3.2 - Create question answering solutions with Azure AI Language

---

Overview

The question answering capability of the Azure AI Language service makes it easy to build applications in which users ask questions using natural language and receive appropriate answers.

Learning objectives

After completing this module, you will be able to:

- Understand question answering and how it compares to language understanding.
- Create, test, publish, and consume a knowledge base.
- Implement multi-turn conversation and active learning.
- Create a question answering bot to interact with using natural language.

Introduction

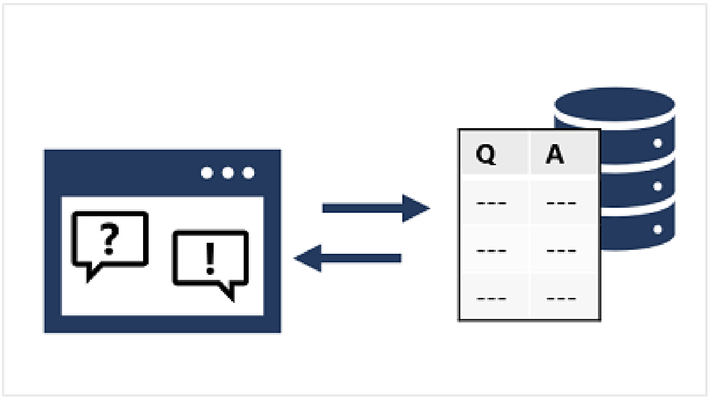
A common pattern for "intelligent" applications is to enable users to ask questions using natural language, and receive appropriate answers. In effect, this kind of solution brings conversational intelligence to a traditional frequently asked questions (FAQ) publication. In this module, you will learn how to use Azure AI Language to create a knowledge base of question and answer pairs that can support an application or bot.

After completing this module, you'll be able to:

- Understand question answering and how it compares to language understanding.
- Create, test, publish and consume a knowledge base.
- Implement multi-turn conversation and active learning.
- Create a question answering bot to interact with using natural language.

Understand question answering

Azure AI Language includes a *question answering* capability, which enables you to define a *knowledge base* of question and answer pairs that can be queried using natural language input. The knowledge base can be published to a REST endpoint and consumed by client applications, commonly *bots*.



The knowledge base can be created from existing sources, including:

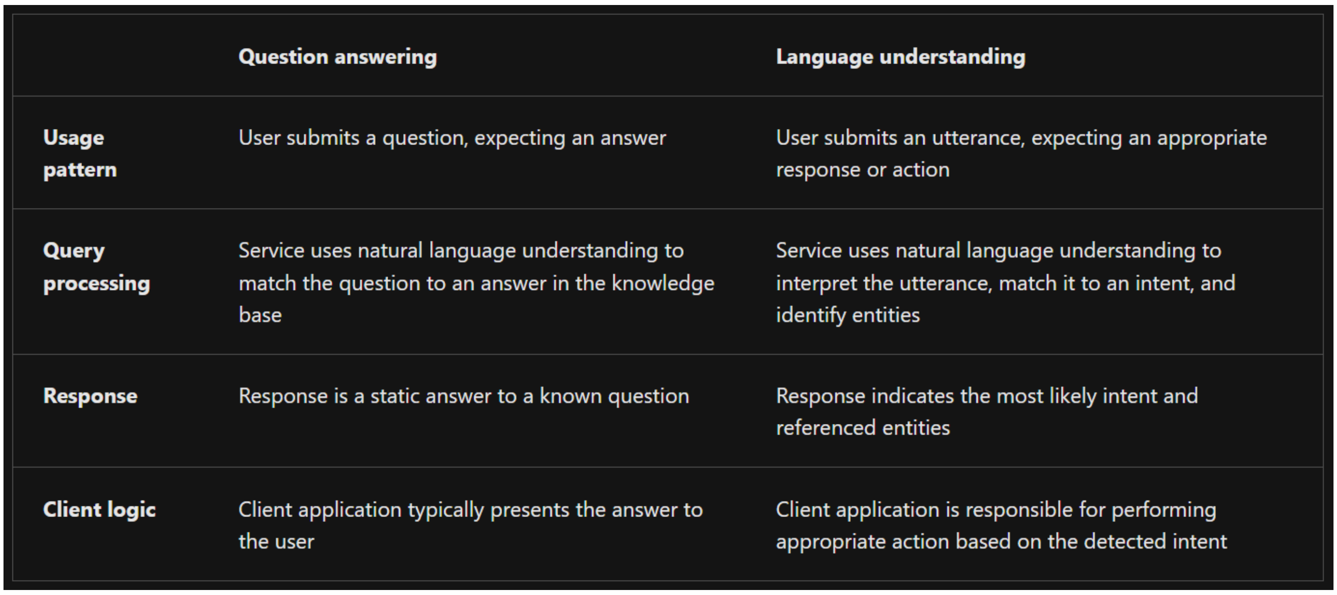
- Web sites containing frequently asked question (FAQ) documentation.
- Files containing structured text, such as brochures or user guides.
- Built-in *chit chat* question and answer pairs that encapsulate common conversational exchanges.

> Note: The question answering capability of Azure AI Language is a newer version of the **QnA Service**, which still exists as a standalone service. To learn how to migrate a QnA Maker knowledge base to Azure AI Language, see the migration guide.

Compare question answering to Azure AI Language understanding

A question answering knowledge base is a form of language model, which raises the question of when to use question answering, and when to use the *conversational language understanding* capabilities of Azure AI Language.

The two features are similar in that they both enable you to define a language model that can be queried using natural language expressions. However, there are some differences in the use cases that they are designed to address, as shown in the following table:



	Question answering	**Language understanding**
Usage pattern	User submits a question, expecting an answer	User submits an utterance, expecting an appropriate response or action
Query processing	Service uses natural language understanding to match the question to an answer in the knowledge base	Service uses natural language understanding to interpret the utterance, match it to an intent, and identify entities
Response	Response is a static answer to a known question	Response indicates the most likely intent and referenced entities
Client logic	Client application typically presents the answer to the user	Client application is responsible for performing appropriate action based on the detected intent

The two services are in fact complementary. You can build comprehensive natural language solutions that combine language understanding models and question answering knowledge bases.

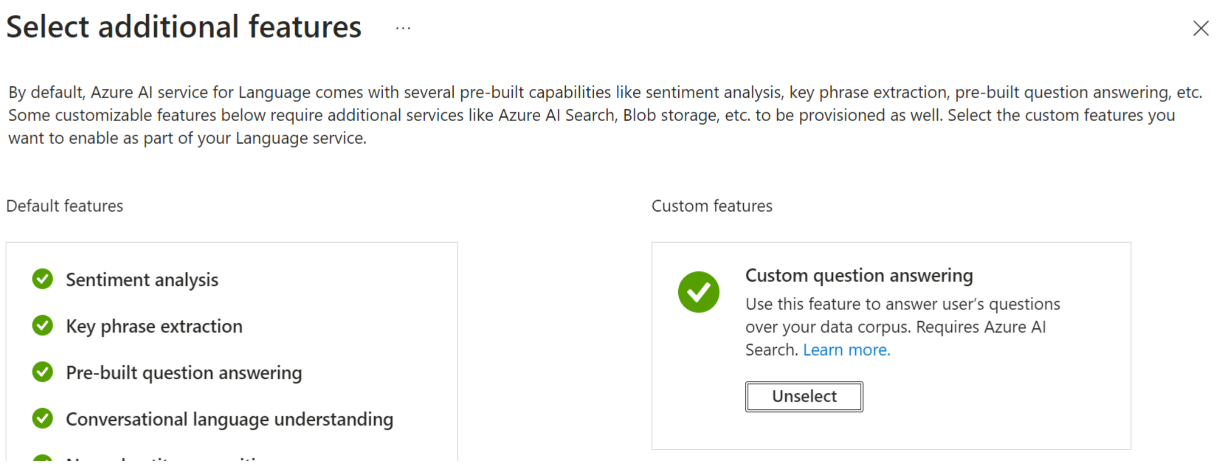
Create a knowledge base

To create a **question answering solution**, you can use the REST API or SDK to write code that defines, trains, and publishes the **knowledge base**. However, it's more common to use the Language Studio web interface to define and manage a knowledge base.

To create a knowledge base you:

1. Sign in to Azure portal.
2. Search for **Azure AI services** using the search field at the top of the portal.
3. Select **Create** under the **Language Service** resource.
4. Create a resource in your Azure subscription:

- Enable the *question answering* feature.



Select additional features ···

By default, Azure AI service for Language comes with several pre-built capabilities like sentiment analysis, key phrase extraction, pre-built question answering, etc. Some customizable features below require additional services like Azure AI Search, Blob storage, etc. to be provisioned as well. Select the custom features you want to enable as part of your Language service.

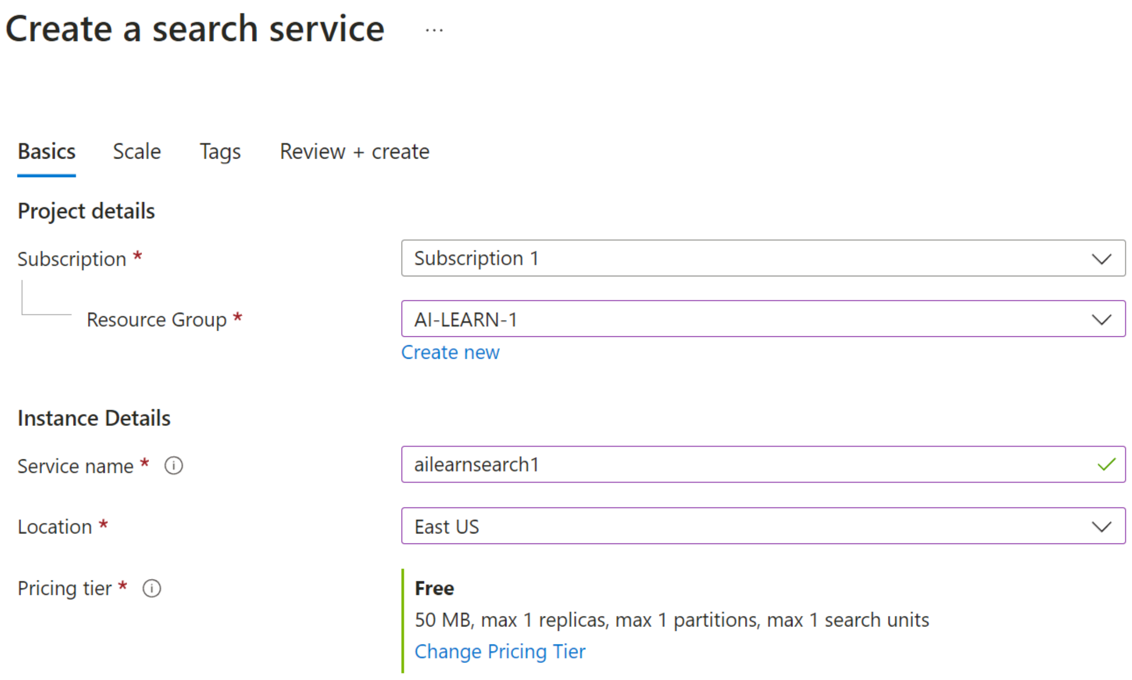
Default features

- ✓ Sentiment analysis
- ✓ Key phrase extraction
- ✓ Pre-built question answering
- ✓ Conversational language understanding

Custom features

✓ Custom question answering
Use this feature to answer user's questions over your data corpus. Requires Azure AI Search. Learn more.

- Create or select an **Azure AI Search** resource to **host the knowledge base index**.



Create a search service ···

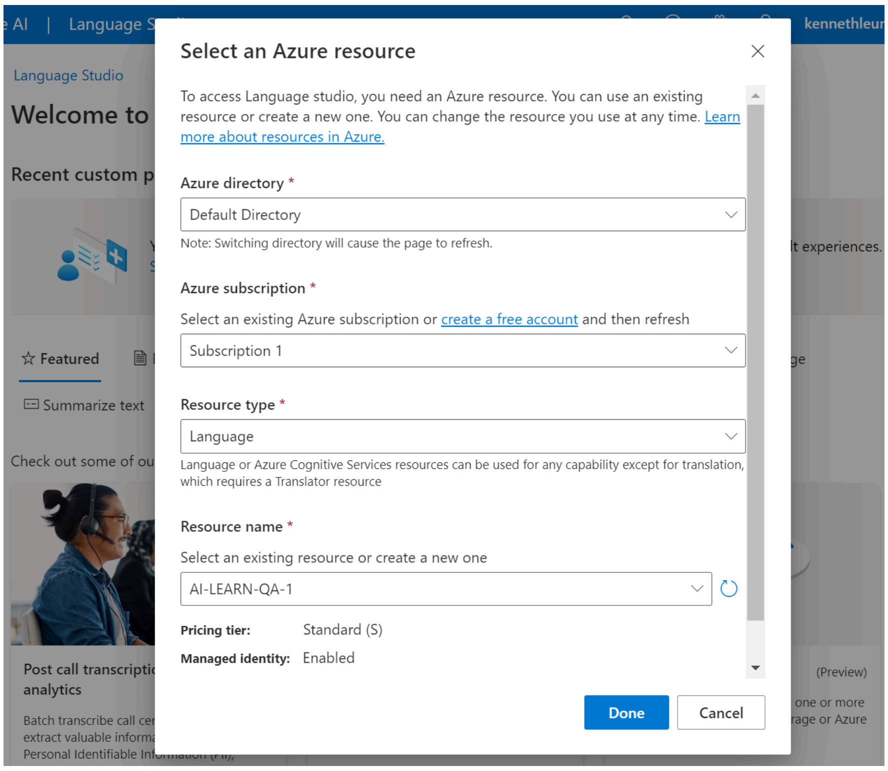
Project details

Subscription *	Subscription 1
Resource Group *	AI-LEARN-1

Instance Details

Service name *	ailearnsearch1
Location *	East US
Pricing tier *	**Free** 50 MB, max 1 replicas, max 1 partitions, max 1 search units

5. In Language Studio, select your Azure AI Language resource and create a **Custom question answering** project.


Select an Azure resource

To access Language studio, you need an Azure resource. You can use an existing resource or create a new one. You can change the resource you use at any time. Learn more about resources in Azure.

Azure directory *

Default Directory

Note: Switching directory will cause the page to refresh.

Azure subscription *

Select an existing Azure subscription or create a free account and then refresh

Subscription 1

Resource type *

Language

Language or Azure Cognitive Services resources can be used for any capability except for translation, which requires a Translator resource

Resource name *

Select an existing resource or create a new one

AI-LEARN-QA-1

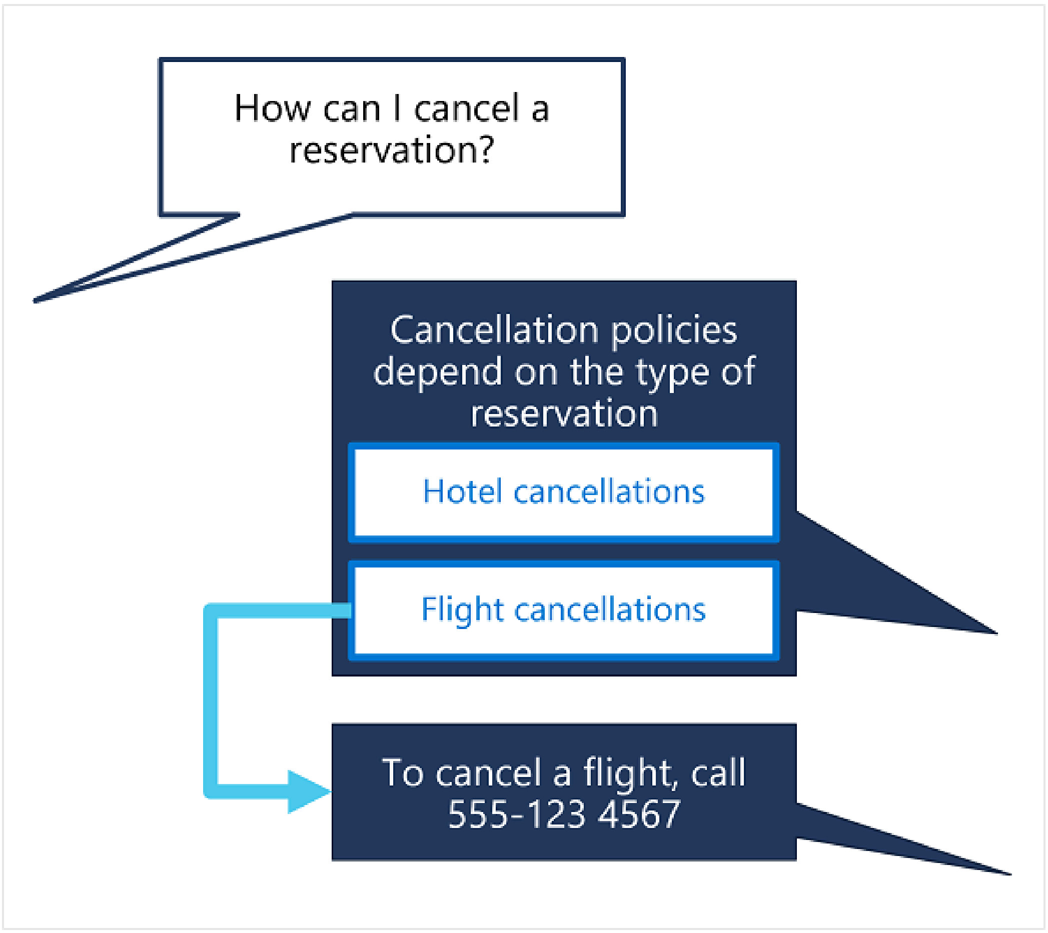
Pricing tier: Standard (S)

Managed identity: Enabled

6. Add one or more data sources to populate the knowledge base:
   - URLs for web pages containing FAQs.
   - Files containing structured text from which questions and answers can be derived.
   - Predefined *chit-chat* datasets that include common conversational questions and responses in a specified style.
7. Edit question and answer pairs in the portal.

Implement multi-turn conversation

Although you can often create an effective knowledge base that consists of individual question and answer pairs, sometimes you might need to ask follow-up questions to elicit more information from a user before presenting a definitive answer. This kind of interaction is referred to as a *multi-turn* conversation.


How can I cancel a reservation?
Cancellation policies depend on the type of reservation
Hotel cancellations
Flight cancellations
To cancel a flight, call 555-123 4567

You can enable multi-turn responses when importing questions and answers from an existing web page or document based on its structure, or you can explicitly define follow-up prompts and responses for existing question and answer pairs.

For example, suppose an initial question for a travel booking knowledge base is "How can I cancel a reservation?". A reservation might refer to a hotel or a flight, so a follow-up prompt is required to clarify this detail. The answer might consist of text such as "Cancellation policies depend on the type of reservation" and include follow-up prompts with links to answers about canceling flights and canceling hotels.

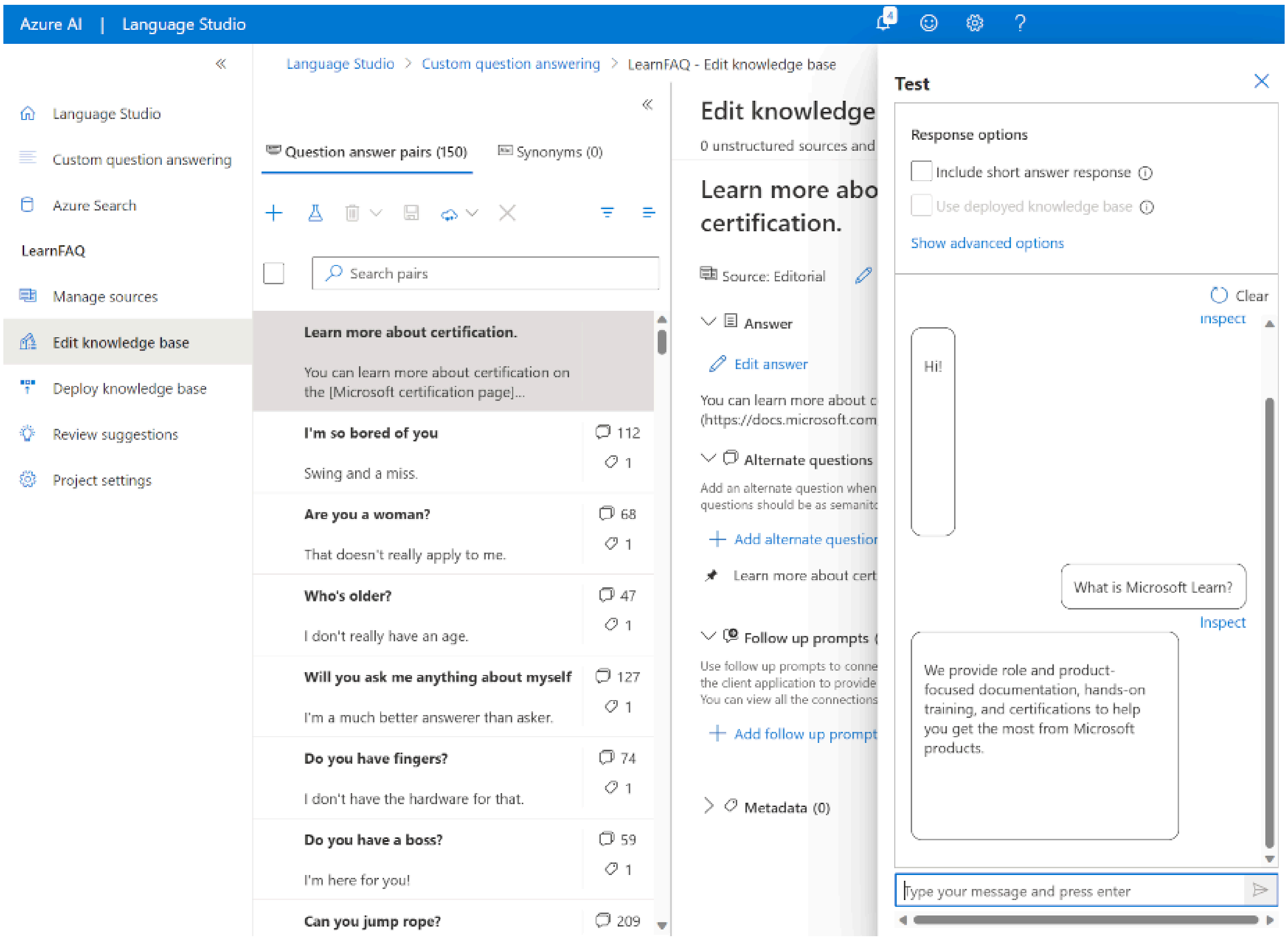
When you define a follow-up prompt for multi-turn conversation, you can link to an existing answer in the knowledge base or define a new answer specifically for the follow-up. You can also restrict the linked answer so that it is only ever displayed in the context of the multi-turn conversation initiated by the original question.

Test and publish a knowledge base

After you have defined a knowledge base, you can train its natural language model, and test it before publishing it for use in an application or bot.

Testing a knowledge base

You can test your knowledge base interactively in Language Studio, submitting questions and reviewing the answers that are returned. You can inspect the results to view their confidence scores as well as other potential answers.


Azure AI | Language Studio

Language Studio
Custom question answering
Azure Search
LearnFAQ
Manage sources
Edit knowledge base
Deploy knowledge base
Review suggestions
Project settings

Language Studio > Custom question answering > LearnFAQ - Edit knowledge base

Question answer pairs (150) Synonyms (0)

Search pairs

Learn more about certification.
You can learn more about certification on the [Microsoft certification page]...

I'm so bored of you 112 1
Swing and a miss.

Are you a woman? 68 1
That doesn't really apply to me.

Who's older? 47 1
I don't really have an age.

Will you ask me anything about myself 127 1
I'm a much better answerer than asker.

Do you have fingers? 74 1
I don't have the hardware for that.

Do you have a boss? 59 1
I'm here for you!

Can you jump rope? 209

Edit knowledge
0 unstructured sources and

Learn more abo
certification.

Source: Editorial

Answer
Edit answer
You can learn more about c
(https://docs.microsoft.com

Alternate questions
Add an alternate question when
questions should be as semanit
Add alternate questio
Learn more about cert

Follow up prompts (
Use follow up prompts to conne
the client application to provide
You can view all the connections
Add follow up prompt

Metadata (0)

Test

Response options
Include short answer response
Use deployed knowledge base
Show advanced options

Clear
Inspect
Hi!
What is Microsoft Learn?
Inspect
We provide role and product-focused documentation, hands-on training, and certifications to help you get the most from Microsoft products.

Type your message and press enter

Deploying a knowledge base

When you are happy with the performance of your knowledge base, you can **deploy it to a REST endpoint** that client applications can use to submit questions and receive answers. You **can deploy it directly from Language Studio.**

Use a knowledge base

To consume the published knowledge base, you can use the REST interface.

The minimal request body for the function contains a question, like this:

```
{
  "question": "What do I need to do to cancel a reservation?",
  "top": 2,
  "scoreThreshold": 20,
  "strictFilters": [
    {
      "name": "category",
      "value": "api"
    }
  ]
}
```

Property	Description
question	Question to send to the knowledge base.
top	Maximum number of answers to be returned.
scoreThreshold	Score threshold for answers returned.
strictFilters	Limit to only answers that contain the specified metadata.

The response includes the closest question match that was found in the knowledge base, along with the associated answer, the confidence score, and other metadata about the question and answer pair:

```
{
  "answers": [
    {
      "score": 27.74823341616769,
      "id": 20,
      "answer": "Call us on 555 123 4567 to cancel a reservation.",
      "questions": [
        "How can I cancel a reservation?"
      ],
      "metadata": [
        {
          "name": "category",
          "value": "api"
        }
      ]
    }
  ]
}
```

Improve question answering performance

After creating and testing a knowledge base, you can improve its performance with *active learning* and by defining *synonyms*.

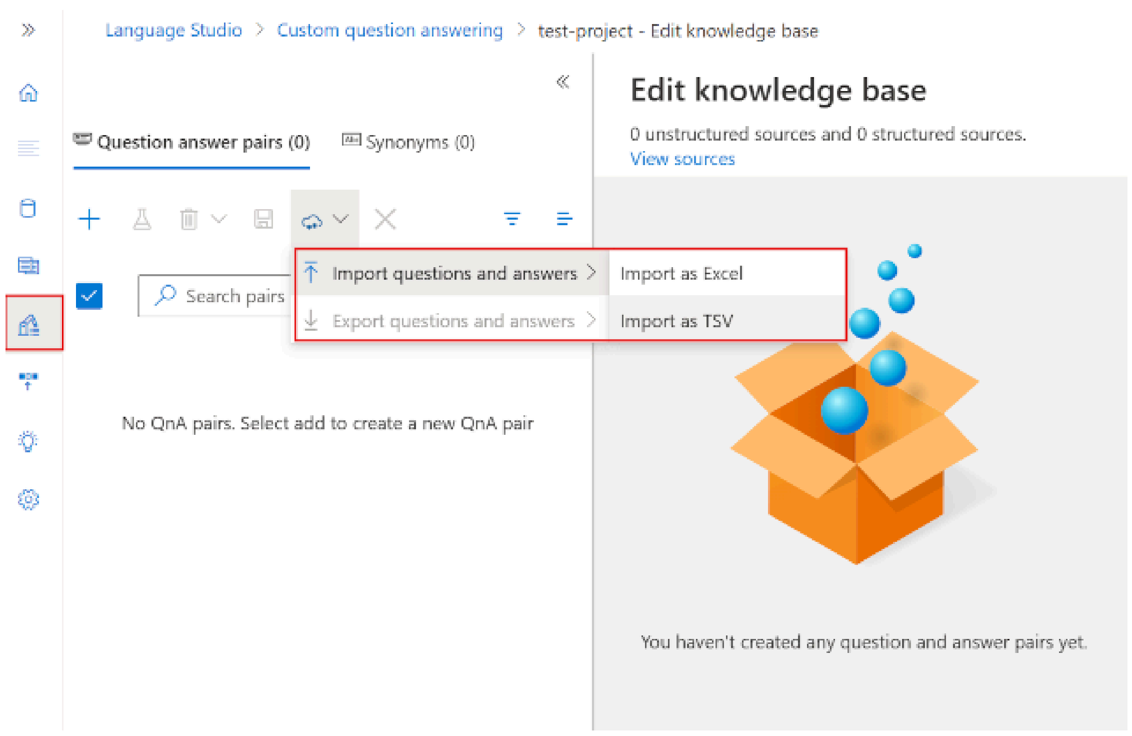
Use active learning

Active learning can help you make continuous improvements to get better at answering user questions correctly over time. People often ask questions that are phrased differently, but ultimately have the same meaning. Active learning can help in situations like this because it enables you to **consider alternate questions to each question and answer pair.** Active learning is enabled by default.

To use active learning, you can do the following:

Create your question and answer pairs

You create pairs of questions and answers in Language Studio for your project. You can also import a file that contains question and answer pairs to upload in bulk.


Language Studio > Custom question answering > test-project - Edit knowledge base
Question answer pairs (0)
Synonyms (0)
Search pairs
Import questions and answers
Export questions and answers
Import as Excel
Import as TSV
View sources

Edit knowledge base

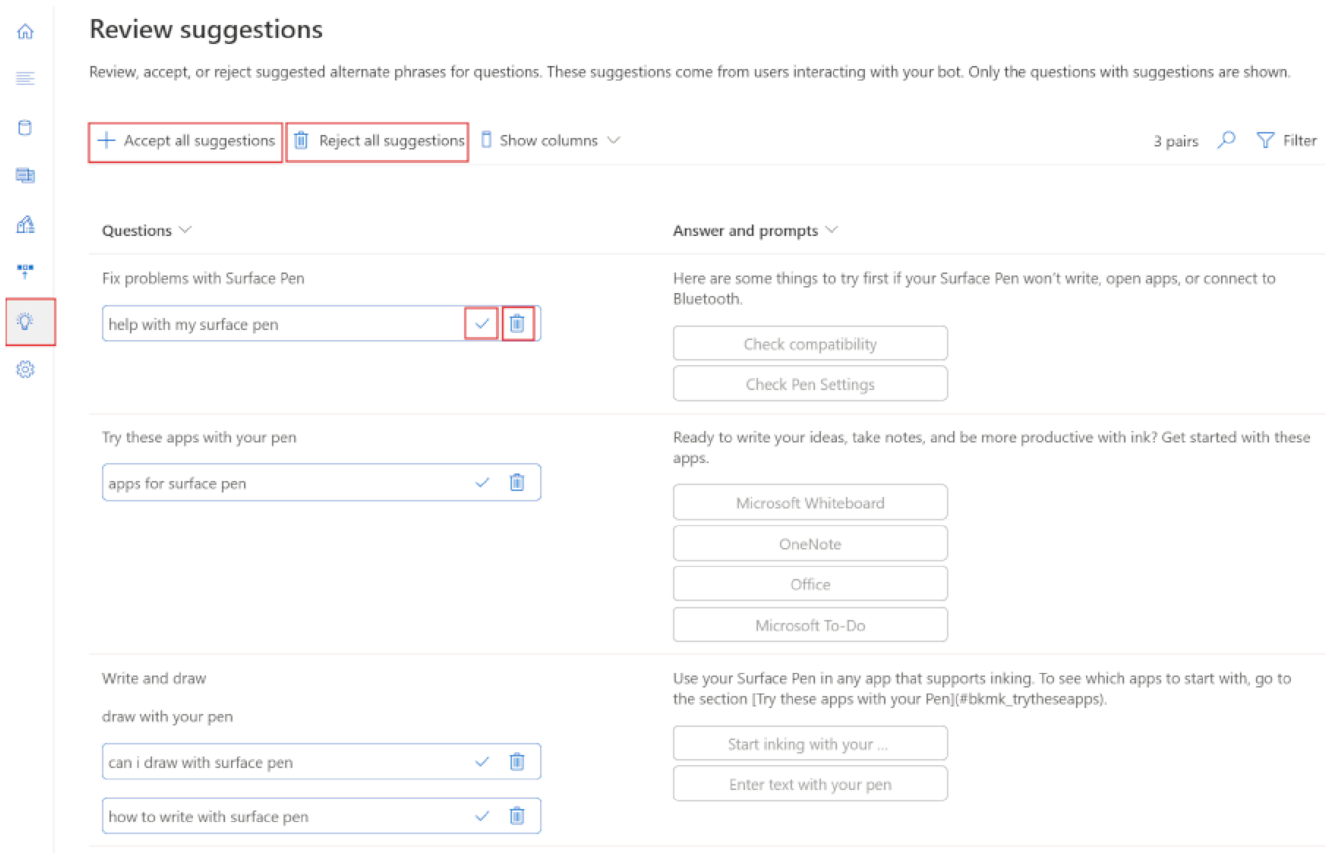
0 unstructured sources and 0 structured sources.

No QnA pairs. Select add to create a new QnA pair

You haven't created any question and answer pairs yet.

Review suggestions

Active learning then begins to offer alternate questions for each question in your question and answer pairs. You access this from the Review suggestions pane:



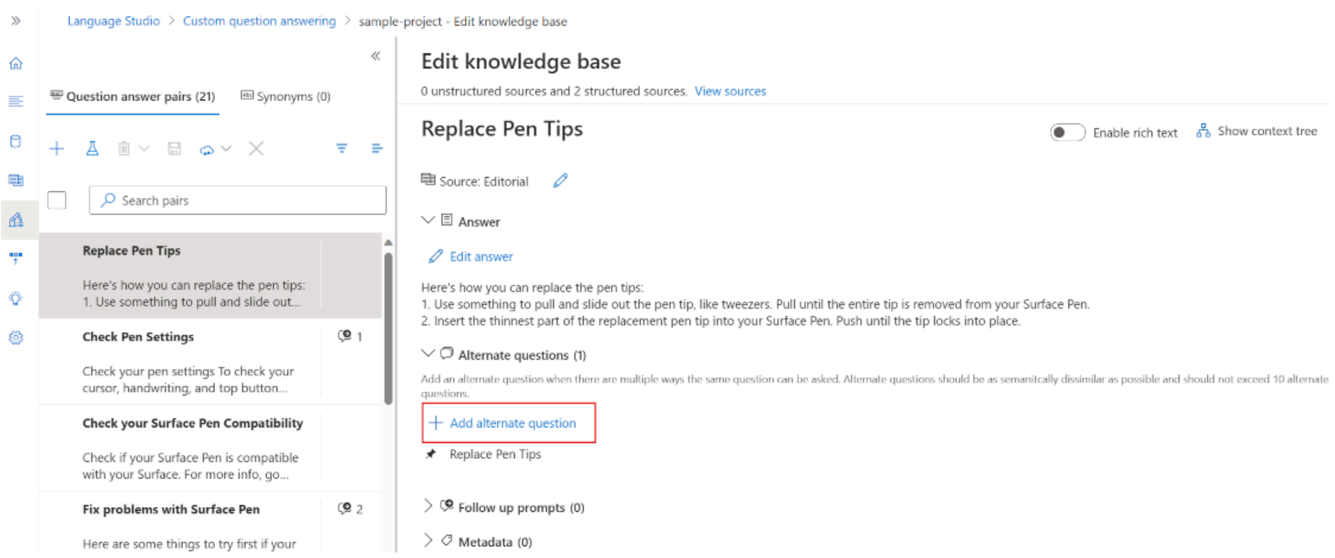
Review suggestions

Review, accept, or reject suggested alternate phrases for questions. These suggestions come from users interacting with your bot. Only the questions with suggestions are shown.

Questions	Answer and prompts
Fix problems with Surface Pen help with my surface pen	Here are some things to try first if your Surface Pen won't write, open apps, or connect to Bluetooth. Check compatibility Check Pen Settings
Try these apps with your pen apps for surface pen	Ready to write your ideas, take notes, and be more productive with ink? Get started with these apps. Microsoft Whiteboard OneNote Office Microsoft To-Do
Write and draw draw with your pen can i draw with surface pen how to write with surface pen	Use your Surface Pen in any app that supports inking. To see which apps to start with, go to the section [Try these apps with your Pen](#bkmk_trytheseapps). Start inking with your ... Enter text with your pen

You review, and then accept or reject these alternate phrases suggested for each question by selecting the checkmark or delete symbol next to the alternate phrase. You can bulk accept or reject suggestions using the **Accept all suggestions** or **Reject all suggestions** option at the top.

You can also **manually add alternate questions** when you select **Add alternate question** for a pair in the Edit knowledge base pane:


Language Studio > Custom question answering > sample-project - Edit knowledge base

Question answer pairs (21) | Synonyms (0)

Search pairs

Replace Pen Tips
Here's how you can replace the pen tips:
1. Use something to pull and slide out...

Check Pen Settings 1
Check your pen settings To check your cursor, handwriting, and top button...

Check your Surface Pen Compatibility
Check if your Surface Pen is compatible with your Surface. For more info, go...

Fix problems with Surface Pen 2
Here are some things to try first if your

Edit knowledge base
0 unstructured sources and 2 structured sources. View sources

Replace Pen Tips
Enable rich text | Show context tree

Source: Editorial

Answer
Edit answer

Here's how you can replace the pen tips:
1. Use something to pull and slide out the pen tip, like tweezers. Pull until the entire tip is removed from your Surface Pen.
2. Insert the thinnest part of the replacement pen tip into your Surface Pen. Push until the tip locks into place.

Alternate questions (1)
Add an alternate question when there are multiple ways the same question can be asked. Alternate questions should be as semanitcally dissimilar as possible and should not exceed 10 alternate questions.

+ Add alternate question

Replace Pen Tips

Follow up prompts (0)

Metadata (0)

> Note: To learn more about active learning, see Enrich your project with active learning.

Define synonyms

Synonyms are useful when questions submitted by users might include multiple different words to mean the same thing. For example, a travel agency customer might refer to a "reservation" or a "booking". **By defining these as synonyms, the question answering service can find an appropriate answer regardless of which term an individual customer uses.**

To define synonyms, you use the REST API to submit synonyms in the following JSON format:

```
{
    "synonyms": [
        {
            "alterations": [
                "reservation",
                "booking"
                ]
        }
    ]
}
```

> Note: To learn more about synonyms, see the Improve quality of response with synonyms.

Exercise - 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.

Azure AI Language 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.

Provision an *Azure AI Language* resource

If you don't already have one in your subscription, you'll need to provision an **Azure AI Language service** resource. Additionally, to create and host a knowledge base for question answering, you need to enable the **Question Answering** feature.

1. Open the Azure portal at `https://portal.azure.com`, and sign in using the Microsoft account associated with your Azure subscription.
2. In the search field at the top enter **Azure AI services**, then press **Enter**.
3. Select **Create** under the **Language Service** resource in the results.
4. **Select** the **Custom question answering** block. Then select **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*.
   - **Region**: *Choose any available location*
   - **Name**: *Enter a unique name*
   - **Pricing tier**: Select **F0** (*free*), or **S** (*standard*) if F is not available.
   - **Azure Search region**: *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)*)
   - **Responsible AI Notice**: *Agree*
5. Select **Create + review**, then select **Create**.

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

6. Wait for deployment to complete, and then go to the deployed resource.
7. View the **Keys and Endpoint** page. You will need the information on this page later in the exercise.

Create a question answering project

To create a knowledge base for question answering in your Azure AI 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.cognitive.azure.com/ and sign in using the Microsoft account associated with your Azure subscription.
2. If you're prompted to choose a Language resource, select the following settings:

   - **Azure Directory**: The Azure directory containing your subscription.
   - **Azure subscription**: Your Azure subscription.
   - **Resource type**: Language
   - **Resource name**: The Azure AI Language resource you created previously.

   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:
   1. On the bar at the top if the page, select the **Settings (⚙)** button.
   2. On the **Settings** page, view the **Resources** tab.
   3. Select the language resource you just created, and click **Switch resource**.
   4. At the top of the page, click **Language Studio** to return to the Language Studio home page.

3. At the top of the portal, in the **Create new** menu, select **Custom question answering**.
4. 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 select **Next**.
5. On the **Enter basic information** page, enter the following details:
    - **Name** `LearnFAQ`
    - **Description**: `FAQ for Microsoft Learn`
    - **Default answer when no answer is returned**: `Sorry, I don't understand the question`
6. Select **Next**.
7. On the **Review and finish** page, select **Create project**.

Add 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, select **+Add url** and set the following name and URL before you select **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 select **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 your **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 choose **Got it** to dismiss them, or select **Skip all**)
2. In the knowledge base, on the **Question answer pairs** tab, select **+**, and create a new question answer pair with the following settings:
    - **Source**: `https://docs.microsoft.com/en-us/learn/support/faq`
    - **Question**: `What are Microsoft credentials?`
    - **Answer**: `Microsoft credentials enable you to validate and prove your skills with Microsoft technologies.`
3. Select **Done**.
4. In the page for the **What are Microsoft credentials?** question that is created, expand **Alternate questions**. Then add the alternate question `How can I demonstrate my Microsoft technology skills?`.
   In some cases, it makes sense to enable the user to follow up on an answer by creating a *multi-turn* conversation that enables the user to iteratively refine the question to get to the answer they need.
5. Under the answer you entered for the certification question, expand **Follow-up prompts** and add the following follow-up prompt:
    - **Text displayed in the prompt to the user**: `Learn more about credentials`.

    - Select the **Create link to new pair** tab, and enter this text: `You can learn more about credentials on the [Microsoft credentials page](https://docs.microsoft.com/learn/credentials/).`
    - Select **Show in contextual flow only**. This option ensures that the answer is only ever returned in the context of a follow-up question from the original certification question.
6. Select **Add prompt**.

Train and test the knowledge base

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

1. Save the changes to your knowledge base by selecting the **Save** button under the **Question answer pairs** tab on the left.
2. After the changes have been saved, select the **Test** button to open the test pane.
3. In the test pane, at the top, deselect **Include short answer response** (if not already unselected). 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 Microsoft credentials`. The answer you created should be returned along with a follow-up prompt link.
7. Select the **Learn more about credentials** 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 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, select **Deploy**. Then select **Deploy** to confirm you want to deploy the knowledge base.
3. When deployment is complete, select **Get prediction URL** to view the REST endpoint for your knowledge base and note that the sample request includes parameters for:
    - **projectName**: The name of your project (which should be *LearnFAQ*)
    - **deploymentName**: The name of your deployment (which should be *production*)
4. Close the prediction URL dialog box.

Prepare to develop an app in Visual Studio Code

You'll develop your question answering app using Visual Studio Code. The code files for your app have been provided in a GitHub repo.

> **Tip**: If you have already cloned the **mslearn-ai-language** repo, open it in Visual Studio code. Otherwise, follow these steps to clone it to your development environment.

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/mslearn-ai-language` 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.

   > **Note**: If Visual Studio Code shows you a pop-up message to prompt you to trust the code you are opening, click on **Yes, I trust the authors** option in the pop-up.

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**.

Configure your application

Applications for both C# and Python have been provided, as well as a sample text file you'll use to test the summarization. Both apps feature the same functionality. First, you'll complete some key parts of the application to enable it to use your Azure AI Language resource.

1. In Visual Studio Code, in the **Explorer** pane, browse to the **Labfiles/02-qna** folder and expand the **CSharp** or **Python** folder depending on your language preference and the **qna-app** folder it contains. Each folder contains the language-specific files for an app into which you're you're going to integrate Azure AI Language question answering functionality.
2. Right-click the **qna-app** folder containing your code files and open an integrated terminal. Then install the Azure AI Language question answering SDK package by running the appropriate command for your language preference:

   C#:

   ```
   dotnet add package Azure.AI.Language.QuestionAnswering
   ```

   Python:

   ```
   pip install azure-ai-language-questionanswering
   ```

3. In the **Explorer** pane, in the **qna-app** folder, open the configuration file for your preferred language
   - **C#**: appsettings.json
   - **Python**: .env
4. Update the configuration values to include the **endpoint** and a **key** from the Azure Language resource you created (available on the **Keys and Endpoint** page for your Azure AI Language resource in the Azure portal). The project name and deployment name for your deployed knowledge base should also be in this file.
5. Save the configuration file.

Add code to the application

Now you're ready to add the code necessary to import the required SDK libraries, establish an authenticated connection to your deployed project, and submit questions.

1. Note that the **qna-app** folder contains a code file for the client application:

   - **C#**: Program.cs
   - **Python**: qna-app.py (View here: qna-app.py)

Open the code file and at the top, under the existing namespace references, find the comment **Import namespaces**. Then, under this comment, add the following language-specific code to import the namespaces you will need to use the Text Analytics SDK:

C#: Programs.cs

```
// import namespaces using Azure; using Azure.AI.Language.QuestionAnswering;
```

Python: qna-app.py (View here: [qna-app.py]())

```
# import namespaces from azure.core.credentials import AzureKeyCredential from azure.ai.language.questionanswering import QuestionAnsweringClient
```

2. In the **Main** function, note that code to load the Azure AI Language service endpoint and key from the configuration file has already been provided. Then find the comment **Create client using endpoint and key**, and add the following code to create a client for the Text Analysis API:

   C#: Programs.cs

   ```
   // Create client using endpoint and key AzureKeyCredential credentials = new AzureKeyCredential(aiSvcKey); Uri endpoint = new Uri(aiSvcEndpoint); QuestionAnsweringClient aiClient = new QuestionAnsweringClient(endpoint, credentials);
   ```

   Python: qna-app.py (View here: [qna-app.py]())

   ```
   # Create client using endpoint and key credential = AzureKeyCredential(ai_key) ai_client = QuestionAnsweringClient(endpoint=ai_endpoint, credential=credential)
   ```

3. In the **Main** function, find the comment **Submit a question and display the answer**, and add the following code to repeatedly read questions from the command line, submit them to the service, and display details of the answers:

   C#: Programs.cs

   ```
   // Submit a question and display the answer string user_question = ""; while (user_question.ToLower() != "quit") { Console.Write("Question: "); user_question = Console.ReadLine(); QuestionAnsweringProject project = new QuestionAnsweringProject(projectName, deploymentName); Response<AnswersResult> response = aiClient.GetAnswers(user_question, project); foreach (KnowledgeBaseAnswer answer in response.Value.Answers) { Console.WriteLine(answer.Answer); Console.WriteLine($"Confidence: {answer.Confidence:P2}"); Console.WriteLine($"Source: {answer.Source}"); Console.WriteLine(); } }
   ```

   Python: qna-app.py (View here: [qna-app.py]())

   ```
   # Submit a question and display the answer user_question = '' while user_question.lower() != 'quit': user_question = input('\nQuestion:\n') response = ai_client.get_answers(question=user_question, project_name=ai_project_name, deployment_name=ai_deployment_name) for candidate in response.answers: print(candidate.answer) print("Confidence: {}".format(candidate.confidence)) print("Source: {}".format(candidate.source))
   ```

4. Save your changes and return to the integrated terminal for the **qna-app** folder, and enter the following command to run the program:
   - **C#**: `dotnet run`
   - **Python**: `python qna-app.py`
5. When prompted, enter a question to be submitted to your question answering project; for example `What is a learning path?`.
6. Review the answer that is returned.

7. Ask more questions. When you're done, enter `quit`.

Clean up resources

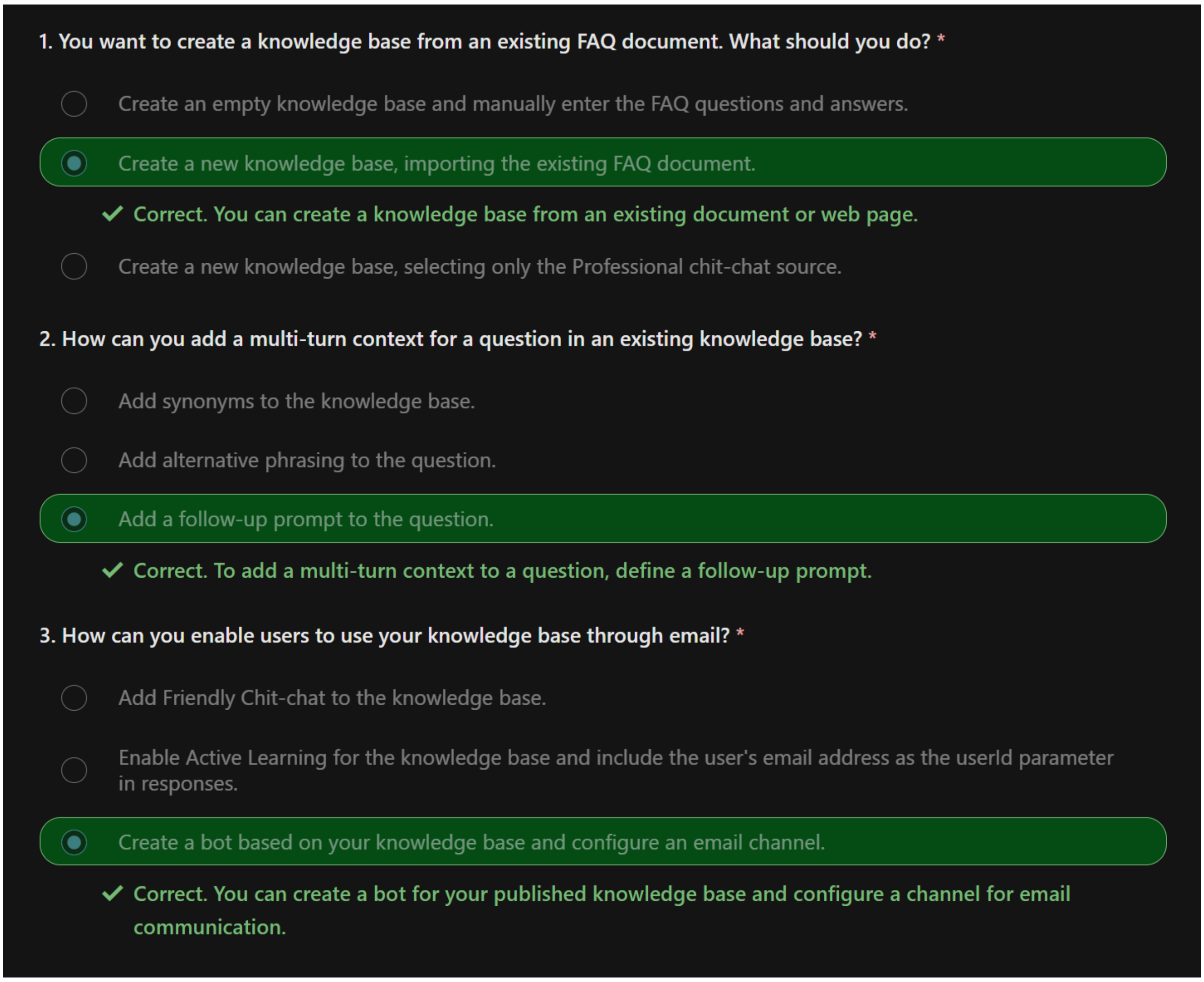
If you're finished exploring the Azure AI Language service, you can delete the resources you created in this exercise. Here's how:

1. Open the Azure portal at `https://portal.azure.com`, and sign in using the Microsoft account associated with your Azure subscription.
2. Browse to the Azure AI Language resource you created in this lab.
3. On the resource page, select **Delete** and follow the instructions to delete the resource.

More information

To learn more about question answering in Azure AI Language, see the Azure AI Language documentation.

Knowledge Check



1. You want to create a knowledge base from an existing FAQ document. What should you do? *

- Create an empty knowledge base and manually enter the FAQ questions and answers.
- [✓ Create a new knowledge base, importing the existing FAQ document.]

  ✓ Correct. You can create a knowledge base from an existing document or web page.
- Create a new knowledge base, selecting only the Professional chit-chat source.

2. How can you add a multi-turn context for a question in an existing knowledge base? *

- Add synonyms to the knowledge base.
- Add alternative phrasing to the question.
- [✓ Add a follow-up prompt to the question.]

  ✓ Correct. To add a multi-turn context to a question, define a follow-up prompt.

3. How can you enable users to use your knowledge base through email? *

- Add Friendly Chit-chat to the knowledge base.
- Enable Active Learning for the knowledge base and include the user's email address as the userId parameter in responses.
- [✓ Create a bot based on your knowledge base and configure an email channel.]

  ✓ Correct. You can create a bot for your published knowledge base and configure a channel for email communication.

Summary

In this module, you have learned how to use the question answering capability of Azure AI Language to create a knowledge base of question and answer pairs that can support an application or bot.

Now that you've completed this module, you can:

- Understand question answering and how it compares to language understanding.
- Create, test, publish and consume a knowledge base.
- Implement multi-turn conversation and active learning.
- Create a question answering bot to interact with using natural language.

To learn more about the question answering capability of Azure AI Language, see the Question answering documentation.

✍️ Compiled by Kenneth Leung (2025)