Lab-1-Explore AI Agent development

In this exercise, you use the Azure AI Agent service tools in the Azure AI Foundry portal to create a simple AI agent that answers questions about expense claims.

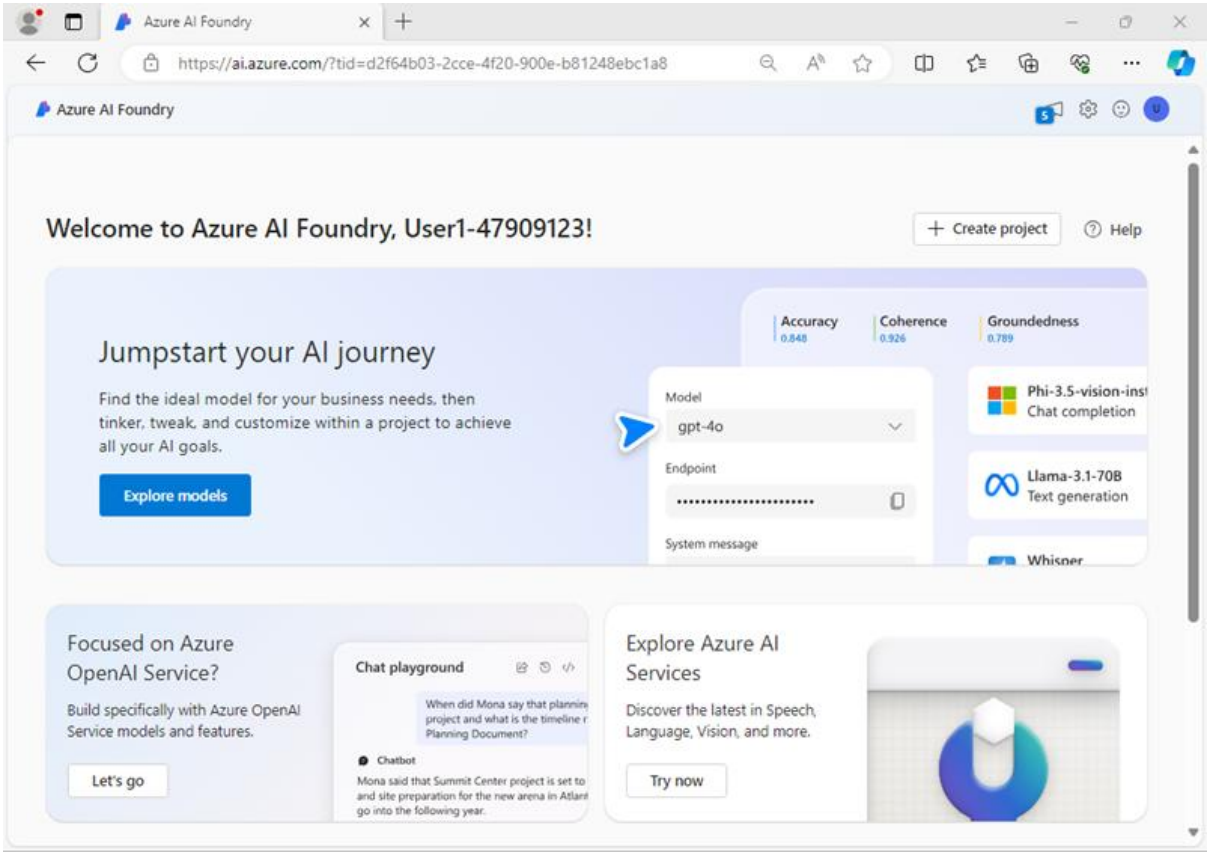
This exercise takes approximately **30** minutes.

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.

Create an Azure AI Foundry project

Let's start by creating an Azure AI Foundry project.

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):

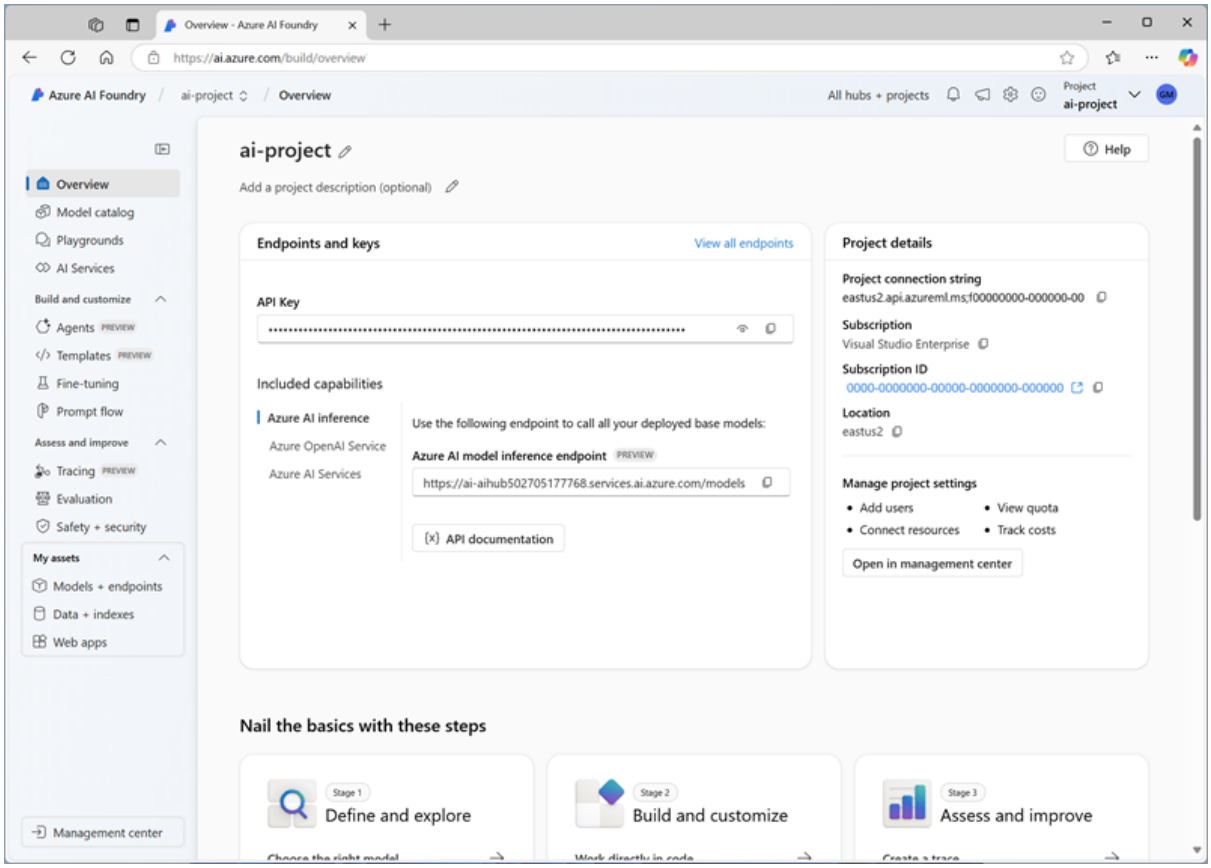


2. In the home page, select **+ Create project**.

3. In the **Create a project** wizard, enter a valid name for your project and if an existing hub is suggested, choose the option to create a new one. Then review the Azure resources that will be automatically created to support your hub and project.
4. Select **Customize** and specify the following settings for your hub:
    - **Hub name**: *A valid name for your hub*
    - **Subscription**: *Your Azure subscription*
    - **Resource group**: *Create or select a resource group*
    - **Location**: Select a region from the following:*
        - eastus
        - eastus2
        - swedencentral
        - westus
        - westus3
    - **Connect Azure AI Services or Azure OpenAI**: *Create a new AI Services resource*
    - **Connect Azure AI Search**: Skip connecting

    > * At the time of writing, these regions support the gpt-4o model for use in agents. Model availability is constrained by regional quotas. In the event of a quota limit being reached later in the exercise, there's a possibility you may need to create another project in a different region.

5. Select **Next** and review your configuration. Then select **Create** and wait for the process to complete.
6. When your project is created, close any tips that are displayed and review the project page in Azure AI Foundry portal, which should look similar to the following image:



Deploy a generative AI model

Now you're ready to deploy a generative AI language model to support your agent.

1. In the pane on the left for your project, in the **My assets** section, select the **Models + endpoints** page.
2. In the **Models + endpoints** page, in the **Model deployments** tab, in the **+ Deploy model** menu, select **Deploy base model**.
3. Search for the **gpt-4o** model in the list, and then select and confirm it.
4. Deploy the model with the following settings by selecting **Customize** in the deployment details:
    - **Deployment name**: *A valid name for your model deployment*
    - **Deployment type**: Global Standard
    - **Automatic version update**: Enabled
    - **Model version**: *Select the most recent available version*
    - **Connected AI resource**: *Select your Azure OpenAI resource connection*
    - **Tokens per Minute Rate Limit (thousands)**: 50K *(or the maximum available in your subscription if less than 50K)*
    - **Content filter**: DefaultV2

> **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 need to wait and resubmit prompts if the rate limit is exceeded.

5. Wait for the deployment to complete.

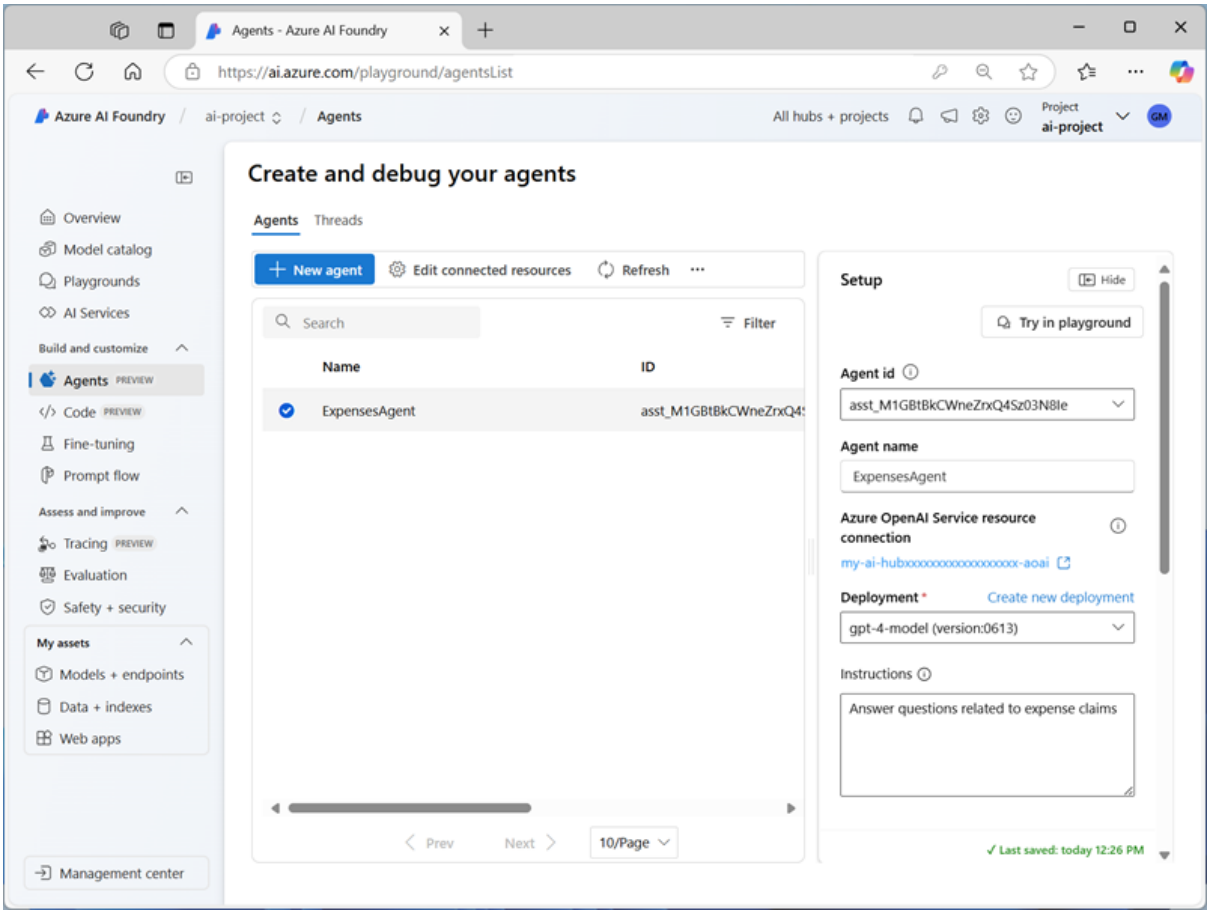
Create an AI agent

Now that you have a model deployed, you're ready to build an AI agent. In this exercise, you'll build a simple agent that answers questions based on a corporate expenses policy. You'll download the expenses policy document, and use it as *grounding* data for the agent.

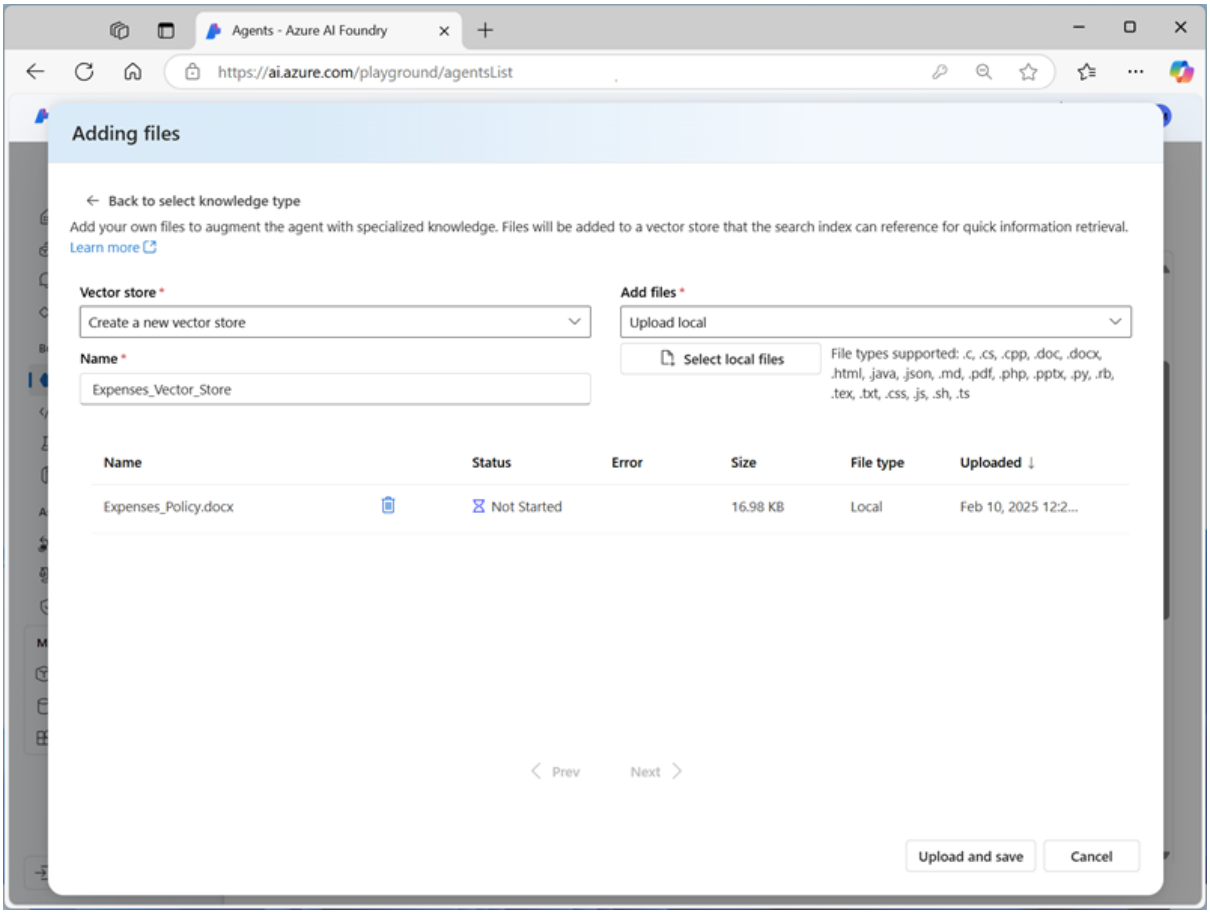
1. Open another browser tab, and download Expenses_policy.docx from `https://raw.githubusercontent.com/MicrosoftLearning/mslearn-ai-agents/main/Labfiles/01-agent-fundamentals/Expenses_Policy.docx` and save it locally. This document contains details of the expenses policy for the fictional Contoso corporation.
2. Return to the browser tab containing the Azure AI Foundry portal, and in the navigation pane on the left, in the **Build and customize** section, select the **Agents** page.
3. If prompted, select your Azure OpenAI service resource and go.

   A new agent with a name like *Agent123* should be created automatically (if not, use the **+ New agent** button to create one).

4. Select your new agent. Then, in the **Setup** pane for your new agent, set the **Agent name** to `ExpensesAgent`, ensure that the gpt-4o model deployment you created previously is selected, and set the **Instructions** to `Answer questions related to expense claims`.



5. Further down in the **Setup** pane, next to the **Knowledge** header, select **+ Add**. Then in the **Add knowledge** dialog box, select **Files**.
6. In the **Adding files** dialog box, create a new vector store named `Expenses_Vector_Store`, uploading and saving the **Expenses_policy.docx** local file that you downloaded previously.



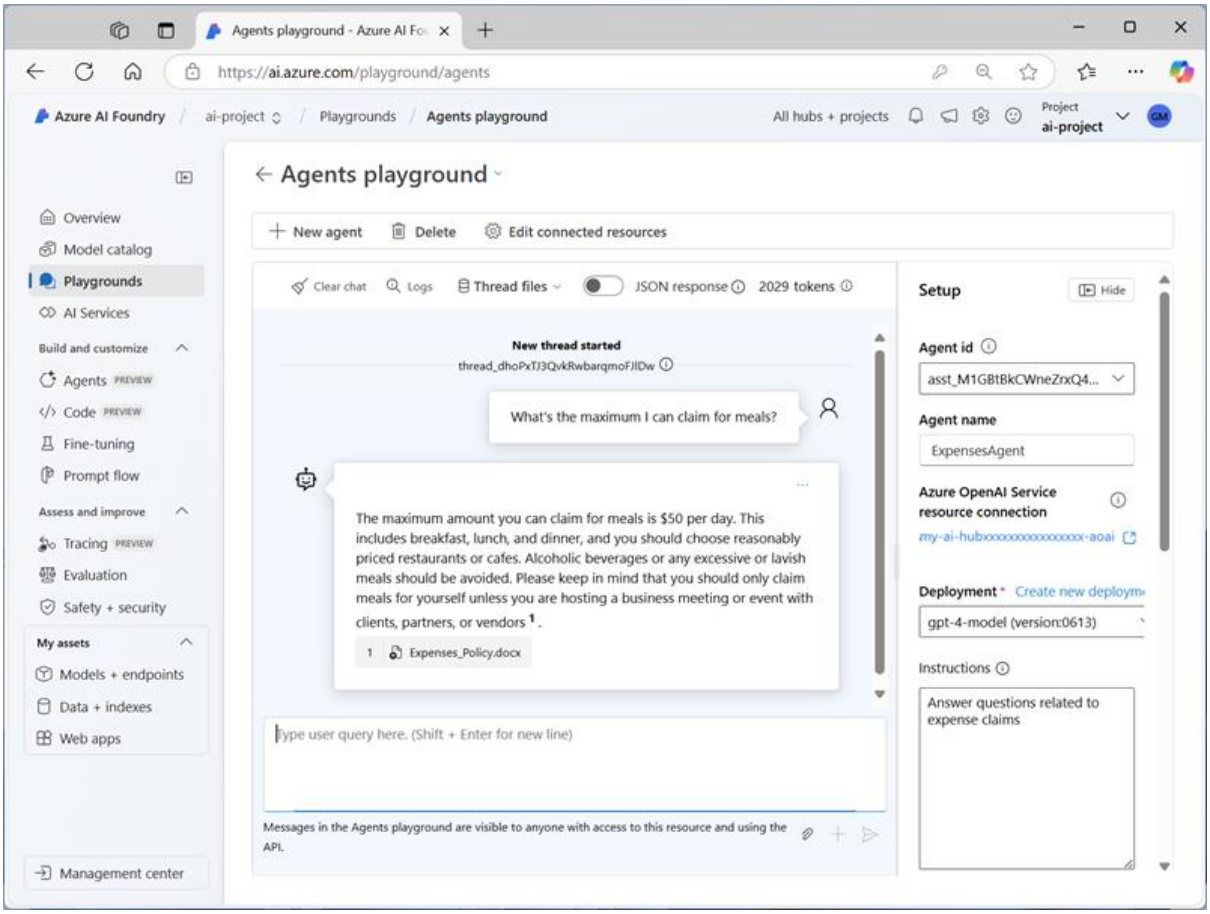
7. In the **Setup** pane, in the **Knowledge** section, verify that **Expenses_Vector_Store** is listed and shown as containing 1 file.

> **Note**: You can also add **Actions** to an agent to automate tasks. In this simple information retrieval agent example, no actions are required.

Test your agent

Now that you've created an agent, you can test it in the Azure AI Foundry portal playground.

1. At the top of the **Setup** pane for your agent, select **Try in playground**.
2. In the playground, enter the prompt `What's the maximum I can claim for meals?` and review the agent's response - which should be based on information in the expenses policy document you added as knowledge to the agent setup.



Note: If the agent fails to respond because the rate limit is exceeded. Wait a few seconds and try again. If there is insufficient quota available in your subscription, the model may not be able to respond.

3. Try a follow-up question, like `What about accommodation?` and review the response.

Clean up

Now that you've finished the exercise, you should delete the cloud resources you've created to avoid unnecessary resource usage.

1. Open the Azure portal at `https://portal.azure.com` and view the contents of the resource group where you deployed the hub 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.