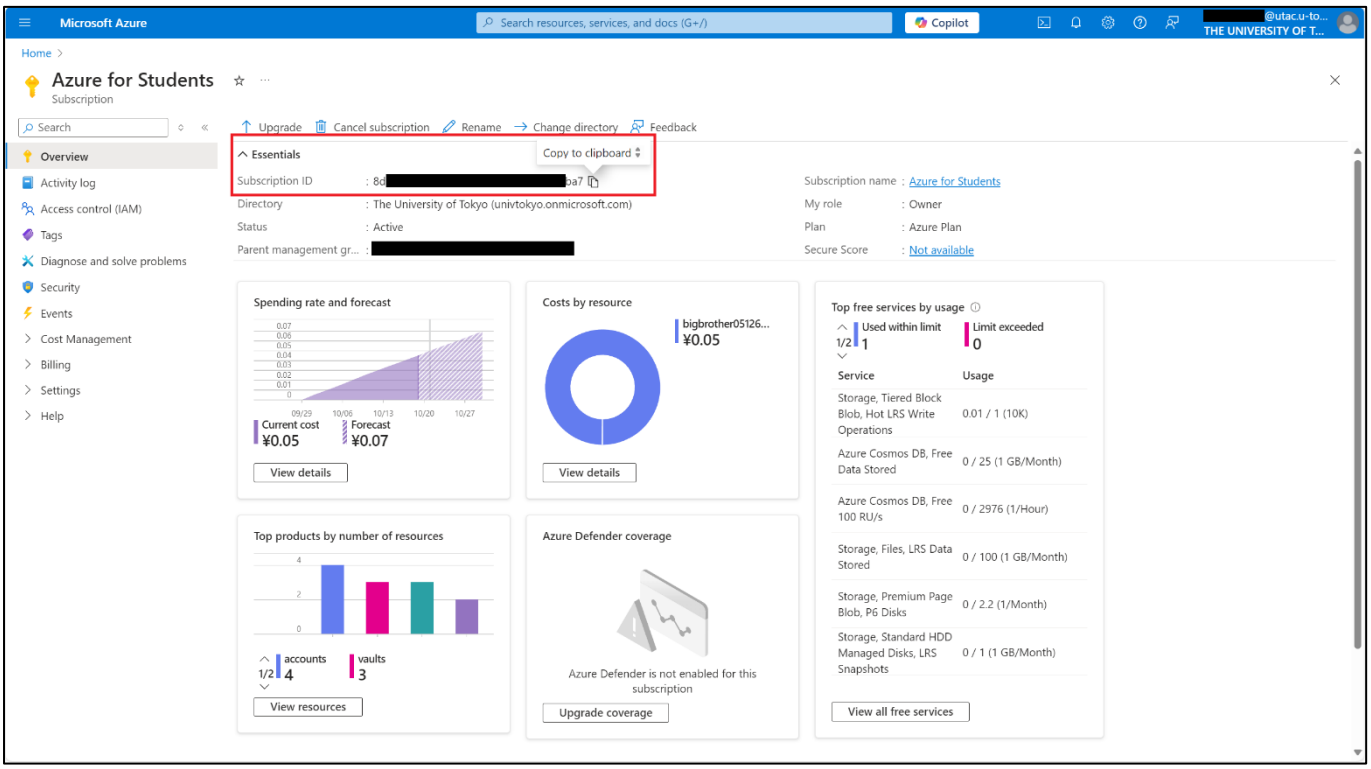
10. Justification *
Describe your scenario to justify the quota increase for the model selected. Please articulate in detail your expected usage needs.
Enter your answer

Follow these steps to find your subscription ID. Do not close quota request form.

On the home page of your project, click on the link under the “Subscription” property in “Project properties”. Here, in the screenshot, the subscription is “Azure for Students”, but your subscription name may be different.



This link will take you to your subscription page in Microsoft Azure Portal, where you can find your subscription ID and other usage data, such as the cost and the predicted cost. If you hover your mouse over this ID, you can find a “Copy to clipboard” button; click on it to copy your subscription ID.



Back in the form, once you have pasted your subscription ID in the field, you will need to fill in the additional information about the requested quota. If you do not understand the meaning of these options, you can choose as follows. In a nutshell, in the “Global standard” configuration, data is stored at the specified location, meanwhile, models can be accessed from any location.

After filling in the quota you need, click “Submit” button.

11. Quota Request Type *

Select the quota request type. You can learn more about different deployment types here https://learn.microsoft.com/en-us/azure/ai-services/openai/how-to/deployment-types

- (●) Global Standard
- ( ) Global Batch
- ( ) Standard
- ( ) Provisioned

12. Global Standard Region *

Select the region where you require increased quota.

As regional availability is limited, your selected region may not be available to grant additional quota for specific models. If quota is NOT AVAILABLE in your preferred region, please let us know if you would prefer your request be declined, or granted quota in an alternate region by selecting that in the next question. (in the past quota was granted in alternative regions automatically, but this has caused customer confusion, so we are changing the process.)

For more information on model & region availability, please see this matrix.

- ( ) Australia East
- ( ) Brazil South
- ( ) East US
- ( ) East US 2
- ( ) France Central
- (●) Japan East
- ( ) Korea Central
- ( ) North Central US
- ( ) Norway East
- ( ) Poland Central
- ( ) Spain Central
- ( ) South Africa North

13. Global Standard Model *

Select the model where you require increased quota. Global Standard is currently accessible through only US Regions.

gpt-4o

14. Global Standard Quota *

Enter total quota required. New value must be integers only. Text quota is allocated in units of 1000.

Example: If you need 400,000 tokens per minute (TPM) for gpt-35-turbo; then, that value would be 400,000 / 1000 units. You would enter 400 as the value below.

400

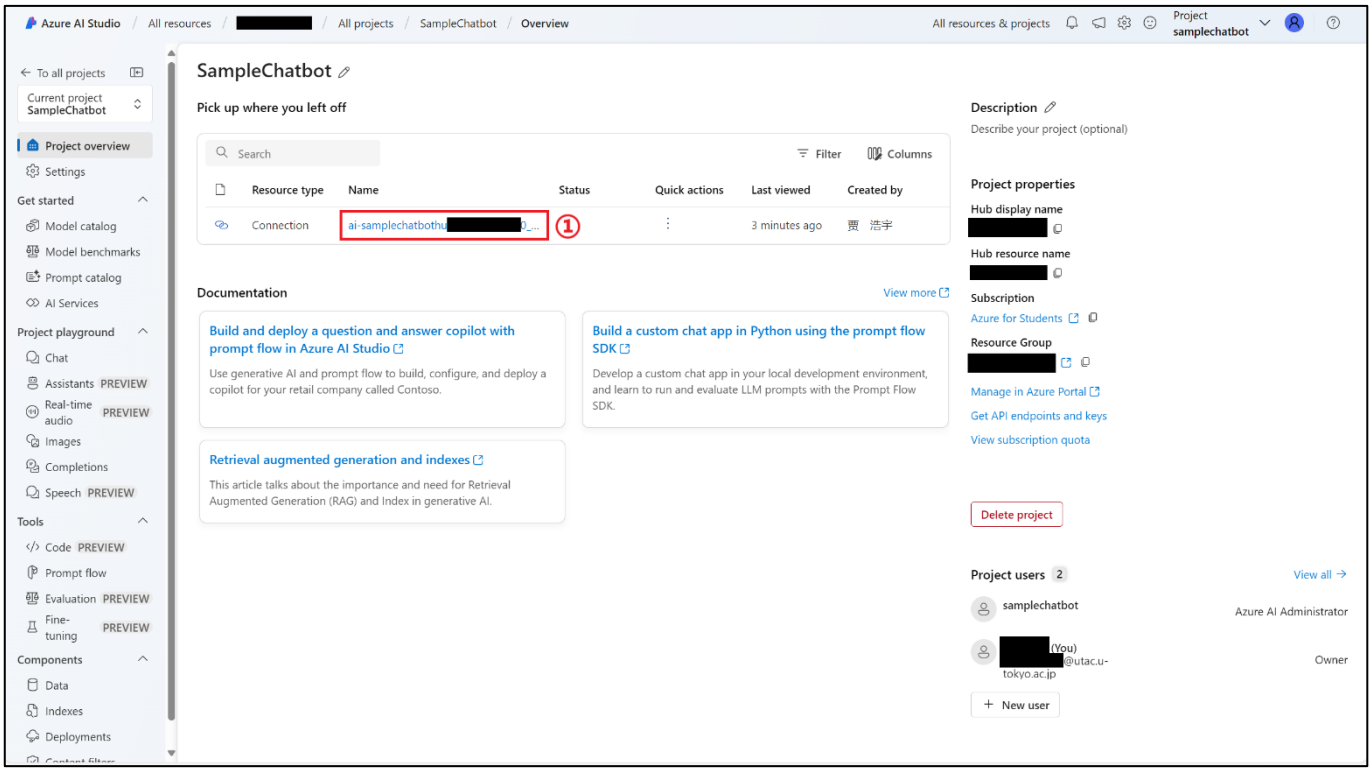
Submit

2. Use Python API

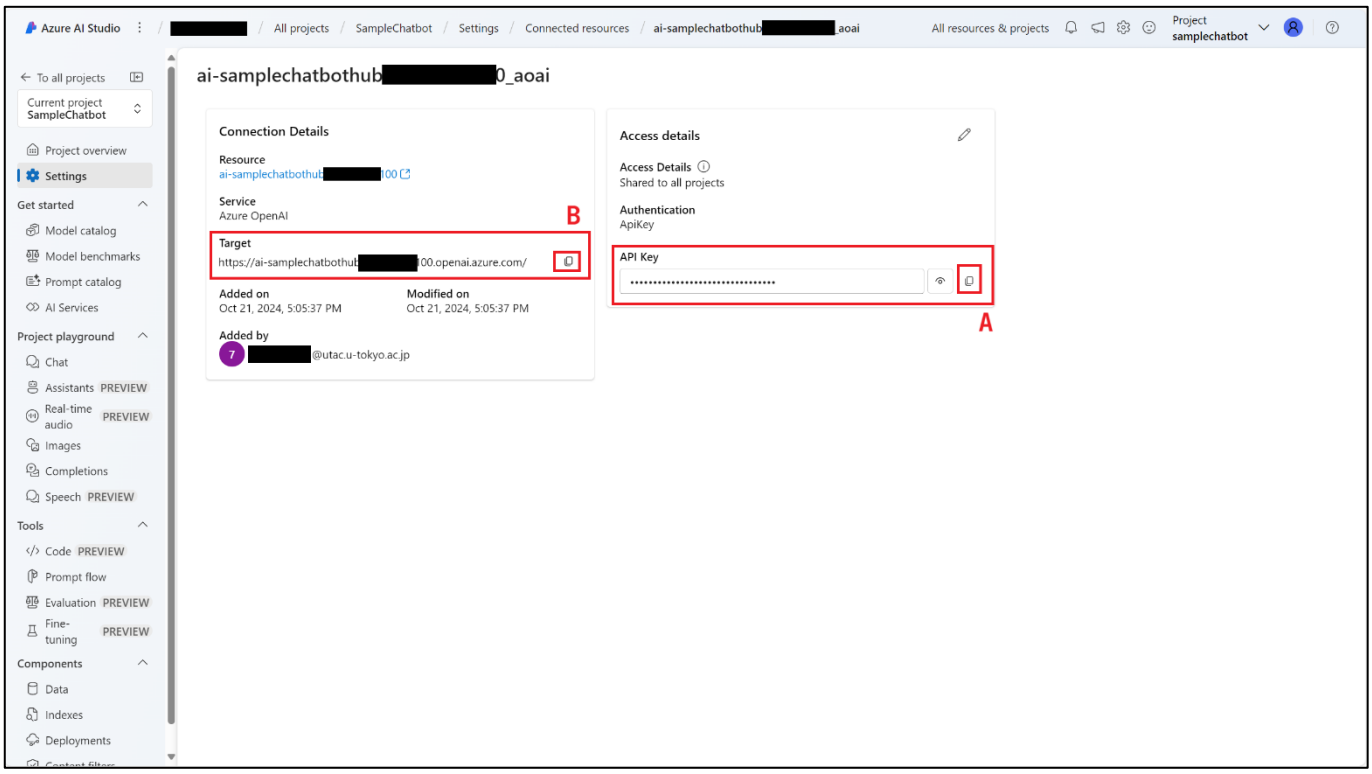
Azure OpenAI Service is compatible with Python client provided by OpenAI. However, some configuration is required. To use services hosted by OpenAI, you only need an API key, but to use models that you manually deployed in Azure AI Studio, you need an endpoint to the “hub” in addition to the API key.

Step 1. Get endpoint, key and model version.

In the home page of your project, click the link next to the “Connection” resource.

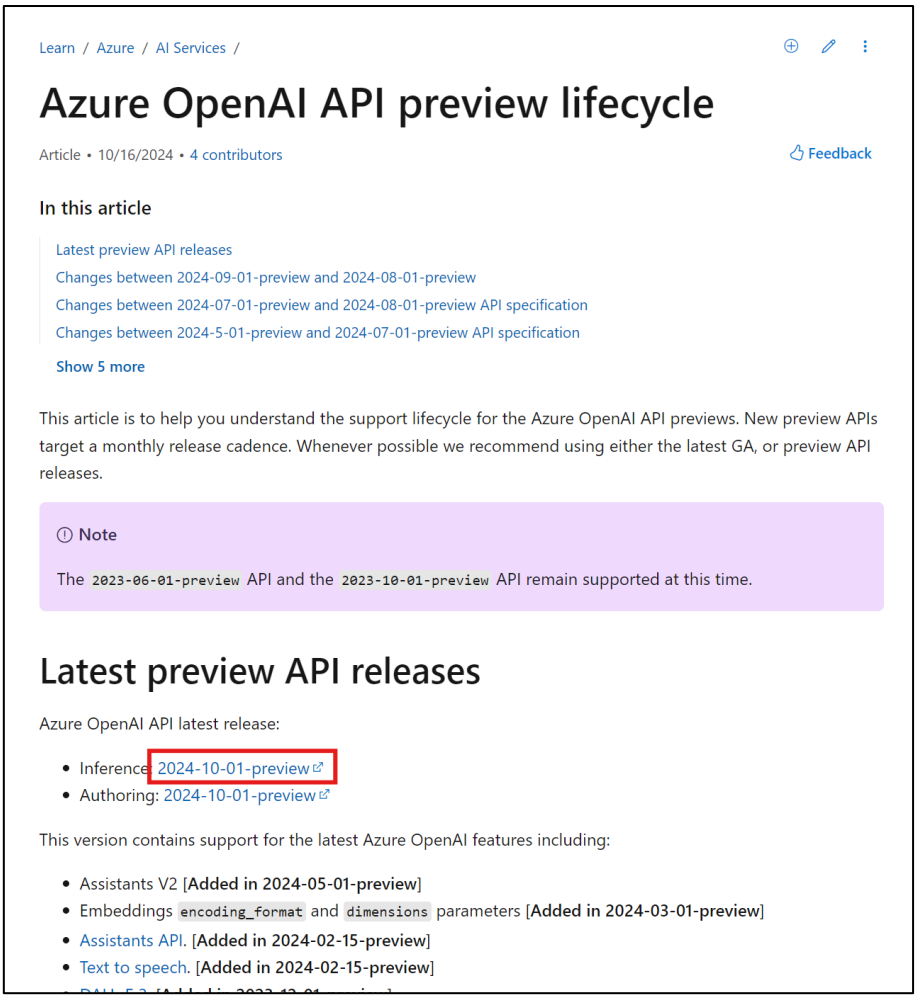


Then, you can find a field named “Target”, which is the endpoint of this hub; also you can a field named “API Key”.



Notice: The resource link on this page will take you to the details page in Azure Portal, where you can regenerate these API keys or find endpoints for other services such as Text-to-Speech.

In this page (https://learn.microsoft.com/en-us/azure/ai-services/openai/api-version-deprecation), you can find the API versions. In this screenshot, the latest API version is "2024-10-01-preview".



Learn / Azure / AI Services /

Azure OpenAI API preview lifecycle

Article • 10/16/2024 • 4 contributors

Feedback

In this article

Latest preview API releases
Changes between 2024-09-01-preview and 2024-08-01-preview
Changes between 2024-07-01-preview and 2024-08-01-preview API specification
Changes between 2024-5-01-preview and 2024-07-01-preview API specification
Show 5 more

This article is to help you understand the support lifecycle for the Azure OpenAI API previews. New preview APIs target a monthly release cadence. Whenever possible we recommend using either the latest GA, or preview API releases.

Note

The `2023-06-01-preview` API and the `2023-10-01-preview` API remain supported at this time.

Latest preview API releases

Azure OpenAI API latest release:

- Inference: 2024-10-01-preview
- Authoring: 2024-10-01-preview

This version contains support for the latest Azure OpenAI features including:

- Assistants V2 [**Added in 2024-05-01-preview**]
- Embeddings `encoding_format` and `dimensions` parameters [**Added in 2024-03-01-preview**]
- Assistants API. [**Added in 2024-02-15-preview**]
- Text to speech. [**Added in 2024-02-15-preview**]

Step 2. Install Python package for OpenAI.

Type "pip install openai" in the terminal to install the package for OpenAI.

Step 3. Configure the Python code for Azure OpenAI.

```python
from openai import OpenAI

client = OpenAI(
    api_key="YOUR_OPENAI_KEY"
)

completion = client.chat.completions.create(
    model="gpt-4",
    messages=[
        {"role": "system", "content": "You are a helpful assistant."},
        {
            "role": "user",
            "content": "Write a haiku about recursion in programming."
        }
    ]
)

print(completion.choices[0].message)
```

```python
from openai import AzureOpenAI

client = AzureOpenAI(
    api_key="[redacted]",
    api_version="2024-10-01-preview",
    azure_endpoint="https://ai-samplechatbothub[redacted].openai.azure.com/"
)

completion = client.chat.completions.create(
    model="gpt-4",
    messages=[
        {"role": "system", "content": "You are a helpful assistant."},
        {
            "role": "user",
            "content": "Write a haiku about recursion in programming."
        }
    ]
)

print(completion.choices[0].message)
```

There are two major differences: (1) class “OpenAI” is replaced by “AzureOpenAI”; (2) “api_version”, and “azure_endpoints” are provided as constructor parameters. Run this script, you can see the reply from GPT (which is likely not to be exactly the same to the following result):

```
ChatCompletionMessage(content='In code loops within,\nA function calls itself—\nDepth in each step found.',
refusal=None, role='assistant', audio=None, function_call=None, tool_calls=None)
```