Explore question answering

For customer support scenarios, it's common to create a bot that can interpret and answer frequently asked questions through a website chat window, email, or voice interface. Underlying the bot interface is a knowledge base of questions and appropriate answers that the bot can search for suitable responses.

Create a custom question answering knowledge base

The Language service's custom question answering feature enables you to quickly create a knowledge base, either by entering question and answer pairs or from an existing document or web page. It can then use some built-in natural language processing capabilities to interpret questions and find appropriate answers.

1. Open the Azure portal at https://portal.azure.com, signing in with your Microsoft account.
2. Click the **+Create a resource** button, search for *Language service*, and create a **Language service** resource with the following settings, and then click **Continue to create your resource**: **Select Additional Features**
    - **Default features**: *Keep the default features*.
    - **Custom features**: *Select custom question answering*.

Select additional features ···   ✕

By default, Azure Cognitive 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 Cognitive 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 (preview)
- ✅ Named entity recognition
- ✅ Text Summarization
- ✅ Text analytics for Health

Custom features

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

[Unselect]

✅ **Custom text classification & extraction (preview)** ⓘ
Use this feature to customize text classification or custom text extraction. Requires Azure Storage. Learn more.

[Select]

[Continue to create your resource]

---

3. On the **Create Language** page, specify the following settings:
   - **Subscription**: *Your Azure subscription.*
   - **Resource group**: *Select an existing resource group or create a new one.*
   - **Name**: *A unique name for your Language resource.*
   - **Pricing tier**: S (1K Calls per minute)
   - **Azure search region**: *Any available location.*
   - **Azure search pricing tier**: Free F (3 Indexes) - (*If this tier is not available, select Standard S (50 Indexes)*)
   - **By checking this box I certify that I have reviewed and acknowledge the terms in the Responsible AI Notice**: *Selected.*

   > **Note** If you have already provisioned a free-tier **Azure Cognitive Search** resources, your quota may not allow you to create another one. In which case, select a tier other than **Free F**.

4. Click **Review and Create** and then click **Create**. Wait for the deployment of the Language service that will support your custom question answering knowledge base.
5. In a new browser tab, open the Language Studio portal at https://language.azure.com and sign in using the Microsoft account associated with your Azure subscription.
6. If prompted to choose a Language resource, select the following settings:

- o **Azure directory**: The Azure directory containing your subscription.
- o **Azure subscription**: Your Azure subscription.
- o **Language resource**: The Language resource you created previously.
7. 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:
    - o On the bar at the top if the page, click the **Settings (⚙)** button.
    - o On the **Settings** page, view the **Resources** tab.
    - o Select the language resource you just created, and click **Switch resource**.
    - o At the top of the page, click **Language Studio** to return to the Language Studio home page.
8. At the top of the Language Studio portal, in the **Create new** menu, select **Custom question answering**.
9. On the **Choose language setting for resource *your resource*** page, select **I want to select the language when I create a project in this resource** and click **Next**.
10. On the **Enter basic information** page, enter the following details and click **Next**:
    - o **Language resource**: *choose your language resource*.
    - o **Azure search resource**: *choose your Azure search resource*.
    - o **Name**: MargiesTravel
    - o **Description**: A simple knowledge base
    - o **Source language**: English
    - o **Default answer when no answer is returned**: No answer found
11. On the **Review and finish** page, click **Create project**.
12. You will be taken to the **Manage sources** page. Click ➕**Add source** and select **URLs**.
13. In the **Add URLs** box, click **+ Add url**. Type in the following and select **Add all**:
    - o **URL name**: MargiesKB
    - o **URL**: https://raw.githubusercontent.com/MicrosoftLearning/AI-900-AIFundamentals/main/data/qna/margies_faq.docx
    - o **Classify file structure**: *Auto-detect*

Edit the knowledge base

Your knowledge base is based on the details in the FAQ document and some pre-defined responses. You can add custom question-and-answer pairs to supplement these.

1. Click **Edit knowledge base** on the left hand panel. Then click **+ Add question pair**.

2. In the **Questions** box, type `Hello`, then click **Submit changes**.
3. Click **+ Add alternate phrase** and type `Hi`, then click **Submit changes**.
4. In the **Answer and prompts** box, type `Hello`. Keep the **Source**: Editorial.
5. Click **Submit**. Then at the top of the page click **Save changes**. You may need to change the size of your window to see the button.

Train and test the knowledge base

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

1. At the top of the page, click **Test** to test your knowledge base.
2. In the test pane, at the bottom enter the message *Hi*. The response **Hello** should be returned.
3. In the test pane, at the bottom enter the message *I want to book a flight*. An appropriate response from the FAQ should be returned.

> **Note** The response includes a *short answer* as well as a more verbose *answer passage* - the answer passage shows the full text in the FAQ document for the closest matched question, while the short answer is intelligently extracted from the passage. You can control whether the short answer is from the response by using the **Display short answer** checkbox at the top of the test pane.

4. Try another question, such as *How can I cancel a reservation?*
5. When you're done testing the knowledge base, click **Test** to close the test pane.

Create a bot for the knowledge base

The knowledge base provides a back-end service that client applications can use to answer questions through some sort of user interface. Commonly, these client applications are bots. To make the knowledge base available to a bot, you must publish it as a service that can be accessed over HTTP. You can then use the Azure Bot Service to create and host a bot that uses the knowledge base to answer user questions.

1. At the left of the Language Studio page, click **Deploy knowledge base**.
2. At the top of the page, click **Deploy**, and the click **Deploy** again.
3. After the service has been deployed, click **Create a bot**. This opens the Azure portal in a new browser tab so you can create a Web App Bot in your Azure subscription.
4. In the Azure portal, create a Web App Bot with the following settings (most of these will be pre-populated for you):

- o **Bot handle**: *A unique name for your bot*
- o **Subscription**: *Your Azure subscription*
- o **Resource group**: *The resource group containing your Language resource*
- o **Location**: *The same location as your Language service.*
- o **Pricing tier**: Free (F0)
- o **App name**: *Same as the **Bot handle** with **.azurewebsites.net** appended automatically*
- o **SDK language**: *Choose either C# or Node.js*
- o **Language Resource Key**: *automatically generated, if you do not see it, you need to start by creating a question answering project in the Language Studio*
- o **App service plan/Location**: *Select the arrow to create a plan. Then create a unique App service plan name and choose a suitable location*
- o **Application Insights**: Off
- o **Microsoft App ID and password**: *Auto create App ID and password*

5. Wait for your bot to be created (the notification icon at the top right, which looks like a bell, will be animated while you wait). Then in the notification that deployment has completed, 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.)
6. In the left-hand pane of your bot look for **Settings**, click on **Test in Web Chat**, and wait until the bot displays the message **Hello and Welcome** (it may take a few seconds to initialize).
7. Use the test chat interface to ensure your bot answers questions from your knowledge base as expected. For example, try submitting *I need to cancel my hotel*.

Experiment with the bot. You'll probably find that it can answer questions from the FAQ quite accurately, but it will have limited ability to interpret questions that it has not been trained with. You can always use the Language Studio to edit the knowledge base to improve it, and republish it.

Learn more

- To learn more about the Question Answering service, view the documentation.
- To learn more about the Microsoft Bot Service, view the Azure Bot Service page.