


# Create a Question Answering Solution

One of the most common conversational scenarios is providing support through a knowledge base of frequently asked questions (FAQs). Many organizations publish FAQs as documents or web pages, which works well for a small set of question and answer pairs, but large documents can be difficult and time-consuming to search.

The **Language** service includes a *question answering* capability that enables you to create a knowledge base of question and answer pairs that can be queried using natural language input, and is most commonly used as a resource that a bot can use to look up answers to questions submitted by users.

 **Note:** The question answering capability in the Language service is a newer version of the QnA Maker service - which is still available as a separate service.

## Clone the repository for this course

If you have not already cloned **AI-102-AIEngineer** code repository to the environment where you're working on this lab, follow these steps to do so. Otherwise, open the cloned folder in Visual Studio Code.

1. Start Visual Studio Code.
2. Open the palette (SHIFT+CTRL+P) and run a **Git: Clone** command to clone the `https://github.com/MicrosoftLearning/AI-102-AIEngineer` repository to a local folder (it doesn't matter which folder).
3. When the repository has been cloned, open the folder in Visual Studio Code.
4. Wait while additional files are installed to support the C# code projects in the repo.

 **Note:** If you are prompted to add required assets to build and debug, select **Not Now**.

## Create a Language resource

To create and host a knowledge base for question answering, you need a **Language service** resource in your Azure subscription.

1. Open the Azure portal at `https://portal.azure.com`, and sign in using the Microsoft account associated with your Azure subscription.
2. Select the **+ Create a resource** button, search for *Language*, and create a **Language service** resource.
3. Click **Select** on the **Custom question answering** block. Then click **Continue to create your resource**. You will need to enter the following settings:
  - **Subscription:** *Your Azure subscription*
  - **Resource group:** *Choose or create a resource group (if you are using a restricted subscription, you may not have permission to create a new resource group - use the one provided)*
  - **Region:** *Choose any available location*
  - **Name:** *Enter a unique name*
  - **Pricing tier:** *Standard S*
  - **Azure Search location\*:** *Choose a location in the same global region as your Language resource.*
  - **Azure Search pricing tier:** *Free (F) (If this tier is not available, select Basic (B))*
  - **Legal Terms:** *Agree*
  - **Responsible AI Notice:** *Agree*

\*Custom Question Answering uses Azure Search to index and query the knowledge base of questions and answers.

4. Wait for deployment to complete, and then view the deployment details.

## Create a question answering project

To create a knowledge base for question answering in your Language resource, you can use the Language Studio portal to create a question answering project. In this case, you'll create a knowledge base containing questions and answers about [Microsoft Learn](#).

1. In a new browser tab, go to the *Language Studio* portal at <https://language.azure.com> and sign in using the Microsoft account associated with your Azure subscription.
2. If prompted to choose a Language resource, select the following settings:
  - **Azure Directory:** The Azure directory containing your subscription.
  - **Azure subscription:** Your Azure subscription.
  - **Language resource:** The Language resource you created previously.
3. If you are not prompted to choose a language resource, it may be because you have multiple Language resources in your subscription; in which case:
  - a. On the bar at the top of the page, click the **Settings (⚙)** button.
  - b. On the **Settings** page, view the **Resources** tab.
  - c. Select the language resource you just created, and click **Switch resource**.
  - d. At the top of the page, click **Language Studio** to return to the Language Studio home page.
4. At the top of the portal, in the **Create new** menu, select **Custom question answering**.
5. In the **\*Create a project** wizard, on the **Choose language setting** page, select the option to set the language for all projects in this resource, and select **English** as the language. Then click **Next**.
6. On the **Enter basic information** page, enter the following details and then click **Next**:
  - **Name** LearnFAQ
  - **Description:** FAQ for Microsoft Learn
  - **Default answer when no answer is returned:** Sorry, I don't understand the question
7. On the **Review and finish** page, click **Create project**.

## Add a sources to the knowledge base

You can create a knowledge base from scratch, but it's common to start by importing questions and answers from an existing FAQ page or document. In this case, you'll import data from an existing FAQ web page for Microsoft learn, and you'll also import some pre-defined "chit chat" questions and answers to support common conversational exchanges.

1. On the **Manage sources** page for your question answering project, in the **+ Add source** list, select **URLs**. Then in the **Add URLs** dialog box, click **+ Add url** and add the following URL, and then click **Add all** to add it to the knowledge base:
  - **Name:** [Learn FAQ Page](#)
  - **URL:** <https://docs.microsoft.com/en-us/learn/support/faq>
2. On the **Manage sources** page for your question answering project, in the **+ Add source** list, select **Chitchat**. The in the **Add chit chat** dialog box, select **Friendly** and click **Add chit chat**.

## Edit the knowledge base

Your knowledge base has been populated with question and answer pairs from the Microsoft Learn FAQ, supplemented with a set of conversational *chit-chat* question and answer pairs. You can extend the knowledge base by adding additional question and answer pairs.

1. In the **LearnFAQ** project in Language Studio, select the **Edit knowledge base** page to see the existing question and answer pairs (if some tips are displayed, read them and click **Got it** to dismiss them, or click **Skip all**).
2. In the knowledge base, select **+ Add question pair**.
3. In the **Question** box, type [What is Microsoft certification?](#) and press **Enter**.
4. Select **+ Add alternative phrasing** and type [How can I demonstrate my Microsoft technology skills?](#) and press **Enter**.

5. In the **Answer** box, type

The Microsoft Certified Professional program enables you to validate and prove your skills with Microsoft technologies.

Then press **Enter** and click **Submit** to add the question (including alternative phrasing) and answer to the knowledge base.

In some cases, it makes sense to enable the user to follow up on answer to create a *multi-turn* conversation that enables the user to iteratively refine the question to get to the answer they need.

6. Under the answer you entered for the certification question, select **+ Add follow-up prompts**.
7. In the **Follow-up Prompt** dialog box, enter the following settings, and then click **Add prompt**:

- **Text displayed in the prompt to the user:** `Learn more about certification`.
- Select **Create link to new pair**, and enter this text:

You can learn more about certification on the [Microsoft certification page] (<https://docs.microsoft.com/learn/certifications/>).

- **Show in contextual flow only:** Selected. *This option ensures that the answer is only ever returned in the context of a follow-up question from the original certification question.*

## Train and test the knowledge base

Now that you have a knowledge base, you can test it in Language Studio.

1. At the top right of the page, click **Save changes**.
2. After the changes have been saved, click **Test** to open the test pane.
3. In the test pane, at the top, *deselect* the option to display short answers. Then at the bottom enter the message `Hello`. A suitable response should be returned.
4. In the test pane, at the bottom enter the message `What is Microsoft Learn?`. An appropriate response from the FAQ should be returned.
5. Enter the message `Thanks!`. An appropriate chit-chat response should be returned.
6. Enter the message `Tell me about certification`. The answer you created should be returned along with a follow-up prompt link.
7. Select the **Learn more about certification** follow-up link. The follow-up answer with a link to the certification page should be returned.
8. When you're done testing the knowledge base, close the test pane.

## Deploy and test the knowledge base

The knowledge base provides a back-end service that client applications can use to answer questions. Now you are ready to publish your knowledge base and access its REST interface from a client.

1. In the **LearnFAQ** project in Language Studio, select the **Deploy knowledge base** page.
2. At the top of the page, click **Deploy**. Then click **Deploy** to confirm you want to deploy the knowledge base.
3. When deployment is complete, click **Get prediction URL** to view the REST endpoint for your knowledge base, and copy it to the clipboard (but don't close the dialog box yet).
4. In Visual Studio Code, in the **12-qna** folder, open **ask-question.cmd**. This script uses *Curl* to call the REST interface of a question answering endpoint.
5. In the script, replace `YOUR_PREDICTION_ENDPOINT` with the prediction endpoint you copied (ensuring it is enclosed in the quotation marks).
6. Return to the browser and in the **Get prediction URL** dialog box, note that the sample request includes a value for the **Ocp-Apim-Subscription-Key** parameter, which looks similar to `ab12c345de678fg9hijk01lmno2pqrs34`. This is the authorization key for your resource. Copy it to the clipboard, and then click **Got it** to close the dialog box.
7. Return to Visual Studio Code, and in the **ask-question.cmd** script, replace `YOUR_KEY` with the key you copied (ensuring it is enclosed in the quotation marks).
8. Note that the Curl command in the script submits a **question** parameter with the value **What is a Learning Path?**.
9. Verify that the entire script looks similar to the following code, then save the file.

```
curl -X POST -H "Ocp-Apim-Subscription-Key: 'ab12c345de678fg9hijk01lmno2pqrs34'" -d '{"question": "What is a Learning Path?"}' https://<prediction endpoint>
```

Code Copy

```
@echo off
SETLOCAL ENABLEDELAYEDEXPANSION

rem Set variables
set prediction_url="https://some-name.cognitiveservices.azure.com/language/....."
set key="ab12c345de678fg9hijk01lmno2pqrs34"

curl -X POST !prediction_url! -H "Ocp-Apim-Subscription-Key: !key!" -H "Content-Type: application/json" -d '{"question': 'What is a learning Path?' }"
```

10. In Visual Studio Code in the Explorer pane, right-click the **ask-question.cmd** script and select **Open in Integrated Terminal**.
11. In the terminal pane, enter the command `ask-question.cmd` to run the script and view the JSON response that is returned by the service, which should contain an appropriate answer to the question *What is a learning path?*.

## Create a bot for the knowledge base

Most commonly, the client applications used to retrieve answers from a knowledge base are bots.

1. Return to Language Studio in the browser, and in the **Deploy knowledge base** page, select **Create Bot**. This opens the Azure portal in a new browser tab so you can create a Web App Bot in your Azure subscription (if prompted, sign in).
2. In the Azure portal, create a Web App Bot with the following settings (most of these will be pre-populated for you):

*If some values are missing, refresh your browser.*

- **Bot handle:** A unique name for your bot
- **Subscription:** Your Azure subscription
- **Resource group:** The resource group containing your Language resource
- **Location:** The same location as your Text Analytics service.
- **Pricing tier:** F0
- **App name:** Same as the **Bot handle** with a unique ID and \*.azurewebsites.net appended automatically
- **SDK language:** Choose either C# or Node.js
- **QnA Auth Key:** This should automatically be set to the authentication key for your QnA knowledge base
- **App service plan/location:** This may be set automatically to a suitable plan and location if one exists. If not, create a new plan
- **Application Insights:** Off
- **Microsoft App ID and password:** Auto create App ID and password.

1. Wait for your bot to be created . Then click **Go to resource** (or alternatively, on the home page, click **Resource groups**, open the resource group where you created the web app bot, and click it.)
2. In the blade for your bot, view the **Test in Web Chat** page, and wait until the bot displays the message **Hello and welcome!** (it may take a few seconds to initialize).
3. Use the test chat interface to ensure your bot answers questions from your knowledge base as expected. For example, try submitting `What is Microsoft certification?` .

## More information

To learn more about question answering in the Language service, see the [Langage service documentation](#).

