

# 01\_Prepare for an AI development project

In this exercise, you use Azure AI Foundry portal to create a project, ready to build an AI solution.

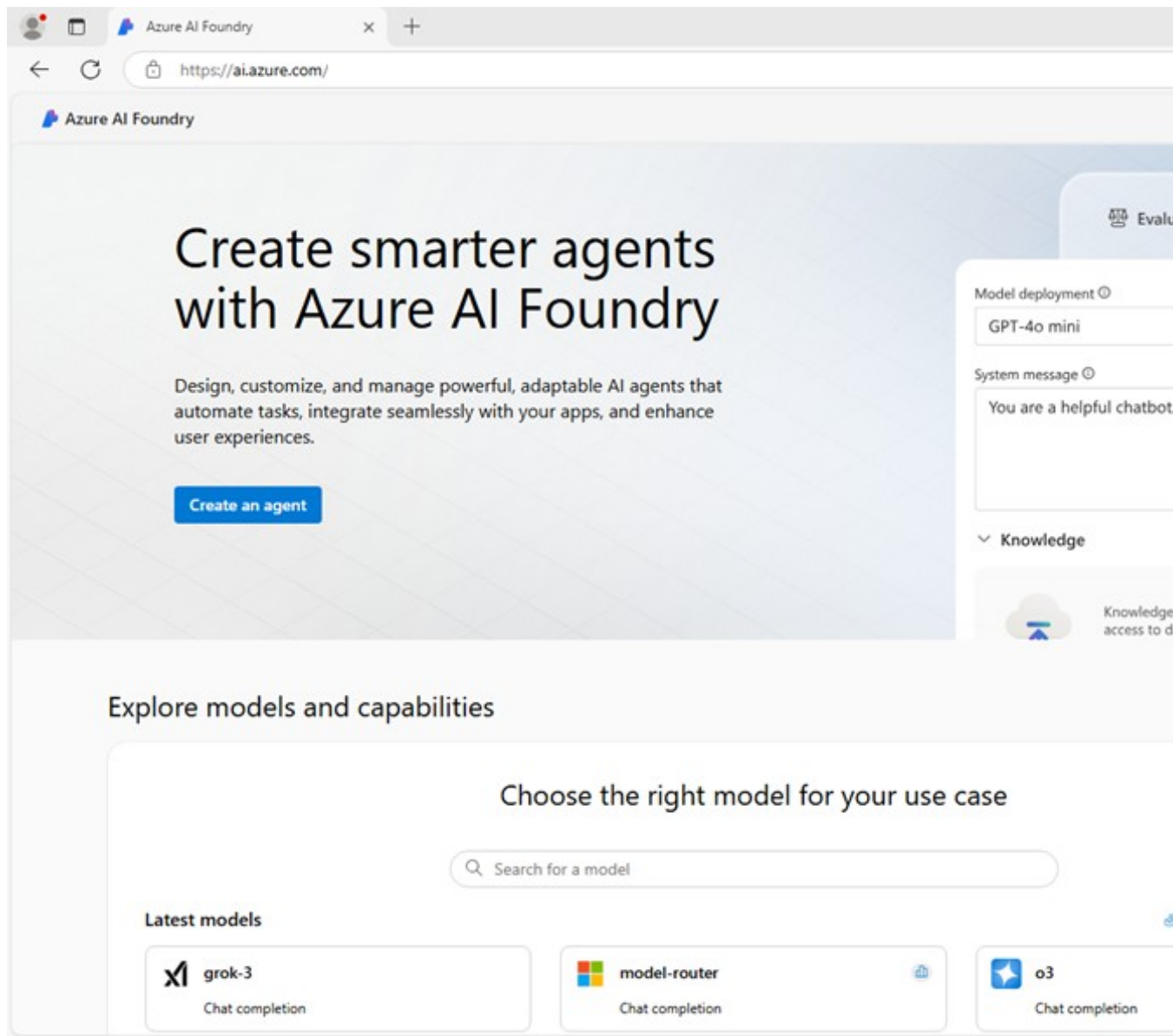
This exercise takes approximately **30** minutes.

**Note:** Some of the technologies used in this exercise are in preview or in active development. You may experience some unexpected behavior, warnings, or errors.

## Open Azure AI Foundry portal

Let's start by signing into Azure AI Foundry portal.

1. In a web browser, open the [Azure AI Foundry portal](#) at `https://ai.azure.com` and sign in using your Azure credentials. Close any tips or quick start panes that are opened the first time you sign in, and if necessary use the **Azure AI Foundry** logo at the top left to navigate to the home page, which looks similar to the following image (close the **Help** pane if it's open):



2. Review the information on the home page.

## Create a project

An Azure AI project provides a collaborative workspace for AI development. Let's start by choosing a model that we want to work with and creating a project to use it in.

**Note:** AI Foundry projects can be based on an Azure AI Foundry resource, which provides access to AI models (including Azure OpenAI), Azure AI services, and other resources for developing AI agents and chat solutions. Alternatively, projects can be based on AI hub resources; which include connections to Azure resources for secure storage, compute, and specialized tools. Azure AI Foundry based projects are great for developers who want to manage resources for AI agent or chat app development. AI hub based projects are more suitable for enterprise development teams working on complex AI solutions.

1. In the home page, in the **Explore models and capabilities** section, search for the **gpt-4o** model; which we'll use in our project.

2. In the search results, select the **gpt-4o** model to see its details, and then at the top of the page for the model, select **Use this model**.

3. When prompted to create a project, enter a valid name for your project and expand **Advanced options**.

4. Select **Customize** and specify the following settings for your project:

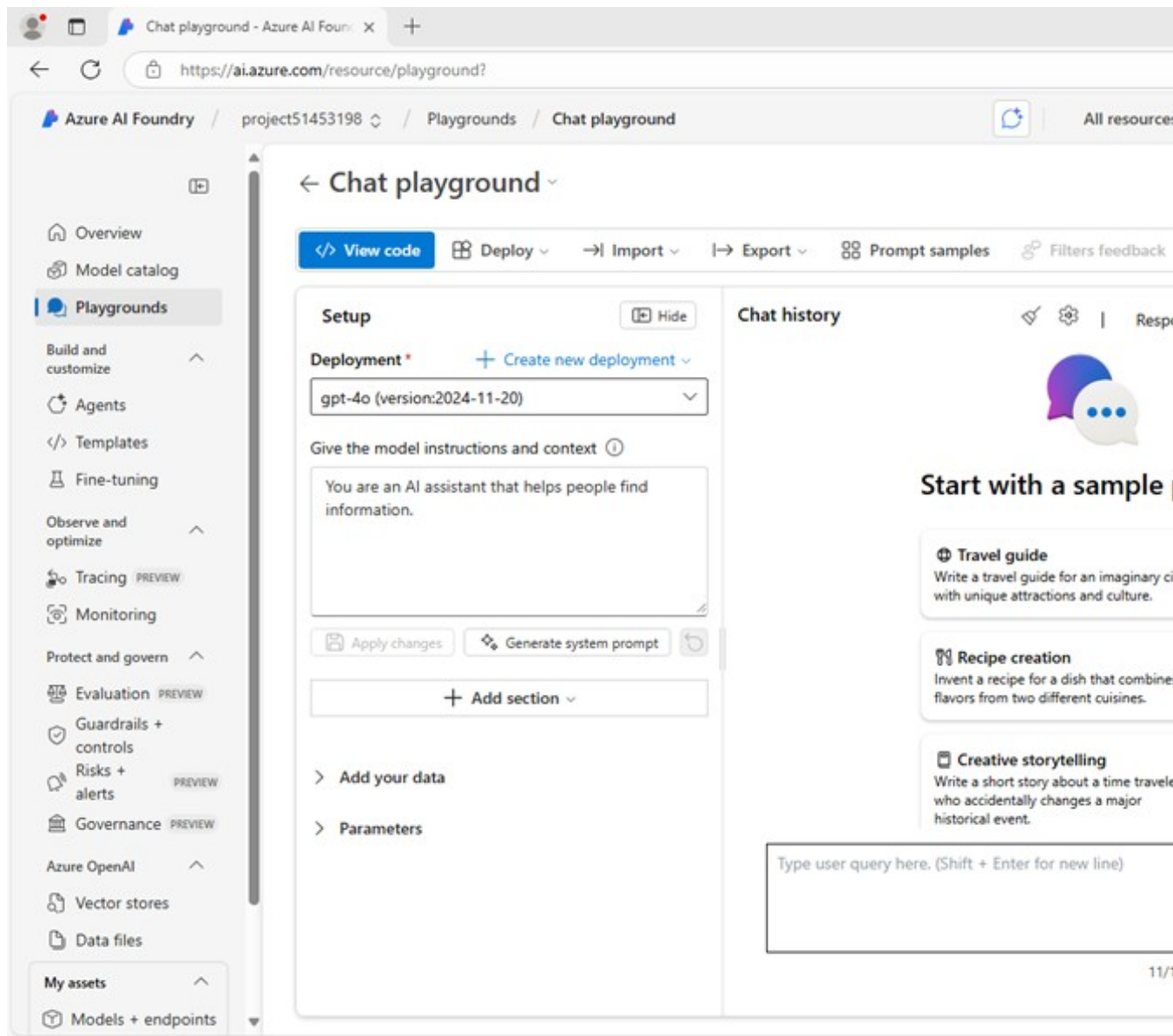
- **Azure AI Foundry resource:** A valid name for your Azure AI Foundry resource
- **Subscription:** Your Azure subscription
- **Resource group:** Create or select a resource group
- **Region:** Select any **AI Services supported location**\*

\* Some Azure AI resources are constrained by regional model quotas. In the event of a quota limit being exceeded later in the exercise, there's a possibility you may need to create another resource in a different region.

5. Select **Create** and wait for your project to be created. If prompted, deploy the gpt-4o model using the **Global standard** deployment type and customize the deployment details to set a **Tokens per minute rate limit** of 50K (or the maximum available if less than 50K).

**Note:** Reducing the TPM helps avoid over-using the quota available in the subscription you are using. 50,000 TPM should be sufficient for the data used in this exercise. If your available quota is lower than this, you will be able to complete the exercise but you may experience errors if the rate limit is exceeded.

6. When your project is created, the chat playground will be opened automatically so you can test your model:



## [Get Started Data from the lab practice](#)

Below are example code snippets for a few use cases. For additional information about Azure OpenAI SDK, see full documentation and samples .

### 1. Authentication using API Key

For OpenAI API Endpoints, deploy the Model to generate the endpoint URL and an API key to authenticate against the service. In this sample endpoint and key are strings holding the endpoint URL and the API Key.

The API endpoint URL and API key can be found on the Deployments + Endpoint page once the model is deployed.

To create a client with the OpenAI SDK using an API key, initialize the client by passing your API key to the SDK's configuration. This allows you to authenticate and interact with OpenAI's services seamlessly:

```
import os
```

```

from openai import AzureOpenAI

client = AzureOpenAI(
    api_version="2024-12-01-preview",
    azure_endpoint="https://ai-dev-project-resource-0603.cognitiveservices.azure.com/",
    api_key=subscription_key,
)

```

## 2. Install dependencies

Install the Azure Open AI SDK using pip (Requires: Python >=3.8):

```
pip install openai
```

## 3. Run a basic code sample

This sample demonstrates a basic call to the chat completion API. The call is synchronous.

```

import os
from openai import AzureOpenAI

endpoint = "https://ai-dev-project-resource-0603.cognitiveservices.azure.com/"
model_name = "gpt-4o"
deployment = "gpt-4o"

subscription_key = "<your-api-key>"
api_version = "2024-12-01-preview"

client = AzureOpenAI(
    api_version=api_version,
    azure_endpoint=endpoint,
    api_key=subscription_key,
)

response = client.chat.completions.create(
    messages=[
        {
            "role": "system",
            "content": "You are a helpful assistant.",
        },
        {
            "role": "user",
            "content": "I am going to Paris, what should I see?",
        }
    ],
    max_tokens=4096,
    temperature=1.0,
    top_p=1.0,
    model=deployment
)

print(response.choices[0].message.content)

```

## 4. Explore more samples

### Run a multi-turn conversation

This sample demonstrates a multi-turn conversation with the chat completion API. When using the model for a chat application, you'll need to manage the history of that conversation and send the latest messages to the model.

```

import os
from openai import AzureOpenAI

```

```

endpoint = "https://ai-dev-project-resource-0603.cognitiveservices.azure.com/"
model_name = "gpt-4o"
deployment = "gpt-4o"

subscription_key = "<your-api-key>"
api_version = "2024-12-01-preview"

client = AzureOpenAI(
    api_version=api_version,
    azure_endpoint=endpoint,
    api_key=subscription_key,
)

response = client.chat.completions.create(
    messages=[
        {
            "role": "system",
            "content": "You are a helpful assistant.",
        },
        {
            "role": "user",
            "content": "I am going to Paris, what should I see?",
        },
        {
            "role": "assistant",
            "content": "Paris, the capital of France, is known for its stunning
architecture, art museums, historical landmarks, and romantic atmosphere. Here are
some of the top attractions to see in Paris:\n\n 1. The Eiffel Tower: The iconic
Eiffel Tower is one of the most recognizable landmarks in the world and offers
breathtaking views of the city.\n 2. The Louvre Museum: The Louvre is one of the
world's largest and most famous museums, housing an impressive collection of art
and artifacts, including the Mona Lisa.\n 3. Notre-Dame Cathedral: This beautiful
cathedral is one of the most famous landmarks in Paris and is known for its Gothic
architecture and stunning stained glass windows.\n\n These are just a few of the
many attractions that Paris has to offer. With so much to see and do, it's no
wonder that Paris is one of the most popular tourist destinations in the world.",
        },
        {
            "role": "user",
            "content": "What is so great about #1?",
        },
    ],
    max_tokens=4096,
    temperature=1.0,
    top_p=1.0,
    model=deployment
)

print(response.choices[0].message.content)

```

## Stream the output

For a better user experience, you will want to stream the response of the model so that the first token shows up early and you avoid waiting for long responses.

```

import os
from openai import AzureOpenAI

endpoint = "https://ai-dev-project-resource-0603.cognitiveservices.azure.com/"
model_name = "gpt-4o"
deployment = "gpt-4o"

```

```
subscription_key = "<your-api-key>"
api_version = "2024-12-01-preview"

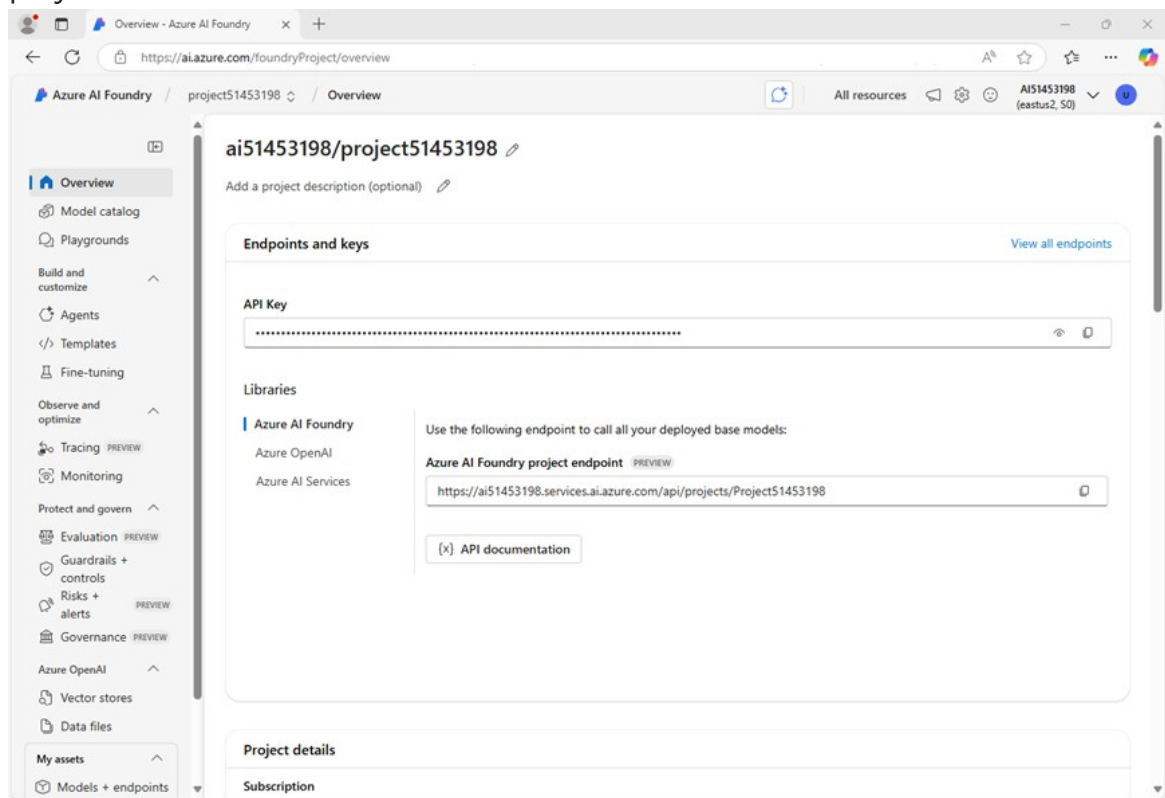
client = AzureOpenAI(
    api_version=api_version,
    azure_endpoint=endpoint,
    api_key=subscription_key,
)

response = client.chat.completions.create(
    stream=True,
    messages=[
        {
            "role": "system",
            "content": "You are a helpful assistant.",
        },
        {
            "role": "user",
            "content": "I am going to Paris, what should I see?",
        }
    ],
    max_tokens=4096,
    temperature=1.0,
    top_p=1.0,
    model=deployment,
)

for update in response:
    if update.choices:
        print(update.choices[0].delta.content or "", end="")

client.close()
```

7. In the navigation pane on the left, select **Overview** to see the main page for your project; which looks like this:



8. At the bottom of the navigation pane on the left, select **Management center**. The management center is where you can configure settings at both the resource and project levels; which are both shown in the navigation pane.



The screenshot shows the 'Overview' page for an Azure AI Foundry resource. The left navigation pane includes 'Management center' (All resources, Quota), 'Resource (AI51453198)' (Overview, Users, Connected resources), and 'Project (Project51453198)' (Overview, Connected resources, Go to project). The main content area is titled 'AI Resource Overview' and contains a 'Resource configuration' section with fields for Name (AI51453198), Subscription (MOCOAI-Iod49253185), API key 1 and 2 (masked), Resource group (ResourceGroup1), and Azure OpenAI endpoint (https://ai51453198.cognitiveservices.azure.com/). Below this is a table with one row: project51453198, created on 2025-05-19T21:45:04....

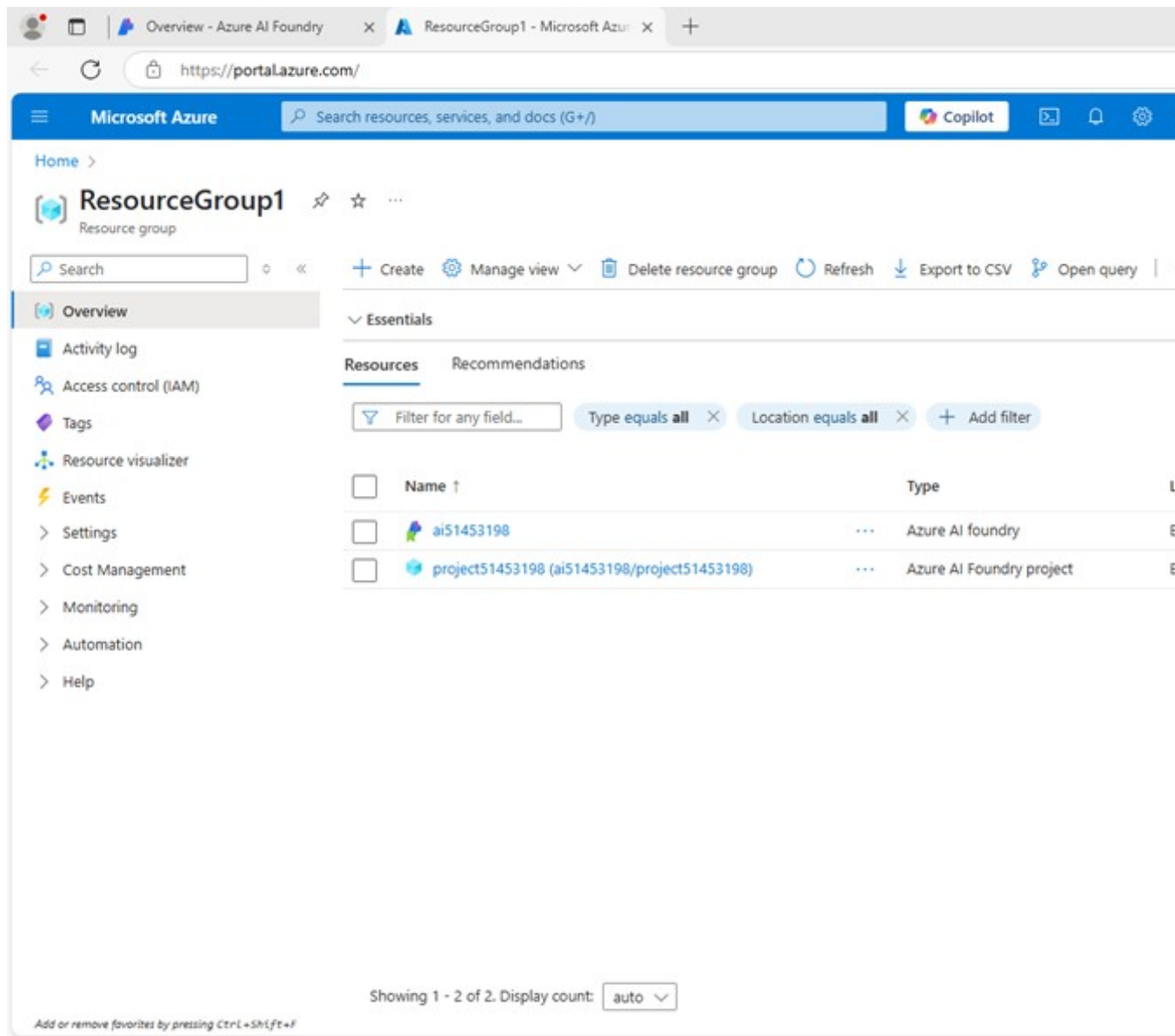
The resource level relates to the **Azure AI Foundry** resource that was created to support your project. This resource includes connections to Azure AI Services and Azure AI Foundry models; and provides a central place to manage user access to AI development projects.

The project level relates to your individual project, where you can add and manage project-specific resources.

9. In the navigation pane, in the section for your Azure AI Foundry resource, select the **Overview** page to view its details.

10. Select the link to the **Resource group** associated with the resource to open a new browser tab and navigate to the Azure portal. Sign in with your Azure credentials if prompted.

11. View the resource group in the Azure portal to see the Azure resources that have been created to support your Azure AI Foundry resource and your project.



Note that the resources have been created in the region you selected when creating the project.

12. Close the Azure portal tab and return to the Azure AI Foundry portal.

## Review project endpoints

The Azure AI Foundry project includes a number of endpoints that client applications can use to connect to the project and the models and AI services it includes.

1. In the Management center page, in the navigation pane, under your project, select **Go to resource**.
2. In the project **Overview** page, view the **Endpoints and keys** section; which contains endpoints and authorization keys that you can use in your application code to access:
  - The Azure AI Foundry project and any models deployed in it.
  - Azure OpenAI in Azure AI Foundry models.

- Azure AI services

## Test a generative AI model

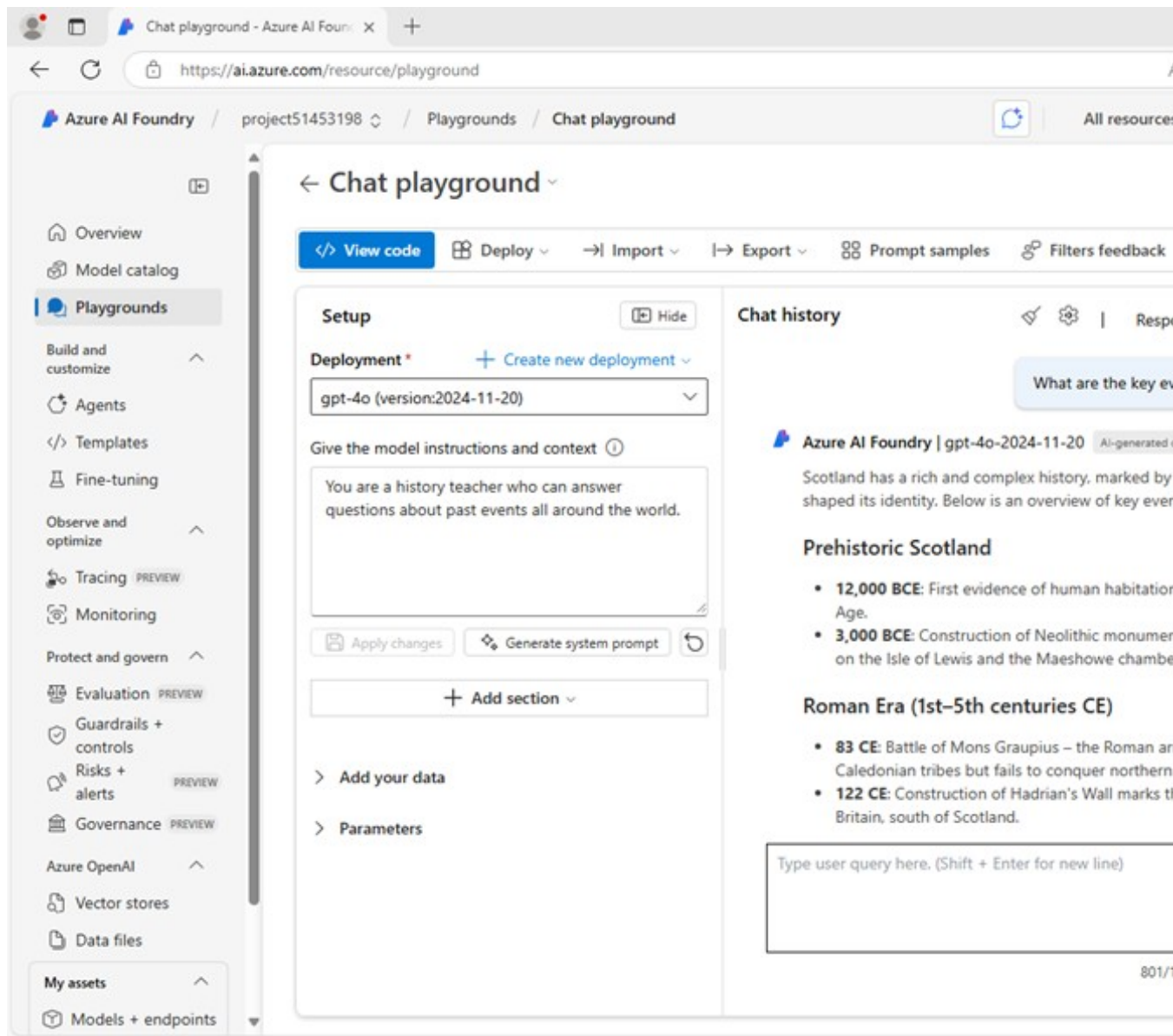
Now that you know something about the configuration of your Azure AI Foundry project, you can return to the chat playground to explore the model you deployed.

- 1.In the navigation pane on the left for your project, select **Playgrounds**
- 2.Open the **Chat playground**, and ensure that your **gpt-4o** model deployment is selected in the **Deployment** section.
- 3.In the **Setup** pane, in the **Give the model instructions and context** box, enter the following instructions:

Code

**You** are a history teacher who can answer questions about past events **all** around the world.

- 4.Apply the changes to update the system message.
- 5.In the chat window, enter a query such as **What are the key events in the history of Scotland?** and view the response:



## Summary

In this exercise, you've explored Azure AI Foundry, and seen how to create and manage projects and their related resources.

## Clean up

If you've finished exploring Azure AI Foundry portal, you should delete the resources you have created in this exercise to avoid incurring unnecessary Azure costs.

1. In the [Azure portal](https://portal.azure.com) at <https://portal.azure.com>, view the contents of the resource group where you deployed the resources used in this exercise.
2. On the toolbar, select **Delete resource group**.
3. Enter the resource group name and confirm that you want to delete it.