Develop an AI agent with Microsoft Foundry Agent Service

1. Introduction

https://learn.microsoft.com/en-us/training/modules/develop-ai-agent-azure/1-introduction

Introduction

Completed

- 2 minutes

Microsoft Foundry Agent Service is a fully managed service designed to empower developers to securely build, deploy, and scale high-quality, extensible AI agents without needing to manage the underlying compute and storage resources.

Imagine you're working in the healthcare industry, where there's a need to automate patient interactions and streamline administrative tasks. Your organization wants to develop an AI agent that can handle patient inquiries, schedule appointments, and provide medical information based on real-time data. However, managing the infrastructure and ensuring data security are significant challenges. Microsoft Foundry Agent Service offers a solution by allowing you to create AI agents tailored to your needs through custom instructions and advanced tools. This service simplifies the development process, reduces the amount of code required, and manages the underlying infrastructure, enabling you to focus on building high-quality AI solutions.

In this module, you'll learn how to use the Foundry Agent Service to develop agents.

2. What is an AI agent

https://learn.microsoft.com/en-us/training/modules/develop-ai-agent-azure/2-what-is-ai-agent

What is an AI agent

Completed

- 6 minutes

An AI agent is a software service that uses generative AI to understand and perform tasks on behalf of a user or another program. These agents use advanced AI models to understand context, make decisions, utilize grounding data, and take actions to achieve specific goals. Unlike traditional applications, AI agents can operate independently, executing complex workflows and automating processes without the need of constant human intervention. The evolution of generative AI enables agents to behave intelligently on our behalf, transforming how we can use and integrate these agents.

Understanding what an AI agent is and how to utilize them is crucial for effectively using AI to automate tasks, make informed decisions, and enhance user experiences. This knowledge enables organizations to deploy AI agents strategically, maximizing their potential to drive innovation, improve efficiency, and achieve business objectives.

Why Are AI agents useful?

AI agents are incredibly useful for several reasons:

- **Automation of Routine Tasks**: AI agents can handle repetitive and mundane tasks, freeing up human workers to focus on more strategic and creative activities. This leads to increased productivity and efficiency.
- **Enhanced Decision-Making**: By processing vast amounts of data and providing insights, AI agents support better decision-making. They can analyze trends, predict outcomes, and offer recommendations based on real-time data. AI Agents can even use advanced decision-making algorithms and machine learning models to analyze data and make informed decisions autonomously. This allows them to handle complex scenarios and provide actionable insights, whereas generative AI chat models primarily focus on generating text-based responses.
- **Scalability**: AI agents can scale operations without the need for proportional increases in human resources. This is beneficial for businesses looking to grow without significantly increasing operational costs.
- **24/7 Availability**: Like all software, AI agents can operate continuously without breaks, ensuring that tasks are completed promptly and customer service is available around the clock.

Agents are built to simulate human-like intelligence and can be applied in various domains such as customer service, data analysis, automation, and more.

Examples of AI agent use cases

AI agents have a wide range of applications across various industries. Here are some notable examples:

Personal productivity agents

Personal productivity agents assist individuals with daily tasks such as scheduling meetings, sending emails, and managing to-do lists. For instance, Microsoft 365 Copilot can help users draft documents, create presentations, and analyze data within the Microsoft Office suite.

Research agents

Research agents continuously monitor market trends, gather data, and generate reports. These agents can be used in financial services to track stock performance, in healthcare to stay updated with the latest medical research, or in marketing to analyze consumer behavior.

Sales agents

Sales agents automate lead generation and qualification processes. They can research potential leads, send personalized follow-up messages, and even schedule sales calls. This automation helps sales teams focus on closing deals rather than administrative tasks.

Customer service agents

Customer service agents handle routine inquiries, provide information, and resolve common issues. They can be integrated into chatbots on websites or messaging platforms, offering instant support to customers. For example, Cineplex uses an AI agent to process refund requests, significantly reducing handling time and improving customer satisfaction.

Developer agents

Developer agents help in software development tasks such as code review, bug fixing, and repository management. They can automatically update codebases, suggest improvements, and ensure that coding standards are maintained. GitHub Copilot is a great example of a developer agent.

Understand security risks of AI Agents

As AI agents become more autonomous and integrated into enterprise systems, they introduce new security considerations that go beyond traditional application threats. Because agents can access sensitive data, make decisions, and act independently, developers and organizations must design with security in mind from the start.

The table below summarizes key security risks to consider when developing or deploying AI agents:

Risk Area	**Description**	**Example / Impact**
Data Leakage and Privacy Exposure	AI agents often access sensitive business or user data to perform tasks. Without proper controls, they can unintentionally expose or share confidential information.	An agent summarizing internal files accidentally includes private data in a customer-facing chat.
Prompt Injection and Manipulation Attacks	Malicious users can craft inputs that override an agent's intended behavior, tricking it into revealing data or performing unauthorized actions.	A user embeds hidden instructions in a message, causing the agent to leak system credentials.
Unauthorized Access and Privilege Escalation	Weak authentication or access controls can let agents—or bad actors controlling them—access data or systems they shouldn't.	An AI agent connected to a CRM tool performs admin-level actions, like exporting or deleting records.
Data Poisoning	Attackers may corrupt training or contextual data, causing the agent to make biased, incorrect, or unsafe decisions.	A poisoned dataset causes a customer support agent to recommend fraudulent or harmful content.
Supply Chain Vulnerabilities	Agents often rely on external APIs, plugins, or model endpoints, which expand the attack surface.	A compromised third-party plugin injects malicious code into the agent's workflow.
Over-Reliance on Autonomous Actions	Highly autonomous agents may execute unintended actions if not carefully constrained or validated.	An agent mistakenly sends payments or publishes unverified content.
Inadequate Auditability and Logging	Without detailed logging, it's difficult to trace actions or detect malicious behavior early.	Security teams cannot identify data misuse due to missing or incomplete activity logs.
Model Inversion and Output Leakage	Attackers might exploit model outputs to infer sensitive data used during training or prompting.	Repeated queries extract private information that was part of a fine-tuning dataset.

Mitigation Strategies

To reduce these risks, developers should adopt a **security-by-design** approach that includes:

- Enforcing **role-based access controls (RBAC)** and **least privilege** permissions.

- Adding **prompt filtering and validation** layers to prevent injection attacks.
- Sandboxing or gating sensitive operations behind **human-in-the-loop approvals**.
- Maintaining **comprehensive logging and traceability** for all agent actions.
- Auditing **third-party dependencies** and integrations regularly.
- Continuously retraining and validating models to detect **data drift** or **poisoning attempts**.

By embedding these practices early in development, organizations can deploy AI agents safely and confidently in real-world environments.

Tip

To learn more about GitHub Copilot, explore the GitHub Copilot fundamentals learning path.

Note

You can explore more about agents in general with the Fundamentals of AI agents module.

3. How to use Microsoft Foundry Agent Service

https://learn.microsoft.com/en-us/training/modules/develop-ai-agent-azure/3-how-use-agent-service

How to use Microsoft Foundry Agent Service

Completed

- 7 minutes

Microsoft Foundry Agent Service is a fully managed service designed to empower developers to securely build, deploy, and scale high-quality, extensible AI agents without needing to manage the underlying compute and storage resources. This unit covers the purpose, benefits, key features, and integration capabilities of Microsoft Foundry Agent Service.

Purpose of Microsoft Foundry Agent Service

The Foundry Agent Service allows developers to create AI agents tailored to their needs through custom instructions and advanced tools like code interpreters and custom functions. These agents can answer questions, perform actions, or automate workflows by combining generative AI models with tools that interact with real-world data sources. The service simplifies the development process by reducing the amount of code required and managing the underlying infrastructure.

Previously, developers could create an agent-like experience by using standard APIs in Microsoft Foundry and connect to custom functions or other tools, but doing so would take a significant coding effort. Foundry Agent Service handles all of that for you through AI Foundry to build agents via the portal or in your own app in fewer than 50 lines of code. The exercise in the module explores both methods of building an agent.

Foundry Agent Service is ideal for scenarios requiring advanced language models for workflow automation. It can be used to:

- Answer questions using real-time or proprietary data sources.
- Make decisions and perform actions based on user inputs.
- Automate complex workflows by combining generative AI models with tools that interact with real-world data.

For example, an AI agent can be created to generate reports, analyze data, or even interact with users through apps or chatbots, making it suitable for customer support, data analysis, and automated reporting.

Key features of Foundry Agent Service

Foundry Agent Service offers several key features:

- **Automatic tool calling**: The service handles the entire tool-calling lifecycle, including running the model, invoking tools, and returning results.
- **Securely managed data**: Conversation states are securely managed using threads, eliminating the need for developers to handle this manually.
- **Out-of-the-box tools**: The service includes tools for file retrieval, code interpretation, and interaction with data sources like Bing, Azure AI Search, and Azure Functions.
- **Model selection**: Developers can choose from various Azure OpenAI models.
- **Enterprise-grade security**: The service ensures data privacy and compliance with secure data handling and keyless authentication.
- **Customizable storage solutions**: Developers can use either platform-managed storage or bring their own Azure Blob storage for full visibility and control.

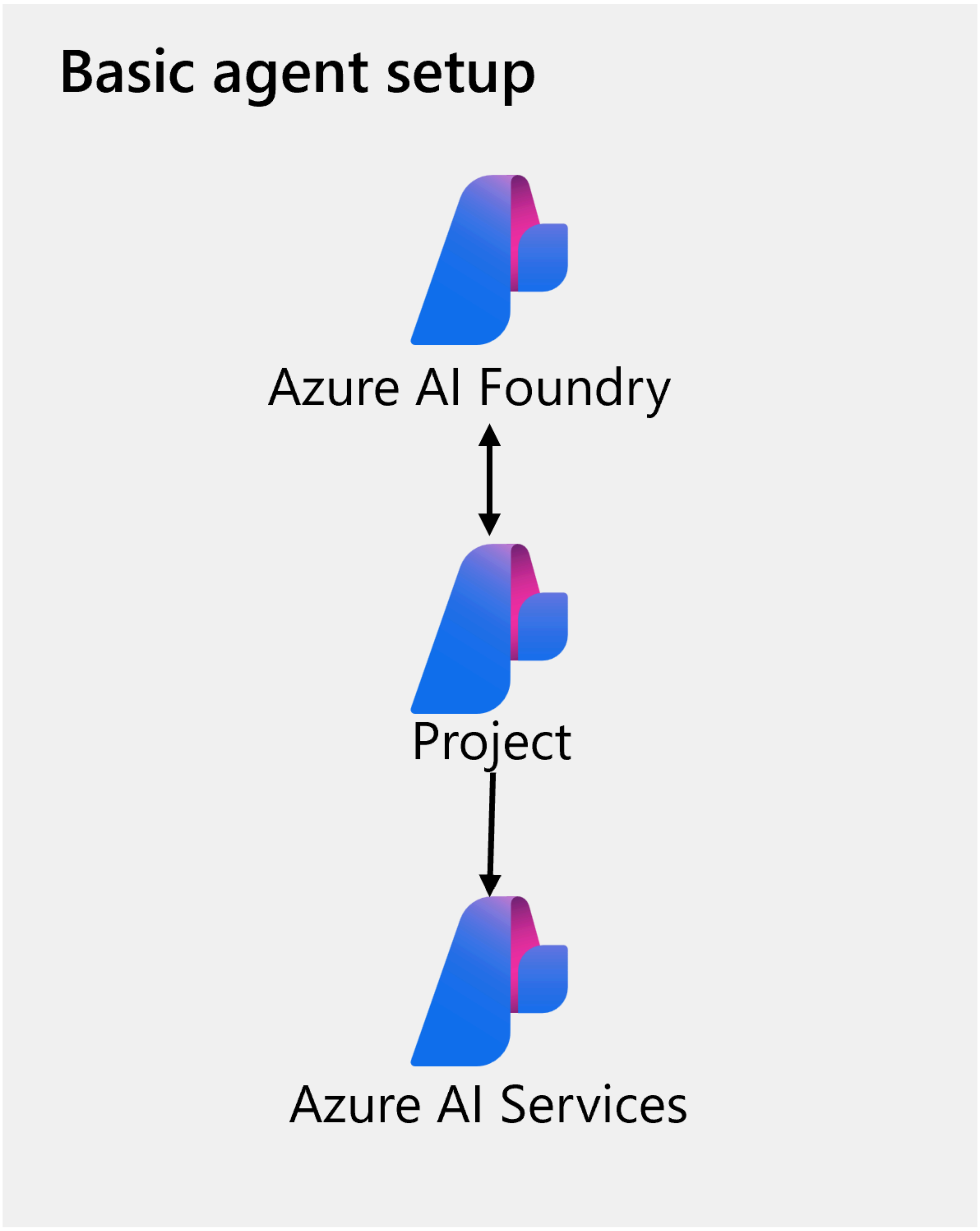
Foundry Agent Service provides a more streamlined and secure way to build and deploy AI agents compared to developing with the Inference API directly.

Foundry Agent Service resources

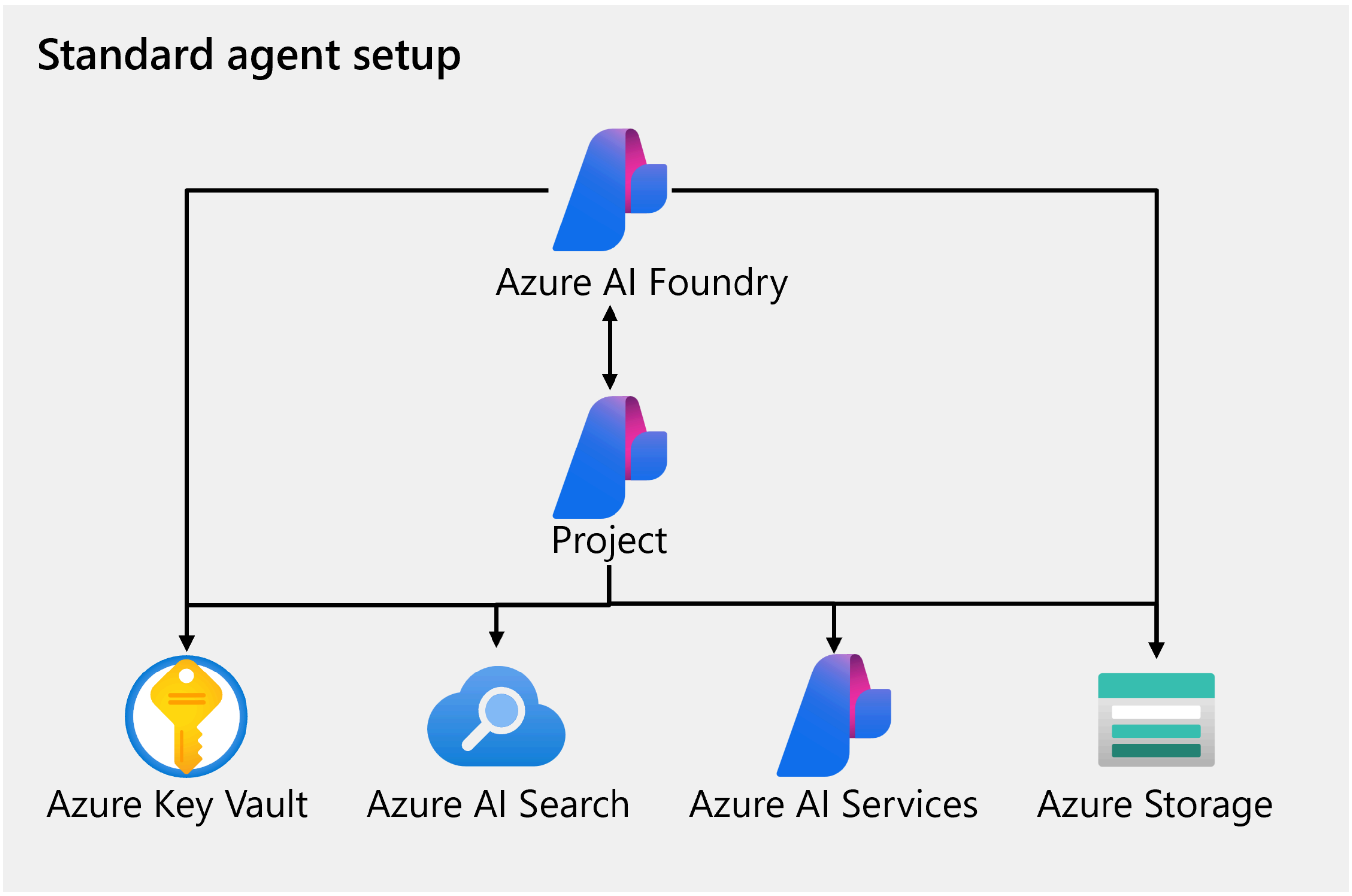
Foundry Agent Service is fully managed and designed to help developers build agents without having to worry about underlying resources. Through Azure, AI Foundry and the Agent Service will provision the necessary cloud resources. If desired, you can choose to connect your own resources when building your agent, giving you the flexibility to utilize Azure however works best for you.

At a minimum, you need to create an Azure AI hub with an Azure AI project for your agent. You can add more Azure services as required. You can create the resources using the Microsoft Foundry portal, or you can use predefined bicep templates to deploy the resources in your subscription. Two common architectures for Foundry Agent Service solutions are:

- **Basic agent setup**: A minimal configuration that includes Azure AI hub, Azure AI project, and Foundry Tools resources.



- **Standard agent setup**: A more comprehensive configuration that includes the basic agent setup plus Azure Key Vault, Azure AI Search, and Azure Storage.



4. Develop agents with the Microsoft Foundry Agent Service

https://learn.microsoft.com/en-us/training/modules/develop-ai-agent-azure/4-when-use-agent-service

Develop agents with the Microsoft Foundry Agent Service

Completed

- 5 minutes

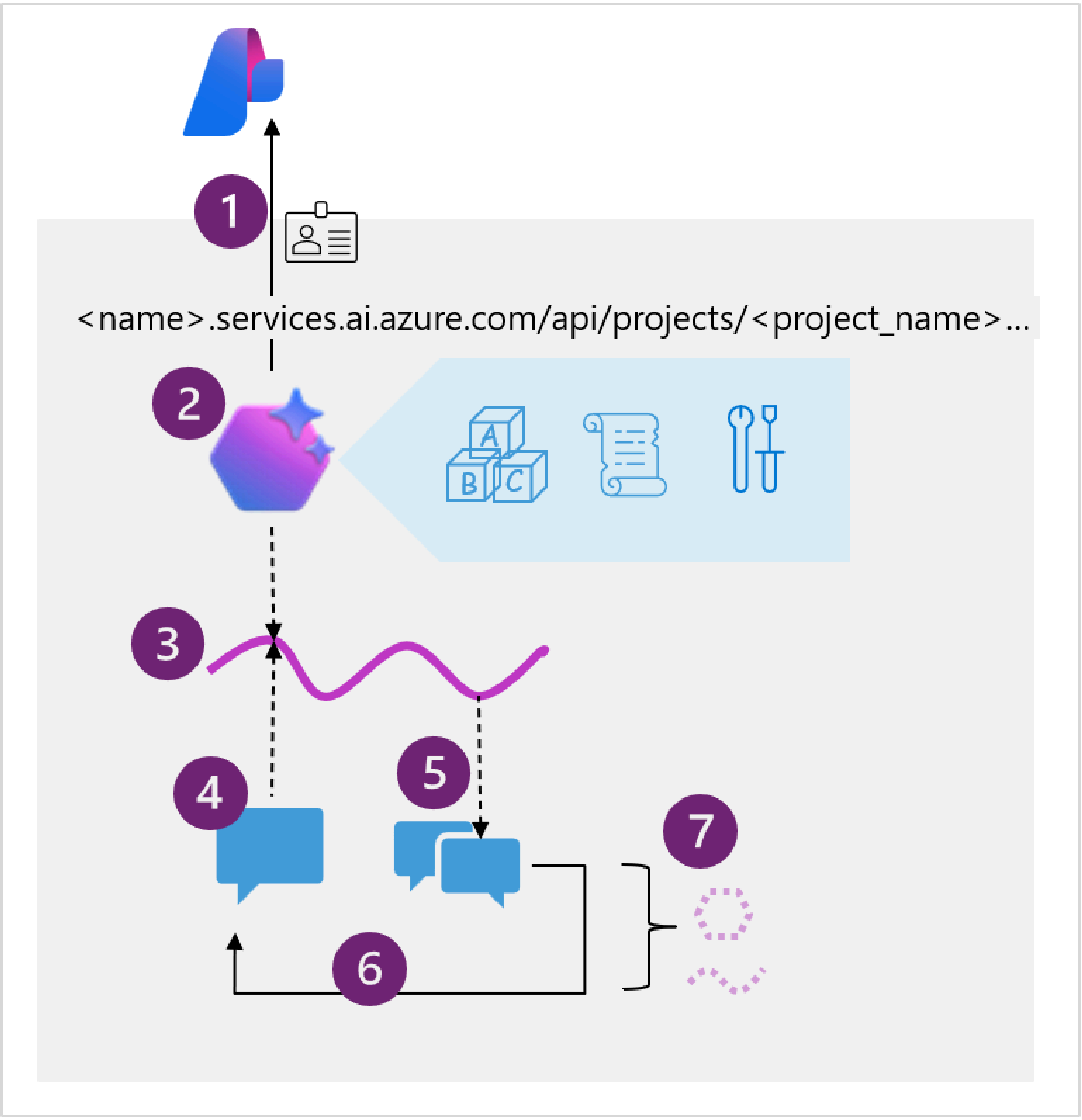
Previous solutions to achieve an agent-like experience took hundreds of lines of code to do things like referencing grounding data or connecting to a custom function. The Agent Service now simplifies all of that, supporting client-side function calling with just a few lines of code and connections to Azure Functions or an OpenAPI defined tool.

Note

> Foundry Agent Service offers several advantages to building agents, but isn't always the right solution for your use case. For example, if you're trying to build an integration with Microsoft 365 you might choose the Copilot Studio lite experience and if you're trying to orchestrate multiple agents, you might choose the Semantic Kernel Agents Framework.
>
> This Fundamentals of AI Agents unit explores more of the options for building agents.

Developing apps that use agents

Foundry Agent Service provides several SDKs and a REST API for you to integrate agents into your app using your preferred programming language. The exercise later in this module focuses on Python, but the overall pattern is the same for REST or other language SDKs.



The diagram shows the following high-level steps that you must implement in your code:

1. Connect to the *AI Foundry project* for your agent, using the project endpoint and Entra ID authentication.
2. Get a reference to an existing agent that you created in the Microsoft Foundry portal, or create a new one specifying:
    - The *model deployment* in the project that the agent should use to interpret and respond to prompts.
    - *Instructions* that determine the functionality and behavior of the agent.
    - *Tools and resources* that the agent can use to perform tasks.
3. Create a *thread* for a chat session with the agent. All conversations with an agent are conducted on a stateful thread that retains message history and data artifacts generated during the chat.
4. Add *messages* to the thread and invoke it with the agent.

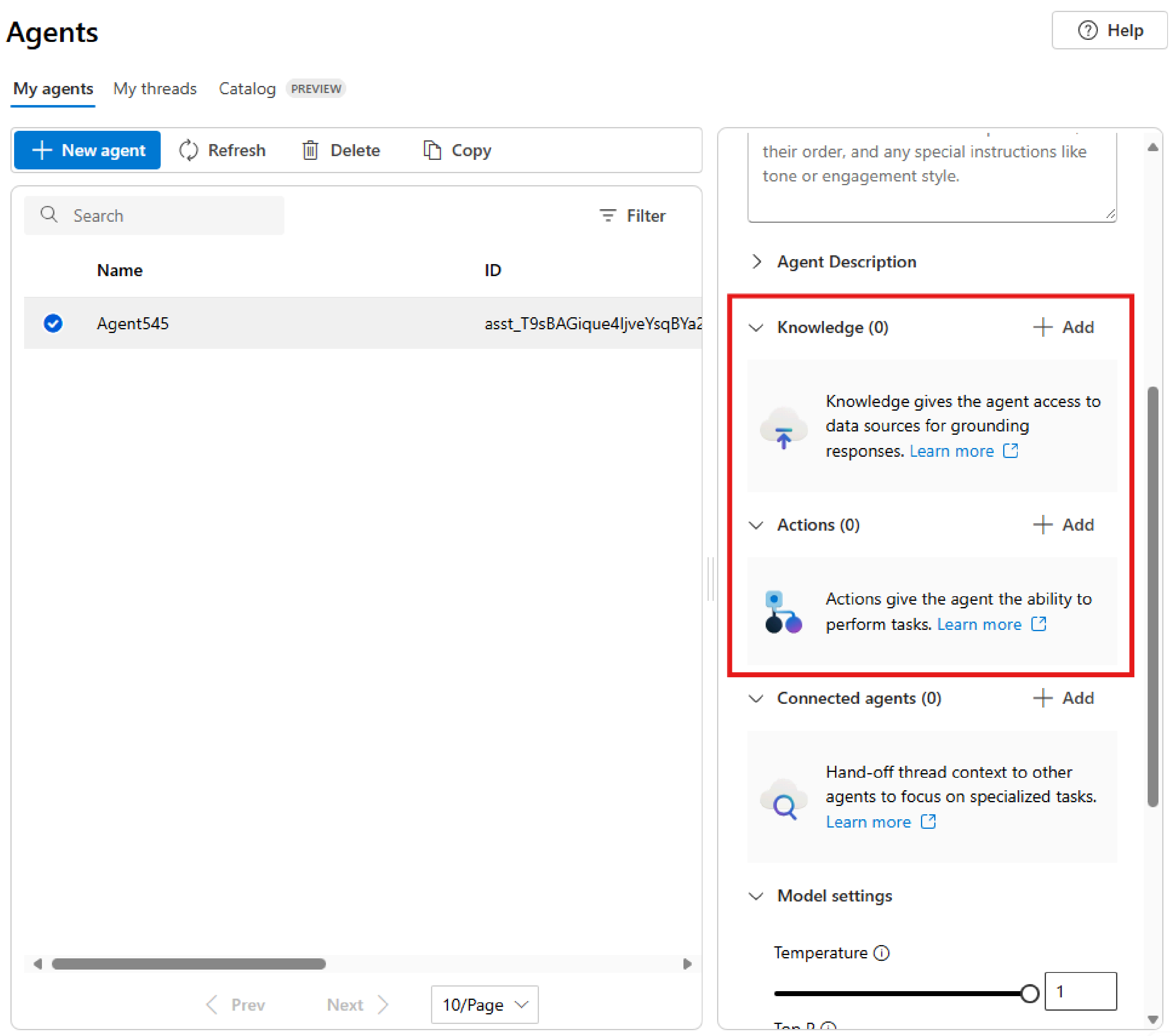
5. Check the thread *status*, and when ready, retrieve the messages and data artifacts.
6. Repeat the previous two steps as a *chat loop* until the conversation can be concluded.
7. When finished, delete the agent and the thread to clean up the resources and delete data that is no longer required.

> **Note**
>
> You'll get a chance to put this pattern into action in the exercise later in this module.

Tools available to your agent

Much of the enhanced functionality of an agent comes from the agent's ability to determine when and how to use *tools*. Tools make additional functionality available to your agent, and if the conversation or task warrants the use of one or more of the tools, the agent calls that tool and handle the response.

You can assign tools when creating an agent in the Microsoft Foundry portal, or when defining an agent in code using the SDK.


Agents

Help

My agents · My threads · Catalog PREVIEW

New agent · Refresh · Delete · Copy

Search · Filter

Name	ID
Agent545	asst_T9sBAGique4ljveYsqBYa2

Prev · Next · 10/Page

their order, and any special instructions like tone or engagement style.

Agent Description

Knowledge (0) + Add

Knowledge gives the agent access to data sources for grounding responses. Learn more

Actions (0) + Add

Actions give the agent the ability to perform tasks. Learn more

Connected agents (0) + Add

Hand-off thread context to other agents to focus on specialized tasks. Learn more

Model settings

Temperature: 1

Top P

For example, one of the tools available is the *code interpreter*. This tool enables your agent to run custom code it writes to achieve something, such as MATLAB code to create a graph or solve a data analytics problem.

Available tools are split into two categories:

Knowledge tools

Knowledge tools enhance the context or knowledge of your agent. Available tools include:

- **Bing Search**: Uses Bing search results to ground prompts with real-time live data from the web.
- **File search**: Grounds prompts with data from files in a vector store.
- **Azure AI Search**: Grounds prompts with data from Azure AI Search query results.
- **Microsoft Fabric**: Uses the Fabric Data Agent to ground prompts with data from your Fabric data stores.

Tip

You can also integrate third-party licensed data by using the OpenAPI Spec action tool (discussed below).

Action tools

Action tools perform an action or run a function. Available tools include:

- **Code Interpreter**: A sandbox for model-generated Python code that can access and process uploaded files.
- **Custom function**: Call your custom function code – you must provide function definitions and implementations.
- **Azure Function**: Call code in serverless Azure Functions.
- **OpenAPI Spec**: Call external APIs based on the OpenAPI 3.0 spec.

By connecting built-in and custom tools, you can allow your agent to perform countless tasks on your behalf.

5. Exercise - Build an AI agent

https://learn.microsoft.com/en-us/training/modules/develop-ai-agent-azure/5-exercise

Exercise - Build an AI agent

Completed

- 30 minutes

Now it's your opportunity to build an agent in Microsoft Foundry. In this exercise, you create an agent and test it in Agent playground. You'll then develop your own app that integrates with an agent through Foundry Agent Service.

Note

If you don't have an Azure subscription, and you want to explore Microsoft Foundry, you can sign up for an account, which includes credits for the first 30 days.

Launch the exercise and follow the instructions.

Launch Exercise

Tip

After completing the exercise, if you're finished exploring Azure AI Agents, delete the Azure resources that you created during the exercise.

6. Module assessment

https://learn.microsoft.com/en-us/training/modules/develop-ai-agent-azure/6-knowledge-check

Module assessment

Completed

- 3 minutes

7. Summary

https://learn.microsoft.com/en-us/training/modules/develop-ai-agent-azure/7-summary

Summary

Completed

- 2 minutes

AI agents represent a significant advancement in the field of artificial intelligence, offering numerous benefits to businesses and individuals alike. By automating routine tasks, enhancing decision-making, and providing scalable solutions, AI agents are transforming how we work and interact with technology. As these agents continue to evolve, their potential applications will only expand, driving further innovation and efficiency across various sectors.

In this module, you learned about the purpose of Foundry Agent Service, its key features, the setup process, and its integration capabilities with other Foundry Tools. We also addressed the challenge of building, deploying, and scaling AI agents. Foundry Agent Service solves several of these challenges, providing a fully managed environment for creating high-quality, extensible AI agents with minimal coding and infrastructure management.

The techniques covered in this module demonstrate several advantages, including automatic tool calling, secure data management, and flexible model selection. These features enable developers to focus on creating intelligent solutions while ensuring enterprise-grade security and compliance. The business impact includes streamlined development processes, reduced operational overhead, and enhanced AI capabilities.

After completing this module, you're now able to:

- Describe the purpose of AI agents
- Explain the key features of Foundry Agent Service
- Build an agent using the Foundry Agent Service
- Integrate an agent in the Foundry Agent Service into your own app

More reading:

- Quickstart Guide for Microsoft Foundry Agent Service
- What's new in Microsoft Foundry Agent Service
- For detailed instructions, see the quickstart guide.