Develop an AI app with the Microsoft Foundry SDK

1. Introduction

https://learn.microsoft.com/en-us/training/modules/ai-foundry-sdk/01-introduction

Introduction

Completed

- 1 minute

Developers creating AI solutions with Microsoft Foundry need to work with a combination of services and software frameworks. The Microsoft Foundry SDK is designed to bring together common services and code libraries in an AI project through a central programmatic access point, making it easier for developers to write the code needed to build effective AI apps on Azure.

In this module, you'll learn how to use the Microsoft Foundry SDK to work with resources in an AI project.

Note

Microsoft Foundry SDK is currently in public preview. Details described in this module are subject to change.

2. What is the Microsoft Foundry SDK?

https://learn.microsoft.com/en-us/training/modules/ai-foundry-sdk/02-azure-ai-foundry-sdk

What is the Microsoft Foundry SDK?

Completed

- 5 minutes

Microsoft Foundry provides a REST API that you can use to work with AI Foundry projects and the resources they contain. Additionally, multiple language-specific SDKs are available, enabling developers to write code that uses resources in A Microsoft Foundry project in their preferred development language. With A Microsoft Foundry SDK, developers can create applications that connect to a project, access the resource connections and models in that project, and use them to perform AI operations, such as sending prompts to a generative AI model and processing the responses.

The core package for working with projects is the **Azure AI Projects** library, which enables you to connect to A Microsoft Foundry project and access the resources defined within it. Available language-specific packages the for Azure AI Projects library include:

- Azure AI Projects for Python
- Azure AI Projects for Microsoft .NET
- Azure AI Projects for JavaScript

Note

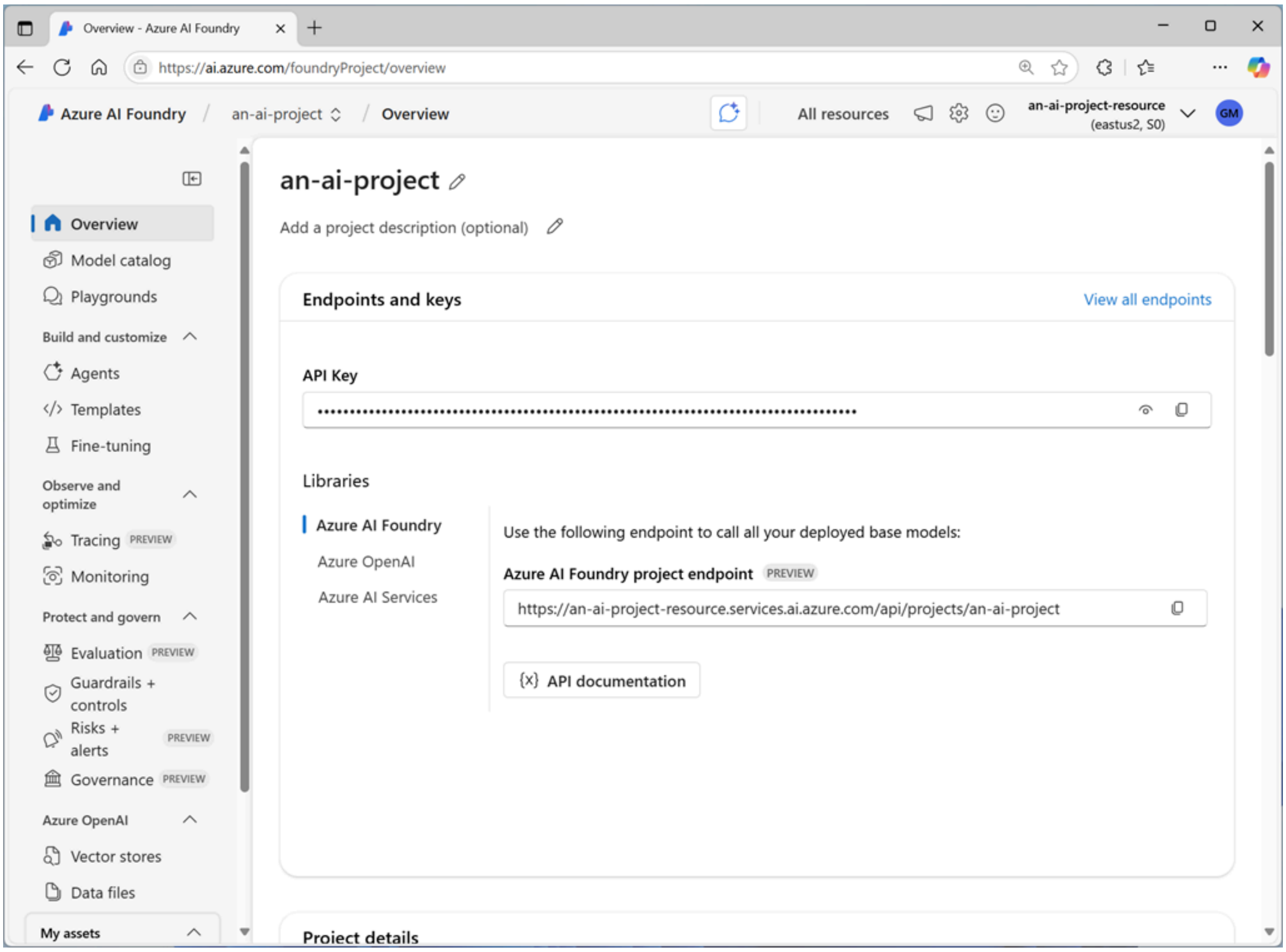
In this module, we'll use Python code examples for common tasks that a developer may need to perform with Microsoft Foundry projects. You can refer to the other language-specific SDK documentation to find equivalent code for your preferred language. Each SDK is developed and maintained independently, so some functionality may be at different stages of implementation for each language.

To use the Azure AI Projects library in Python, you can use the **pip** package installation utility to install the **azure-ai-projects** package from PyPi:

```
pip install azure-ai-projects
```

Using the SDK to connect to a project

The first task in most Microsoft Foundry SDK code is to connect to A Microsoft Foundry project. Each project has a unique *endpoint*, which you can find on the project's **Overview** page in the Microsoft Foundry portal.



Use the following endpoint to call all your deployed base models:

Azure AI Foundry project endpoint PREVIEW

https://an-ai-project-resource.services.ai.azure.com/api/projects/an-ai-project

Note

The project provides multiple endpoints and keys, including:

- An endpoint for the project itself; which can be used to access project connections, agents, and models in the Microsoft Foundry resource.
- An endpoint for Azure OpenAI Service APIs in the project's Microsoft Foundry resource.
- An endpoint for Foundry Tools APIs (such as Azure Vision and Azure Language) in the Microsoft Foundry resource.

You can use the project endpoint in your code to create an **AIProjectClient** object, which provides a programmatic proxy for the project, as shown in this Python example:

```
 from azure.identity import DefaultAzureCredential
from azure.ai.projects import AIProjectClient
...

project_endpoint = "https://......"
project_client = AIProjectClient(
    credential=DefaultAzureCredential(),
    endpoint=project_endpoint)
```

Note

The code uses the default Azure credentials to authenticate when accessing the project. To enable this authentication, in addition to the **azure-ai-projects** package, you need to install the **azure-identity** package:

`pip install azure-identity`

Tip

To access the project successfully, the code must be run in the context of an authenticated Azure session. For example, you could use the Azure command-line interface (CLI) `az-login` command to sign in before running the code.

3. Work with project connections

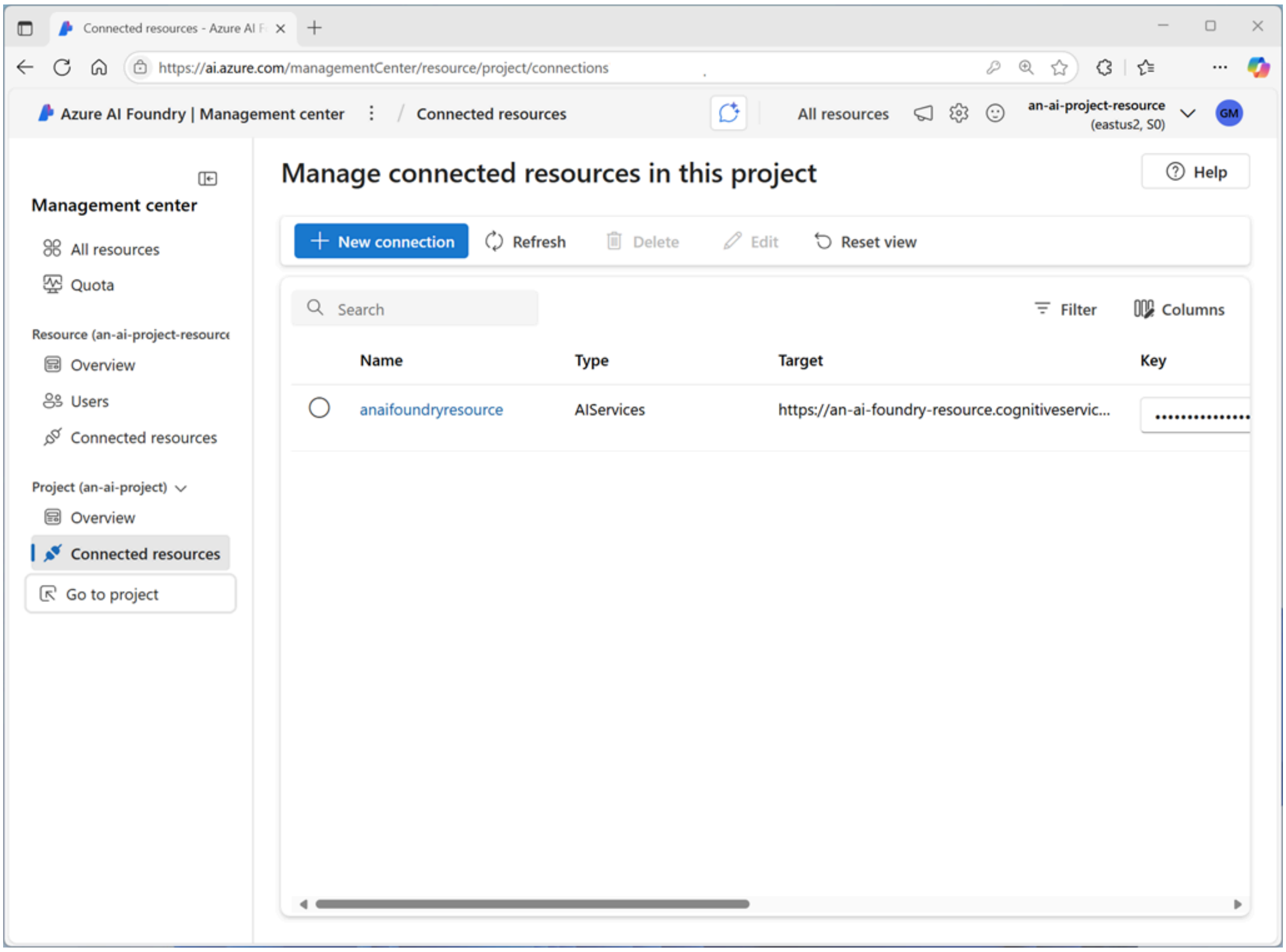
https://learn.microsoft.com/en-us/training/modules/ai-foundry-sdk/03-connections

Work with project connections

Completed

- 5 minutes

Each Microsoft Foundry project includes **connected resources**, which are defined both at the *parent* (Microsoft Foundry resource or hub) level, and at the *project* level. Each resource is a *connection* to an external service, such as Azure storage, Azure AI Search, Azure OpenAI, or another Microsoft Foundry resource.



Manage connected resources in this project

Name	Type	Target	Key
anaifoundryresource	AIServices	https://an-ai-foundry-resource.cognitiveservic...	....

With the Microsoft Foundry SDK, you can connect to a project and retrieve connections; which you can then use to consume the connected services.

For example, the **AIProjectClient** object in Python has a **connections** property, which you can use to access the resource connections in the project. Methods of the **connections** object include:

- `connections.list()` : Returns a collection of connection objects, each representing a connection in the project. You can filter the results by specifying an optional **connection_type** parameter with a valid enumeration, such as `ConnectionType.AZURE_OPEN_AI` .
- `connections.get(connection_name, include_credentials)` : Returns a connection object for the connection with the name specified. If the **include_credentials** parameter is **True** (the default value), the credentials required to connect to the connection are returned - for example, in the form of an API key for a Foundry Tools resource.

The connection objects returned by these methods include connection-specific properties, including credentials, which you can use to connect to the associated resource.

The following code example lists all of the resource connections that have been added to a project:

```
 from azure.identity import DefaultAzureCredential
from azure.ai.projects import AIProjectClient
```

```
try:

    # Get project client
    project_endpoint = "https://....."
    project_client = AIProjectClient(
            credential=DefaultAzureCredential(),
            endpoint=project_endpoint,
        )

    ## List all connections in the project
    connections = project_client.connections
    print("List all connections:")
    for connection in connections.list():
        print(f"{connection.name} ({connection.type})")

except Exception as ex:
    print(ex)
```

4. Create a chat client

https://learn.microsoft.com/en-us/training/modules/ai-foundry-sdk/04-chat-client

Create a chat client

Completed

- 10 minutes

A common scenario in an AI application is to connect to a generative AI model and use *prompts* to engage in a chat-based dialog with it.

While you can use the Azure OpenAI SDK, to connect "directly" to a model using key-based or Microsoft Entra ID authentication; when your model is deployed in a Microsoft Foundry project, you can also use the Microsoft Foundry SDK to retrieve a project client, from which you can then get an authenticated OpenAI chat client for any models deployed in the project's Microsoft Foundry resource. This approach makes it easy to write code that consumes models deployed in your project, switching between them easily by changing the model deployment name parameter.

Tip

You can use the OpenAI chat client provided by a Microsoft Foundry project to chat with any model deployed in the associated Microsoft Foundry resource - even non-OpenAI models, such as Microsoft Phi models.

The following Python code sample uses the **get_openai_client()** method to get an OpenAI client with which to chat with a model that has been deployed in the project's Microsoft Foundry resource.

```python
 from azure.identity import DefaultAzureCredential
from azure.ai.projects import AIProjectClient
from openai import AzureOpenAI

try:

    # connect to the project
    project_endpoint = "https://......"
    project_client = AIProjectClient(
            credential=DefaultAzureCredential(),
            endpoint=project_endpoint,
        )

    # Get a chat client
    chat_client = project_client.get_openai_client(api_version="2024-10-21")

    # Get a chat completion based on a user-provided prompt
    user_prompt = input("Enter a question:")

    response = chat_client.chat.completions.create(
        model=your_model_deployment_name,
        messages=[
            {"role": "system", "content": "You are a helpful AI assistant."},
            {"role": "user", "content": user_prompt}
        ]
    )
    print(response.choices[0].message.content)

except Exception as ex:
    print(ex)
```

Note

In addition to the **azure-ai-projects** and **azure-identity** packages discussed previously, the sample code shown here assumes that the **openai** package has been installed:

```
pip install openai
```

5. Exercise - Create a generative AI chat app

https://learn.microsoft.com/en-us/training/modules/ai-foundry-sdk/05-exercise

Exercise - Create a generative AI chat app

Completed

- 30 minutes

Now it's your turn to try using the Microsoft Foundry SDK!

In this exercise, you use the Microsoft Foundry SDK to connect to a project and create an application that chats with a generative AI model.

> **Note**
>
> To complete this lab, you need an **Azure subscription** in which you have administrative access.

Launch the exercise and follow the instructions.

Launch Exercise

6. Module assessment

https://learn.microsoft.com/en-us/training/modules/ai-foundry-sdk/06-knowledge-check

Module assessment

Completed

- 3 minutes

7. Summary

https://learn.microsoft.com/en-us/training/modules/ai-foundry-sdk/07-summary

Summary

Completed

- 1 minute

By using the Microsoft Foundry SDK, you can develop rich AI applications that use resources in your Microsoft Foundry projects. The Microsoft Foundry SDK **AIProjectClient** class provides a programmatic proxy for a project, enabling you to access connected resources and to use service-specific libraries to consume them.

Learn more

- The Microsoft Foundry SDK
- Microsoft Foundry Discord
- Microsoft Foundry Developer Forum