

# Plan and prepare to develop AI solutions on Azure

---

## 1. Introduction

---

<https://learn.microsoft.com/en-us/training/modules/prepare-azure-ai-development/1-introduction>

## Introduction

---

Completed

- 1 minute

The growth in the use of artificial intelligence (AI) in general, and *generative* AI in particular means that developers are increasingly required to create comprehensive AI solutions. These solutions need to combine machine learning models, AI services, prompt engineering solutions, and custom code.

Microsoft Azure provides multiple services that you can use to create AI solutions. However, before embarking on an AI application development project, it's useful to consider the available options for services, tools, and frameworks as well as some principles and practices that can help you succeed.

This module explores some of the key considerations for planning an AI development project, and introduces **Microsoft Foundry**; a comprehensive platform for AI development on Microsoft Azure.

## 2. What is AI?

---

<https://learn.microsoft.com/en-us/training/modules/prepare-azure-ai-development/2-what-is-ai>

## What is AI?

---

## Completed

- 5 minutes

The term "Artificial Intelligence" (AI) covers a wide range of software capabilities that enable applications to exhibit human-like behavior. AI has been around for many years, and its definition has varied as the technology and use cases associated with it have evolved. In today's technological landscape, AI solutions are built on machine learning *models* that encapsulate semantic relationships found in huge quantities of data; enabling applications to appear to interpret input in various formats, reason over the input data, and generate appropriate responses and predictions.

Common AI capabilities that developers can integrate into a software application include:

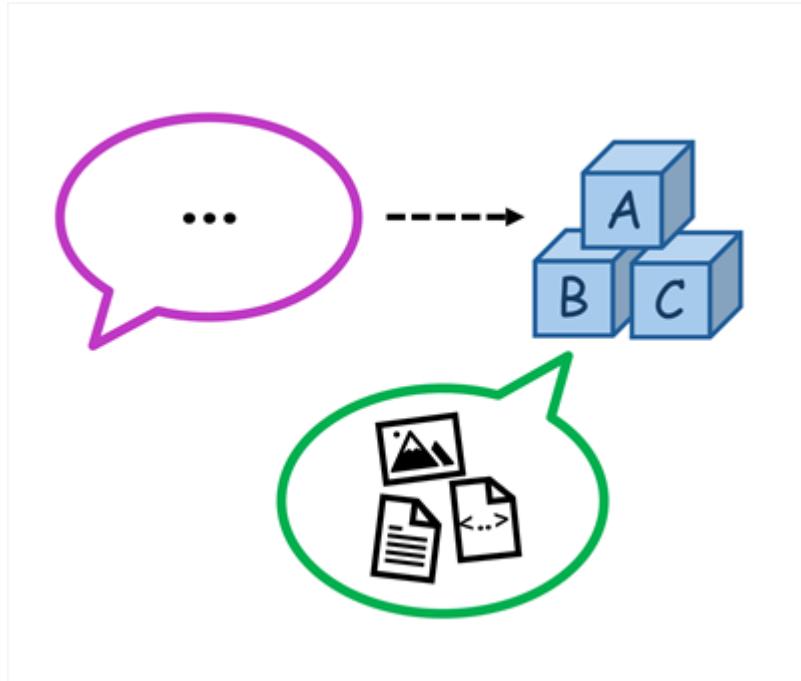
Capability	Description
 <b>Generative AI</b>	<p>The ability to generate original responses to natural language <i>prompts</i>. For example, software for a real estate business might be used to automatically generate property descriptions and advertising copy for a property listing.</p>
 <b>Agents</b>	<p>Generative AI applications that can respond to user input or assess situations autonomously, and take appropriate actions. For example, an "executive assistant" agent could provide details about the location of a meeting on your calendar, or even attach a map or automate the booking of a taxi or rideshare service to help you get there.</p>
 <b>Computer vision</b>	<p>The ability to accept, interpret, and process visual input from images, videos, and live camera streams. For example, an automated checkout in a grocery store might use computer vision to identify which products a customer has in their shopping basket, eliminating the need to scan a barcode or manually enter the product and quantity.</p>
 <b>Speech</b>	<p>The ability to recognize and synthesize speech. For example, a digital assistant might enable users to ask questions or provide audible instructions by speaking into a microphone, and generate spoken output to provide answers or confirmations.</p>

Capability	Description
 <b>Natural language processing</b>	<p>The ability to process natural language in written or spoken form, analyze it, identify key points, and generate summaries or categorizations. For example, a marketing application might analyze social media messages that mention a particular company, translate them to a specific language, and categorize them as positive or negative based on sentiment analysis.</p>
 <b>Information extraction</b>	<p>The ability to use computer vision, speech, and natural language processing to extract key information from documents, forms, images, recordings, and other kinds of content. For example, an automated expense claims processing application might extract purchase dates, individual line item details, and total costs from a scanned receipt.</p>
 <b>Decision support</b>	<p>The ability to use historic data and learned correlations to make predictions that support business decision making. For example, analyzing demographic and economic factors in a city to predict real estate market trends that inform property pricing decisions.</p>

Determining the specific AI capabilities you want to include in your application can help you identify the most appropriate AI services that you'll need to provision, configure, and use in your solution.

## A closer look at generative AI

*Generative AI* represents the latest advance in artificial intelligence, and deserves some extra attention. Generative AI uses *language models* to respond to natural language *prompts*, enabling you to build conversational apps and agents that support research, content creation, and task automation in ways that were previously unimaginable.



The language models used in generative AI solutions can be large language models (LLMs) that have been trained on huge volumes of data and include many millions of parameters; or they can be small language models (SLMs) that are optimized for specific scenarios with lower overhead. Language models commonly respond to text-based prompts with natural language text; though increasingly new *multi-modal* models are able to handle image or speech prompts and respond by generating text, code, speech, or images.

### 3. Foundry Tools

<https://learn.microsoft.com/en-us/training/modules/prepare-azure-ai-development/3-azure-ai-services>

## Foundry Tools

Completed

- 5 minutes

Microsoft Azure provides a wide range of cloud services that you can use to develop, deploy, and manage an AI solution. The most obvious starting point for considering AI development on Azure is Foundry Tools; a set of out-of-the-box prebuilt APIs and models that you can integrate into your applications. The following table lists some commonly used Foundry Tools (for a full list of all available Foundry Tools, see [Available Foundry Tools](#)).

Service	Description
 <b>Azure OpenAI</b>	<p>Azure OpenAI in Foundry Models provides access to OpenAI generative AI models including the GPT family of large and small language models and DALL-E image-generation models within a scalable and securable cloud service on Azure.</p>
 <b>Azure Vision</b>	<p>The Azure Vision service provides a set of models and APIs that you can use to implement common computer vision functionality in an application. With the AI Vision service, you can detect common objects in images, generate captions, descriptions, and tags based on image contents, and read text in images.</p>
 <b>Azure Speech</b>	<p>The Azure Speech service provides APIs that you can use to implement <i>text to speech</i> and <i>speech to text</i> transformation, as well as specialized speech-based capabilities like speaker recognition and translation.</p>
 <b>Azure Language</b>	<p>The Azure Language service provides models and APIs that you can use to analyze natural language text and perform tasks such as entity extraction, sentiment analysis, and summarization. The AI Language service also provides functionality to help you build conversational language models and question answering solutions.</p>
 <b>Microsoft Foundry Content Safety</b>	<p>Microsoft Foundry Content Safety provides developers with access to advanced algorithms for processing images and text and flagging content that is potentially offensive, risky, or otherwise undesirable.</p>
 <b>Azure Translator</b>	<p>The Azure Translator service uses state-of-the-art language models to translate text between a large number of languages.</p>
 <b>Azure AI Face</b>	<p>The Azure AI Face service is a specialist computer vision implementation that can detect, analyze, and recognize human faces. Because of the potential risks associated with personal identification and misuse of this capability, access to some features of the AI Face service are restricted to approved customers.</p>
 <b>Azure AI Custom</b>	<p>The Azure AI Custom Vision service enables you to train and use custom computer vision models for image classification and object detection.</p>

Service	Description
<b>Vision</b>	
 <b>Azure Document Intelligence</b>	With Azure Document Intelligence, you can use pre-built or custom models to extract fields from complex documents such as invoices, receipts, and forms.
 <b>Azure Content Understanding</b>	The Azure Content Understanding service provides multi-modal content analysis capabilities that enable you to build models to extract data from forms and documents, images, videos, and audio streams.
 <b>Azure AI Search</b>	The Azure AI Search service uses a pipeline of AI skills based on other Foundry Tools and custom code to extract information from content and create a searchable index. AI Search is commonly used to create vector indexes for data that can then be used to <i>ground</i> prompts submitted to generative AI language models, such as those provided in Azure OpenAI.

## Considerations for Foundry Tools resources

To use Foundry Tools, you create one or more Azure AI resources in an Azure subscription and implement code in client applications to consume them. In some cases, AI services include web-based visual interfaces that you can use to configure and test your resources - for example to train a custom image classification model using the **Custom Vision** service you can use the visual interface to upload training images, manage training jobs, and deploy the resulting model.

### Note

You can provision Foundry Tools resources in the Azure portal (or by using BICEP or ARM templates or the Azure command-line interface) and build applications that use them directly through various service-specific APIs and SDKs. However, as we'll discuss later in this module, in most medium to large-scale development scenarios it's better to provision Foundry Tools resources as part of an *Microsoft Foundry* project - enabling you to centralize access control and cost management, and making it easier to manage shared resources and build the next generation of generative AI apps and agents.

## Single service or Foundry Tools resource?

Most Foundry Tools, such as **Azure Vision**, **Azure Language**, and so on, can be provisioned as standalone resources, enabling you to create only the Azure resources you specifically need. Additionally, standalone Foundry Tools often include a free-tier SKU with limited functionality, enabling you to evaluate and develop with the service at no cost. Each standalone Azure AI resource provides an endpoint and authorization keys that you can use to access it securely from a client application.

Alternatively, you can provision a Foundry Tools resource that encapsulates multiple AI services in a single Azure resource. Using a Foundry Tools resource can make it easier to manage applications that use multiple AI capabilities. There are two Foundry resource types you can use:

Resource	Description
 <b>Foundry Tools</b>	<p>The Foundry Tools resource type includes the following services, making them available from a single endpoint:</p> <ul style="list-style-type: none"><li>• Azure Speech</li><li>• Azure Language</li><li>• Azure Translator</li><li>• Azure Vision</li><li>• Azure AI Face</li><li>• Azure AI Custom Vision</li><li>• Azure Document Intelligence</li></ul>
 <b>Microsoft Foundry</b>	<p>The Microsoft Foundry resource type includes the following services, and supports working with them through a Microsoft Foundry project*:</p> <ul style="list-style-type: none"><li>• Azure OpenAI</li><li>• Azure Speech</li><li>• Azure Language</li><li>• Microsoft Foundry Content Safety</li><li>• Azure Translator</li><li>• Azure Vision</li><li>• Azure AI Face</li><li>• Azure Document Intelligence</li><li>• Azure Content Understanding</li></ul>

\* Microsoft Foundry is discussed in the next unit.

## Regional availability

Some services and models are available in only a subset of Azure regions. Consider service availability and any regional quota restrictions for your subscription when provisioning Foundry Tools. Use the

[product availability table](#) to check regional availability of Azure services. Use the [model availability table](#) in the Azure OpenAI documentation to determine regional availability for Azure OpenAI models.

## Cost

Foundry Tools are charged based on usage, with different pricing schemes available depending on the specific services being used. As you plan an AI solution on Azure, use the [Foundry Tools pricing](#) documentation to understand pricing for the AI services you intend to incorporate into your application. You can use the [Azure pricing calculator](#) to estimate the costs your expected usage will incur.

## 4. Microsoft Foundry

<https://learn.microsoft.com/en-us/training/modules/prepare-azure-ai-development/4-azure-ai-foundry>

# Microsoft Foundry

Completed

- 5 minutes

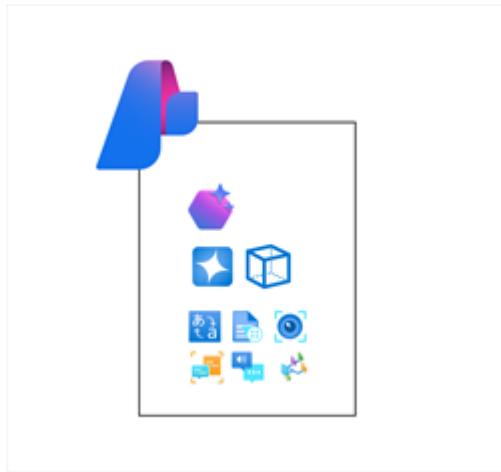
Microsoft Foundry is a platform for AI development on Microsoft Azure. While you *can* provision individual Foundry Tools resources and build applications that consume them without it, the project organization, resource management, and AI development capabilities of Microsoft Foundry makes it the recommended way to build all but the most simple solutions.

Microsoft Foundry provides the *Microsoft Foundry portal*, a web-based visual interface for working with AI projects. It also provides the *Microsoft Foundry SDK*, which you can use to build AI solutions programmatically.

## Microsoft Foundry projects

In Microsoft Foundry, you manage the resource connections, data, code, and other elements of the AI solution in *projects*. There are two kinds of project:

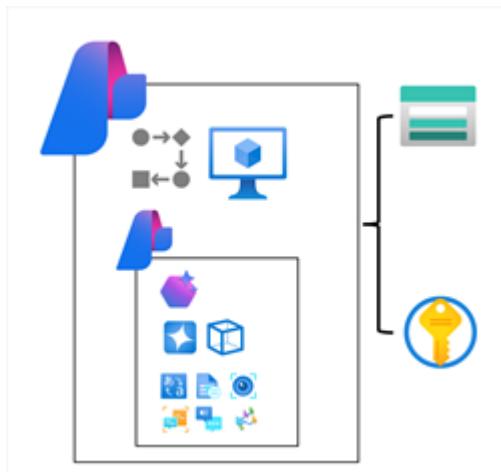
## Foundry projects



*Foundry projects* are associated with a **Microsoft Foundry** resource in an Azure subscription. Foundry projects provide support for Microsoft Foundry Models (including OpenAI models), Microsoft Foundry Agent Service, Foundry Tools, and tools for evaluation and responsible AI development.

A Microsoft Foundry resource supports the most common AI development tasks to develop generative AI chat apps and agents. In most cases, using a Foundry project provides the right level of resource centralization and capabilities with a minimal amount of administrative resource management. You can use Microsoft Foundry portal to work in projects that are based in Microsoft Foundry resources, making it easy to add connected resources and manage model and agent deployments.

## Hub-based projects



*Hub-based projects* are associated with an **Azure AI hub** resource in an Azure subscription. Hub-based projects include a Microsoft Foundry resource, as well as managed compute, support for Prompt Flow development, and connected **Azure storage** and **Azure key vault** resources for secure data storage.

Azure AI hub resources support advanced AI development scenarios, like developing Prompt Flow based applications or fine-tuning models. You can also use Azure AI hub resources in both Microsoft

Foundry portal and Azure Machine learning portal, making it easier to work on collaborative projects that involve data scientists and machine learning specialists as well as developers and AI software engineers

### Tip

For more information about Microsoft Foundry project types, see [What is Microsoft Foundry?](#).

## 5. Developer tools and SDKs

<https://learn.microsoft.com/en-us/training/modules/prepare-azure-ai-development/5-tools-and-sdks>

## Developer tools and SDKs

Completed

- 5 minutes

While you can perform many of the tasks needed to develop an AI solution directly in the Microsoft Foundry portal, developers also need to write, test, and deploy code.

## Development tools and environments

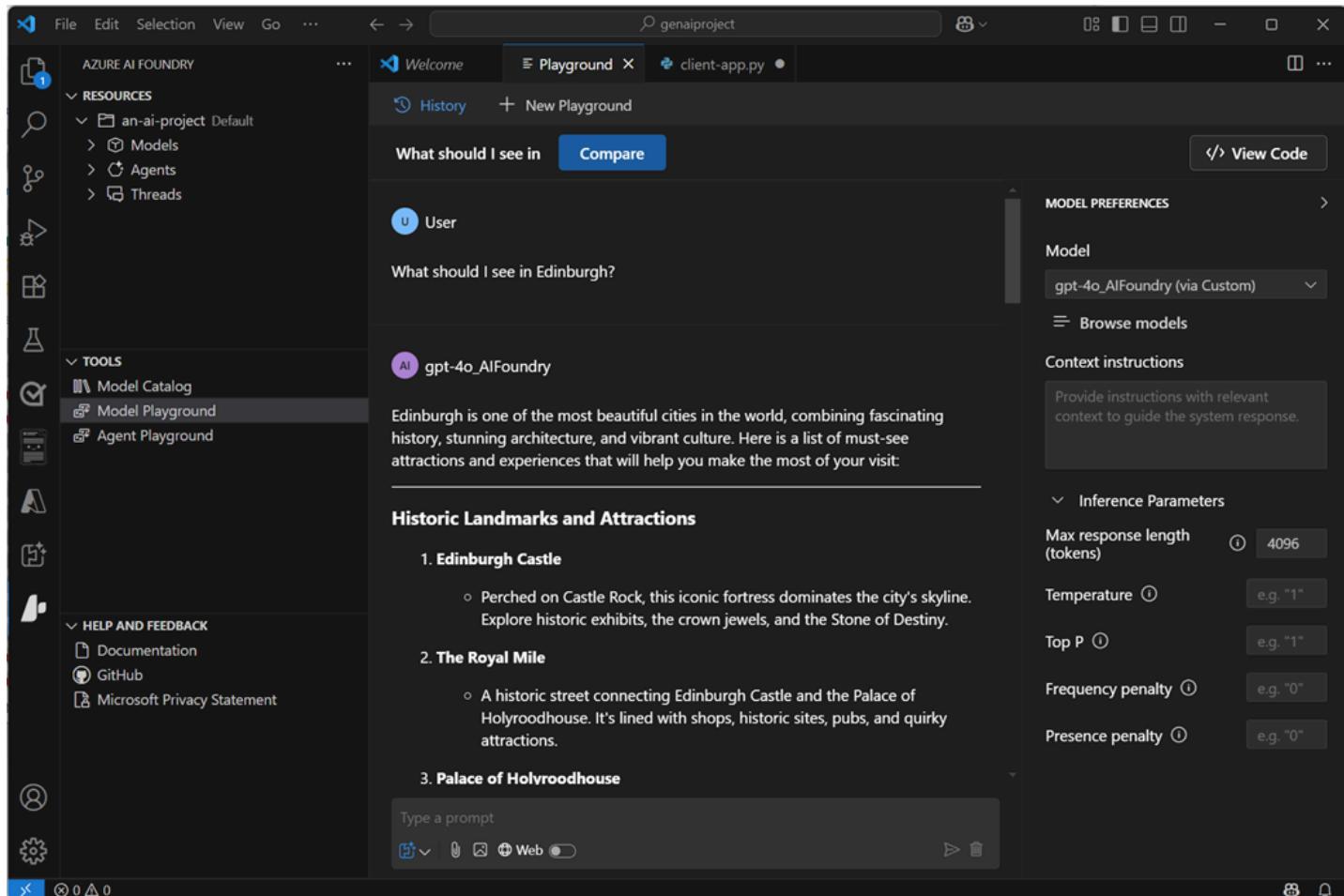
There are many development tools and environments available, and developers should choose one that supports the languages, SDKs, and APIs they need to work with and with which they're most comfortable. For example, a developer who focuses strongly on building applications for Windows using the .NET Framework might prefer to work in an integrated development environment (IDE) like Microsoft Visual Studio. Conversely, a web application developer who works with a wide range of open-source languages and libraries might prefer to use a code editor like Visual Studio Code (VS Code). Both of these products are suitable for developing AI applications on Azure.

### The Microsoft Foundry for Visual Studio Code extension

When developing Microsoft Foundry based generative AI applications in Visual Studio Code, you can use the Microsoft Foundry for Visual Studio Code extension to simplify key tasks in the workflow,

including:

- Creating a project.
- Selecting and deploying a model.
- Testing a model in the playground.
- Creating an agent.



### Tip

For more information about using the Microsoft Foundry for Visual Studio Code extension, see [Work with the Microsoft Foundry for Visual Studio Code extension](#).

## GitHub and GitHub Copilot

GitHub is the world's most popular platform for source control and DevOps management, and can be a critical element of any team development effort. Visual Studio and VS Code both provide native integration with GitHub, and access to GitHub Copilot; an AI assistant that can significantly improve developer productivity and effectiveness.

The screenshot shows the Visual Studio Code interface with the following details:

- Left Sidebar:** Shows the "AZURE AI FOUNDRY" workspace with sections for RESOURCES (an-ai-project Default), TOOLS (Model Catalog, Model Playground, Agent Playground), and HELP AND FEEDBACK (Documentation, GitHub, Microsoft Privacy Statement).
- Top Bar:** Includes File, Edit, Selection, View, Go, and a search bar for "genaiproject".
- Central Area:** A code editor window titled "client-app.py" containing Python code for interacting with an AI travel assistant. The code uses the OpenAI API to generate a completion based on a system message and user input.
- Right Panel:** The GitHub Copilot interface. It displays a message from the AI: "Update this code to add a system message instructing the model to behave as an AI travel assistant that provides guidance to tourists." Below this, it says "I'll update the code to add a system message that instructs the model to behave as an AI travel assistant. Let me modify the messages array in your OpenAI client code." The AI then performs the modification, updating the code to include a system message. The status bar at the bottom right indicates "1 file ch... Keep Undo | 1 of 1 ↑ ↓".

## Tip

For more information about using GitHub Copilot in Visual Studio Code, see [GitHub Copilot in VS Code](#).

## Programming languages, APIs, and SDKs

You can develop AI applications using many common programming languages and frameworks, including Microsoft C#, Python, Node, TypeScript, Java, and others. When building AI solutions on Azure, some common SDKs you should plan to install and use include:

- The [Microsoft Foundry SDK](#), which enables you to write code to connect to Microsoft Foundry projects and access resource connections, which you can then work with using service-specific SDKs.
- The [Microsoft Foundry Models API](#), which provides an interface for working with generative AI model endpoints hosted in Microsoft Foundry.
- The [Azure OpenAI in Microsoft Foundry Models API](#), which enables you to build chat applications based on OpenAI models hosted in Microsoft Foundry.
- [Foundry Tools SDKs](#) - AI service-specific libraries for multiple programming languages and frameworks that enable you to consume Foundry Tools resources in your subscription. You can

also use Foundry Tools through their [REST APIs](#).

- The [\*\*Microsoft Foundry Agent Service\*\*](#), which is accessed through the Microsoft Foundry SDK and can be integrated with frameworks like [Semantic Kernel](#) to build comprehensive AI agent solutions.

## 6. Responsible AI

<https://learn.microsoft.com/en-us/training/modules/prepare-azure-ai-development/6-responsible-ai>

# Responsible AI

Completed

- 5 minutes

It's important for software engineers to consider the impact of their software on users, and society in general; including considerations for its responsible use. When the application is imbued with artificial intelligence, these considerations are particularly important due to the nature of how AI systems work and inform decisions; often based on probabilistic models, which are in turn dependent on the data with which they were trained.

The human-like nature of AI solutions is a significant benefit in making applications user-friendly, but it can also lead users to place a great deal of trust in the application's ability to make correct decisions. The potential for harm to individuals or groups through incorrect predictions or misuse of AI capabilities is a major concern, and software engineers building AI-enabled solutions should apply due consideration to mitigate risks and ensure fairness, reliability, and adequate protection from harm or discrimination.

Let's discuss some core principles for responsible AI that have been adopted at Microsoft.

## Fairness



AI systems should treat all people fairly. For example, suppose you create a machine learning model to support a loan approval application for a bank. The model should make predictions of whether or not the loan should be approved without incorporating any bias based on gender, ethnicity, or other factors that might result in an unfair advantage or disadvantage to specific groups of applicants.

Fairness of machine learned systems is a highly active area of ongoing research, and some software solutions exist for evaluating, quantifying, and mitigating unfairness in machine learned models.

However, tooling alone isn't sufficient to ensure fairness. Consider fairness from the beginning of the application development process; carefully reviewing training data to ensure it's representative of all potentially affected subjects, and evaluating predictive performance for subsections of your user population throughout the development lifecycle.

## Reliability and safety

---



AI systems should perform reliably and safely. For example, consider an AI-based software system for an autonomous vehicle; or a machine learning model that diagnoses patient symptoms and recommends prescriptions. Unreliability in these kinds of system can result in substantial risk to human life.

As with any software, AI-based software application development must be subjected to rigorous testing and deployment management processes to ensure that they work as expected before release. Additionally, software engineers need to take into account the probabilistic nature of machine learning models, and apply appropriate thresholds when evaluating confidence scores for predictions.

## Privacy and security

---



AI systems should be secure and respect privacy. The machine learning models on which AI systems are based rely on large volumes of data, which may contain personal details that must be kept private. Even after models are trained and the system is in production, they use new data to make

predictions or take action that may be subject to privacy or security concerns; so appropriate safeguards to protect data and customer content must be implemented.

## Inclusiveness

---



AI systems should empower everyone and engage people. AI should bring benefits to all parts of society, regardless of physical ability, gender, sexual orientation, ethnicity, or other factors.

One way to optimize for inclusiveness is to ensure that the design, development, and testing of your application includes input from as diverse a group of people as possible.

## Transparency

---



AI systems should be understandable. Users should be made fully aware of the purpose of the system, how it works, and what limitations may be expected.

For example, when an AI system is based on a machine learning model, you should generally make users aware of factors that may affect the accuracy of its predictions, such as the number of cases used to train the model, or the specific features that have the most influence over its predictions. You should also share information about the confidence score for predictions.

When an AI application relies on personal data, such as a facial recognition system that takes images of people to recognize them; you should make it clear to the user how their data is used and retained, and who has access to it.

## Accountability

---



People should be accountable for AI systems. Although many AI systems seem to operate autonomously, ultimately it's the responsibility of the developers who trained and validated the models they use, and defined the logic that bases decisions on model predictions to ensure that the overall system meets responsibility requirements. To help meet this goal, designers and developers of AI-based solution should work within a framework of governance and organizational principles that ensure the solution meets responsible and legal standards that are clearly defined.

### Tip

For more information about Microsoft's principles for responsible AI, see [the Microsoft responsible AI site](#).

## 7. Exercise - Prepare for an AI development project

<https://learn.microsoft.com/en-us/training/modules/prepare-azure-ai-development/7-exercise-explore-ai-foundry>

## Exercise - Prepare for an AI development project

Completed

- 30 minutes

If you have an Azure subscription, you can explore Microsoft Foundry for yourself.

### 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](#)

## 8. Module assessment

<https://learn.microsoft.com/en-us/training/modules/prepare-azure-ai-development/8-knowledge-check>

# Module assessment

Completed

- 3 minutes

## 9. Summary

<https://learn.microsoft.com/en-us/training/modules/prepare-azure-ai-development/9-summary>

# Summary

Completed

- 1 minute

In this module, you explored some of the key considerations when planning and preparing for AI application development. You've also had the opportunity to become familiar with Microsoft Foundry, the recommended platform for developing AI solutions on Azure.

### Tip

For latest news and information about developing AI applications on Azure, see [Azure AI](#).

