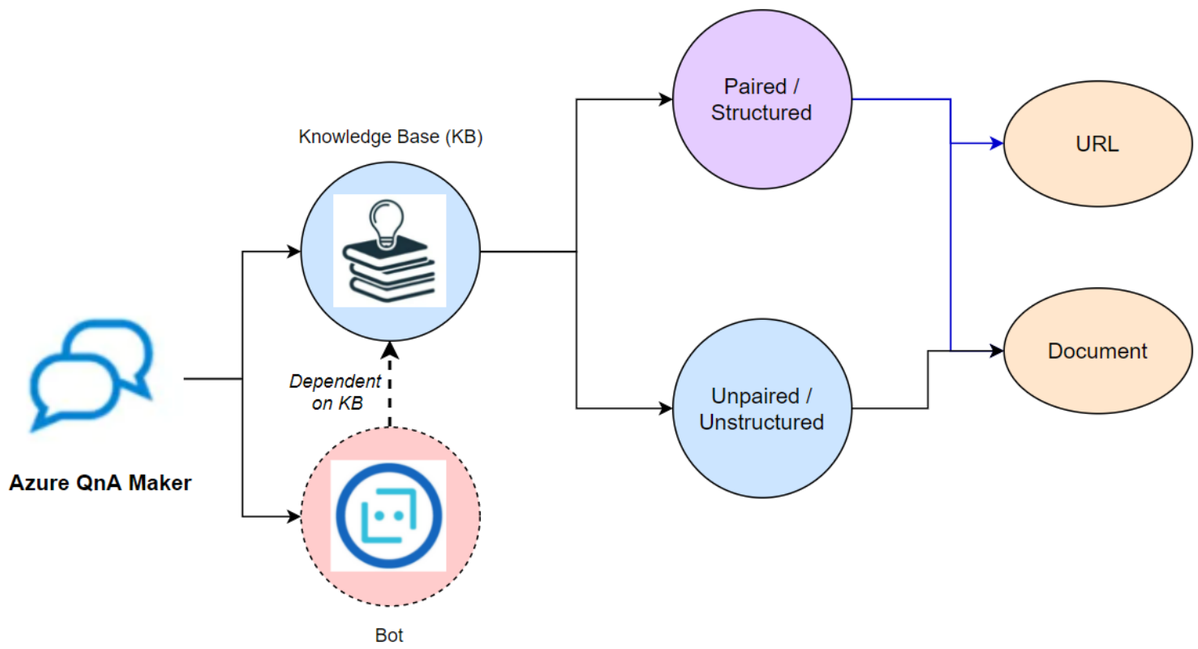
Document Name	HOL – Azure QnA Maker Service v5
Author	Shiva S Tomar & Anupreet Kaur
Reviewer	
Executive Summary	Azure Cognitive APIs enable the developers of all skill levels to add human intelligence in their applications. The services are designed for developers interested in pursuing DS/AI/ML skills and people who want to acquire the deep technical knowledge on the Cognitive APIs of Azure, despite not having Machine Learning expertise.
Purpose	This document is created to help you gain level 350 working knowledge on Azure QnA Maker Service. You will be able to explore each functionality offered by the service through the GUI Portal to train the model & REST APIs to observe the outcomes. We have also shared a sample dataset to replicate what we have used to create the content of this workshop. Once you complete these labs, you'll go from ***Zero to Hero*** on the respective Azure Cognitive service and should be able to ***D****emo, **D**evelop and **D**eploy* your own custom use cases. The important thing to note here is that you don't need to refer any other documents to complete this workshop.
Intent of Guide	This workshop is designed to help you explore all the features of a service offered through their GUI Portal & APIs. The diagram shown in the beginning of the document is its functional Architecture; talking about the functionalities offered by the service in a flow. It also covers the Concepts, How-to and best practices about the service. This document is not intended to enable you with scenarios of deployment in production.

Service brief: Azure QnA Maker Service

Azure's QnA Maker QnA Maker is a cloud-based Natural Language Processing (NLP) service that allows you to create a natural conversational layer over your data. It is used to find the most appropriate answer for any input from your custom Knowledge Base (KB) of information. This is commonly used to build conversational client applications, which include custom virtual assistants, social media applications, chat bots, and speech-enabled desktop applications.

Diagram: Functional Architecture



- Knowledge Base is like a collection of question answer pairs related to a subject area, for example – retail, healthcare etc. These can be extracted from URLs or Documents like Word, PDF, Excel etc. There are 2 ways in which we can feed the data from URLs & Documents into Knowledge Base :
  - Structured aka paired – Where the data has a question and a corresponding answer clearly mapped to it. For example, question in one cell of the excel and answer in corresponding cell. The content on the URL pages or documents can also be semi-structured, meaning that even if there isn't a 100% structural correlation between the question answer pairs, the model is able to recognise the corresponding pairs intelligently, using the content placement, fonts, boldness etc.
  - Unstructured aka unpaired - This includes continuous paragraphs as part of unstructured text and the model intelligently picks the relevant portion while answering the questions. For example, free flowing text in Documents. Here, we can only upload Documents & files, URLs are not yet supported to include completely unstructured text.

- Once you have trained & Published a Knowledge Base, you can leverage it in 2 major ways :
  - Through REST API or SDKs – Embed the generated endpoints and keys in your applications, to provide end user with intelligent conversational experience.
  - Publish a Bot – A bot is like an application to provide conversational AI experiences for your customers. You can publish a bot to different channels such as Teams, Cortana, Email, Direct line etc. In the background, the bot will be leveraging the QnA maker capabilities of the KB that we built. You could also pick up the packaged code and further build on it to add other cognitive services such as Computer Vision, Text Analytics, Speech etc.
    Hop on to the workshops of these cognitive services to learn more about them!

These APIs are available both as REST APIs and language specific SDKs.

However, in this workshop, we will not be building a bot and instead focus on the API aspect of the cognitive service.

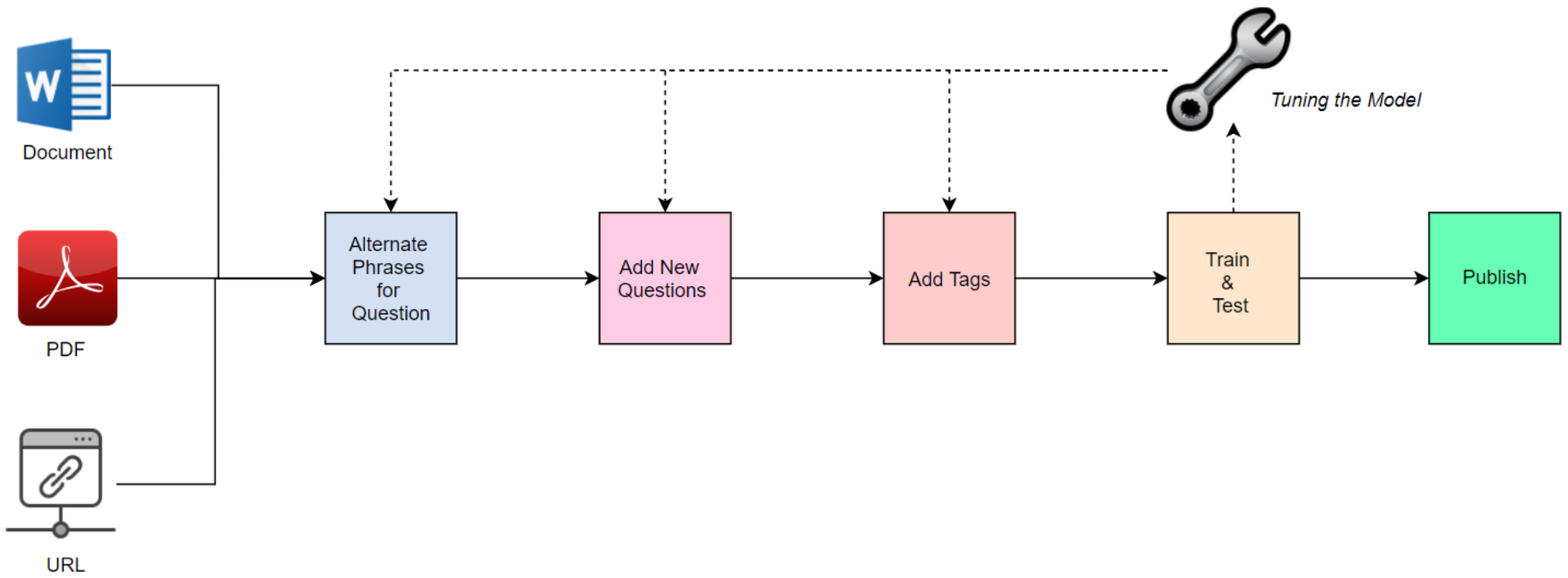
Step by step hands on guide to go from *Zero to Hero*

Pre-requisites

- Download & Install Postman
  - Postman is a free tool which allows you to make API calls
  - You can download the desktop application or get started using the web version (Download Postman | Try Postman for Free)
- An active Azure Account
  - You can use your current Azure Subscription or get started by creating a free trial account (https://azure.microsoft.com/en-in/free)
- Download the data for creating a Knowledge from Data Folder. Some data resources are also provided inline.

Let's get started!

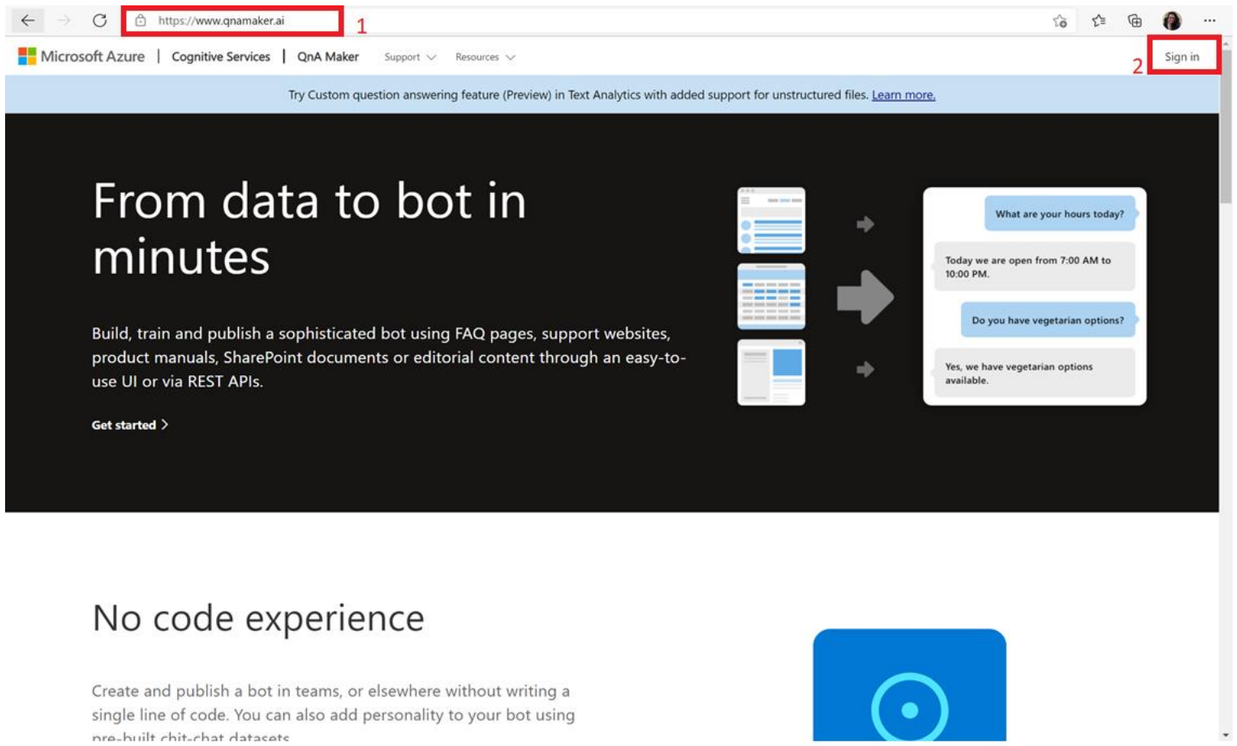
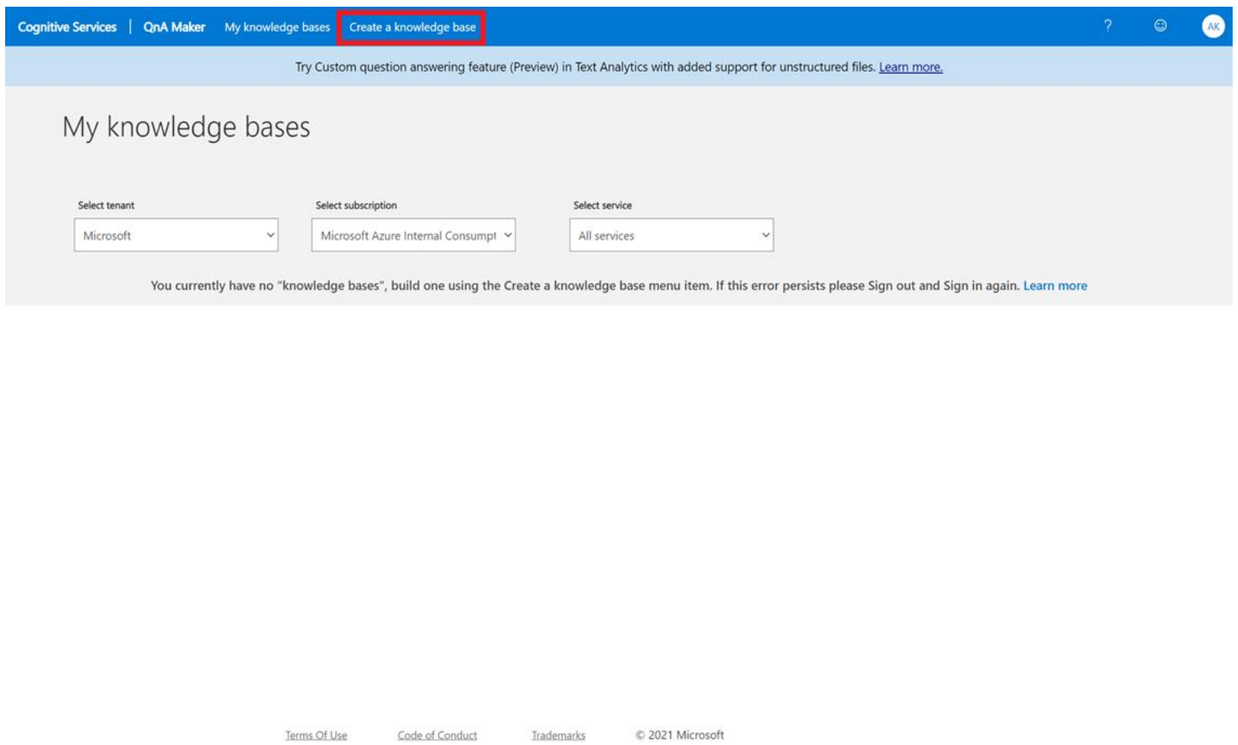
Knowledge Base creation flow

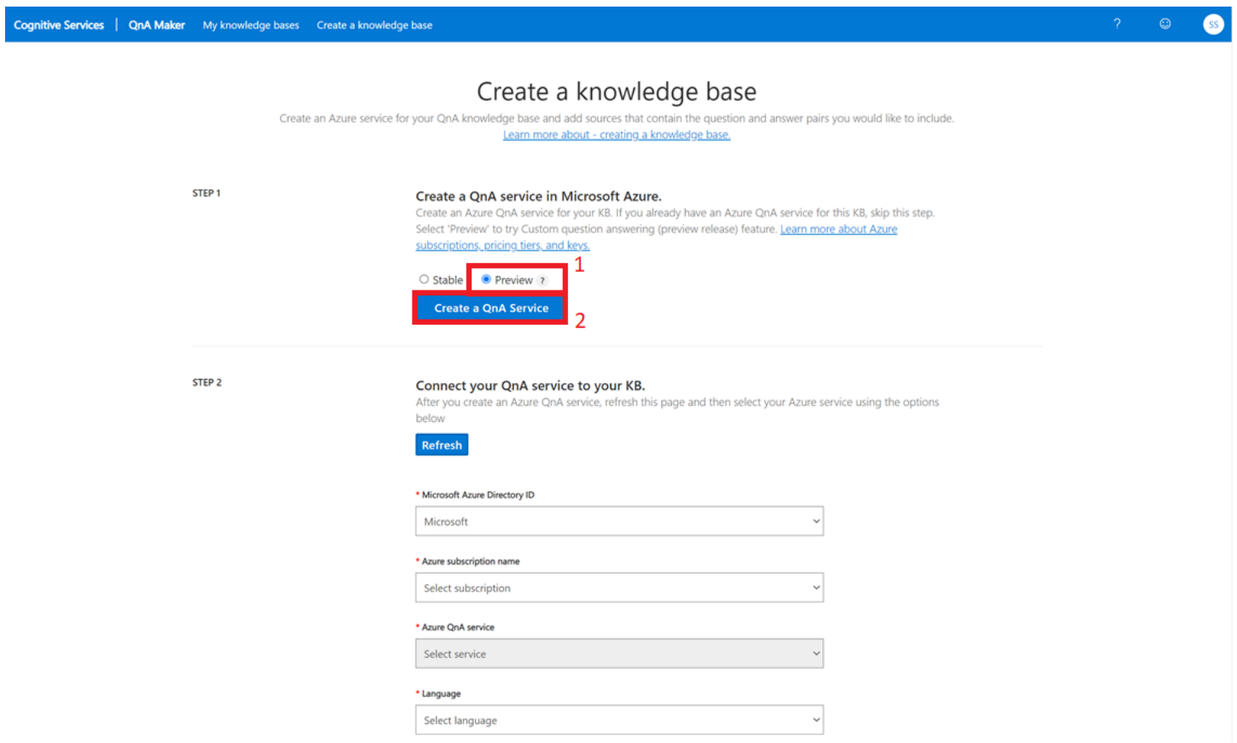


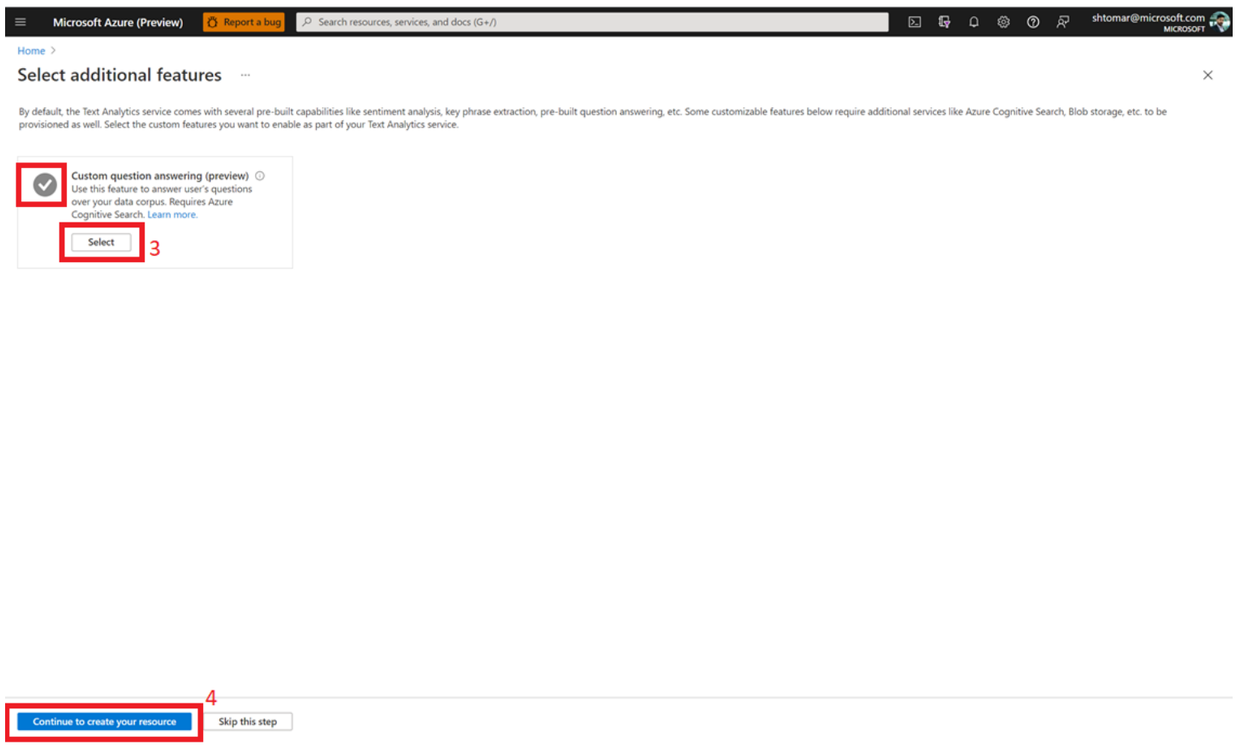
We will follow the steps as listed to create & publish a Knowledge Base :

1. Gather the data for creating the KB (URLs, PDFs etc)
2. Add alternate phrases for questions that were extracted from the input data & improve the answers that were extracted
3. Add new question & answer pairs, in addition to the ones extracted from input data
4. Add tags & metadata to the questions, to improve the search
5. Train & test the model
6. Publish the Knowledge Base to be leveraged for extracting answers in real-time using REST APIs

We have discussed each of these steps in details in the workshop.

Screenshots	Steps & Significance
	Go to QnA Maker Portal. Sign in using your Microsoft Azure Account.
	<u>Create a Knowledge Base</u> Select Create a Knowledge Base.





Clicking on the button will take you to the following page.

1. Select Preview
2. To spin up a QnA service resource, click ‘Create a QnA Service’. This will take you to Azure Portal.

If you already have a QnA service or Text Analytics service with Custom Question Answering enabled, you can go to the next step – ‘Connect your QnA service to you KB’.

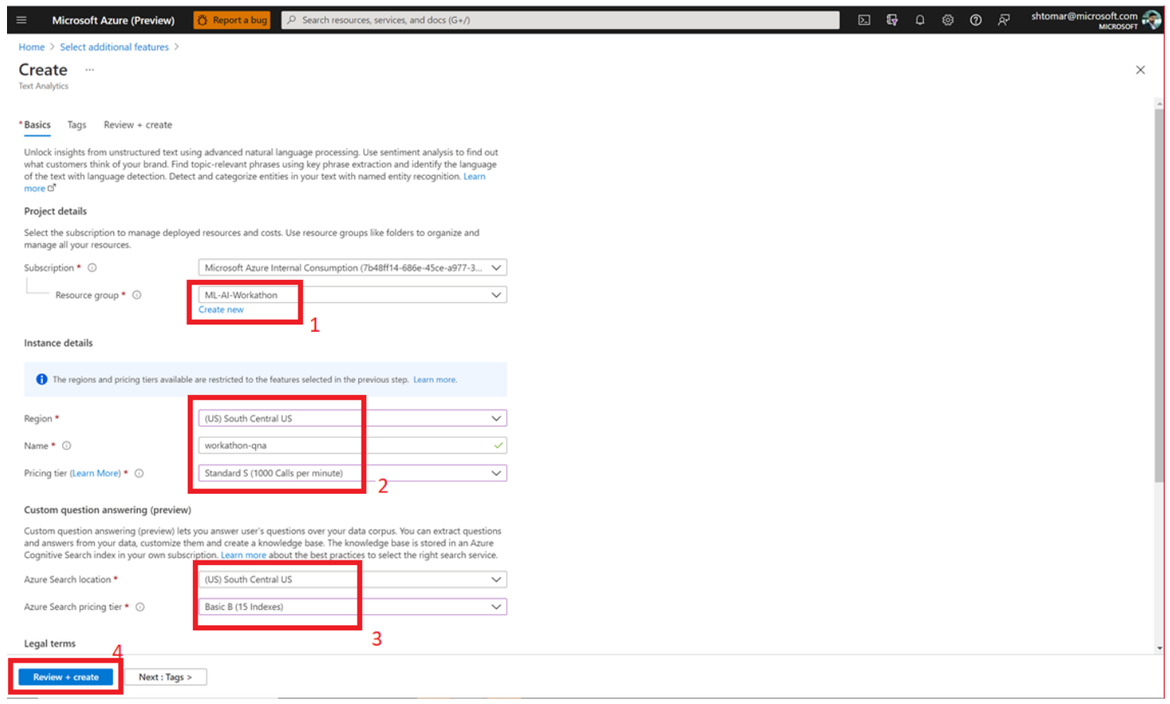
The QnA Maker resource can be provisioned in 2 ways –

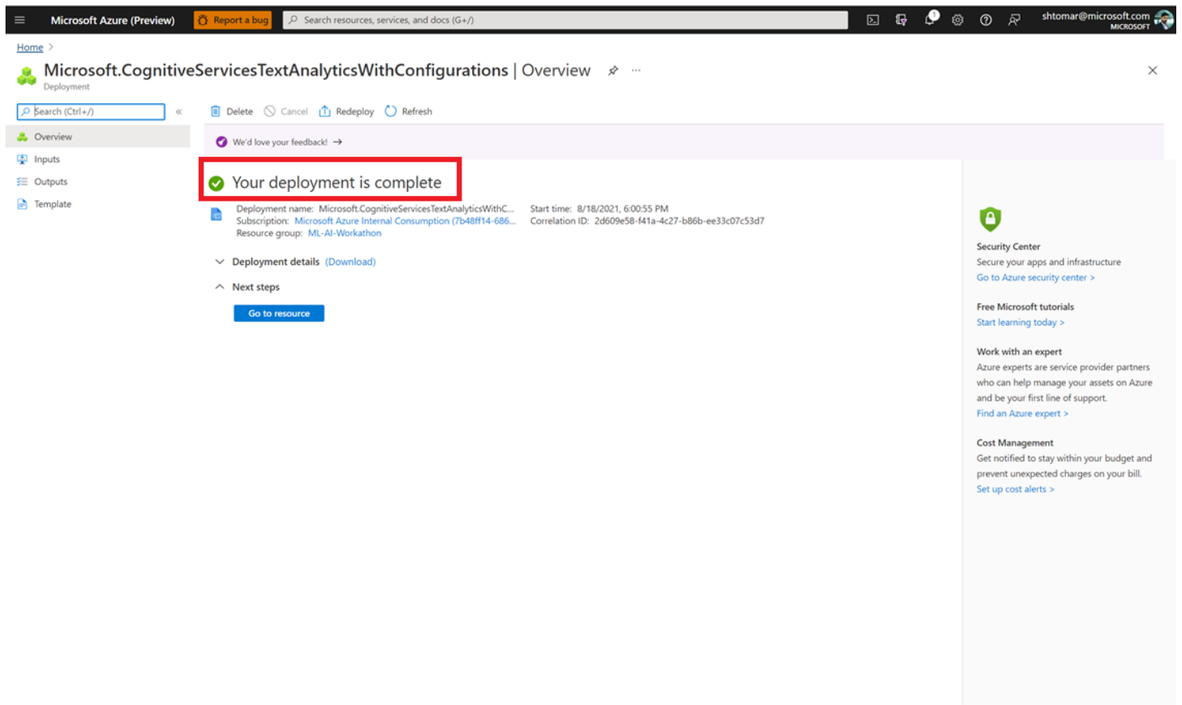
1. **Unmanaged** : This is the older way to provision QnA Maker. This will require you to spin up additional services like Azure Search, App Service, App service plan etc. These additional services have their respective capabilities to offer:
   a. App Service plan & App Service – to host the QnA Maker service
   b. Application Insights – to save telemetry & log chats
   c. Azure Search – to index your data for faster search

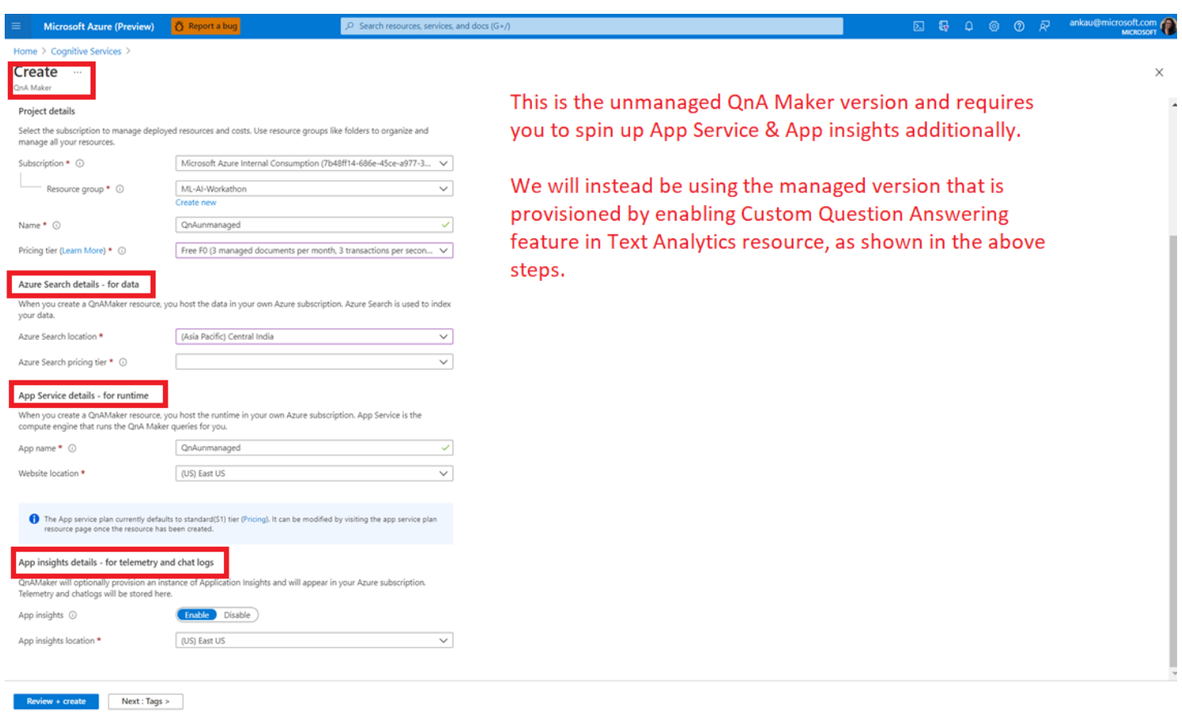
2. **Managed** : This happens through the new Custom Question answering capability in Text Analytics. Here, you just need to spin Azure Search as an additional service. Rest of the services will be managed by default for you.

In this workshop, we will be provisioning QnA Maker using managed approach. This allows us to leverage Text Analytics resource for QnA Maker.

3. In Azure Portal, click Select, so the box turns green
4. Click Continue to create your resource







This is the unmanaged QnA Maker version and requires you to spin up App Service & App insights additionally.

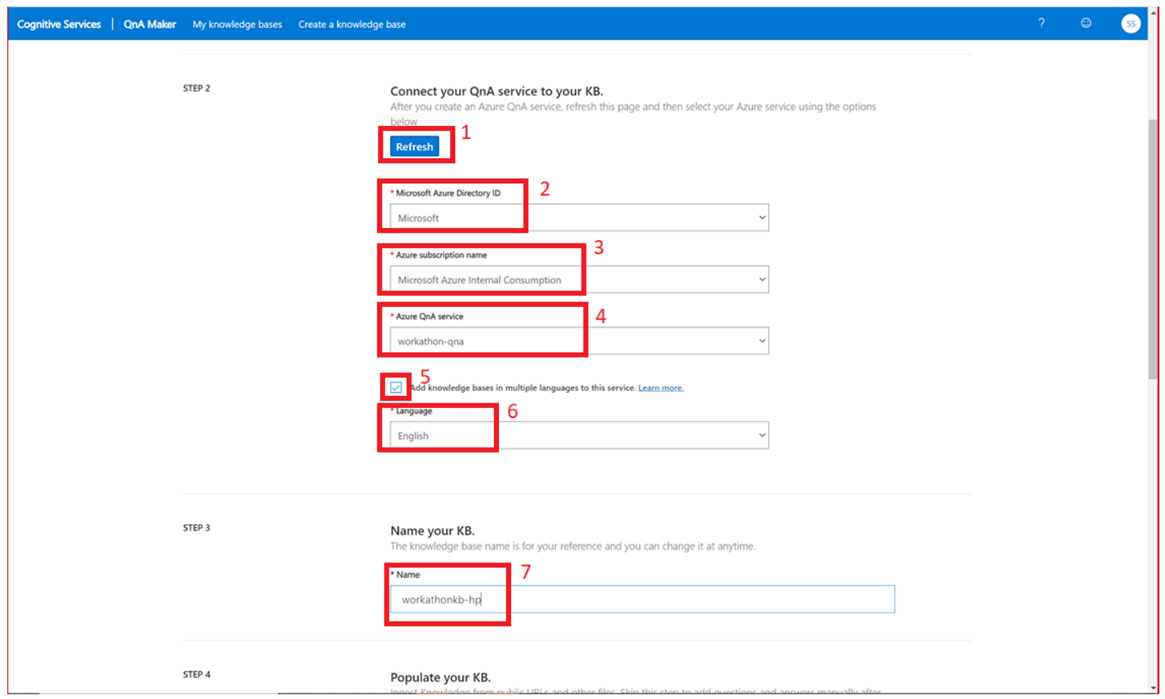
We will instead be using the managed version that is provisioned by enabling Custom Question Answering feature in Text Analytics resource, as shown in the above steps.

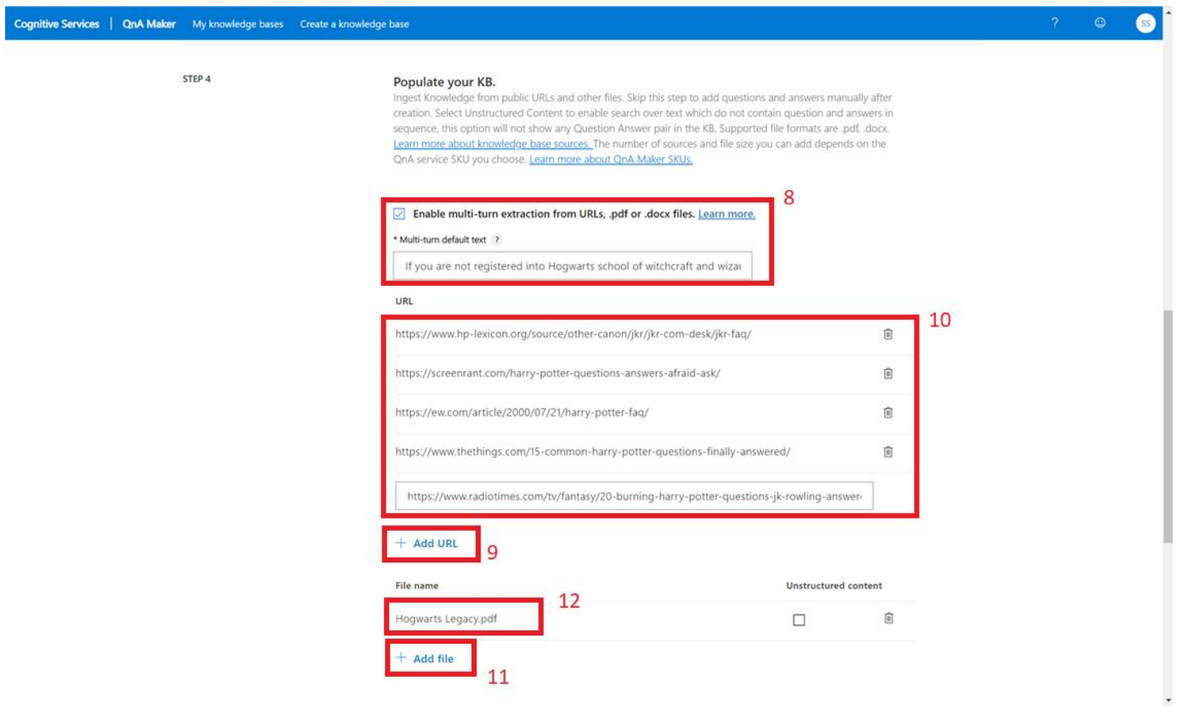
<u>Provisioning Text Analytics with Custom Question Answering enabled (Managed Approach)</u>

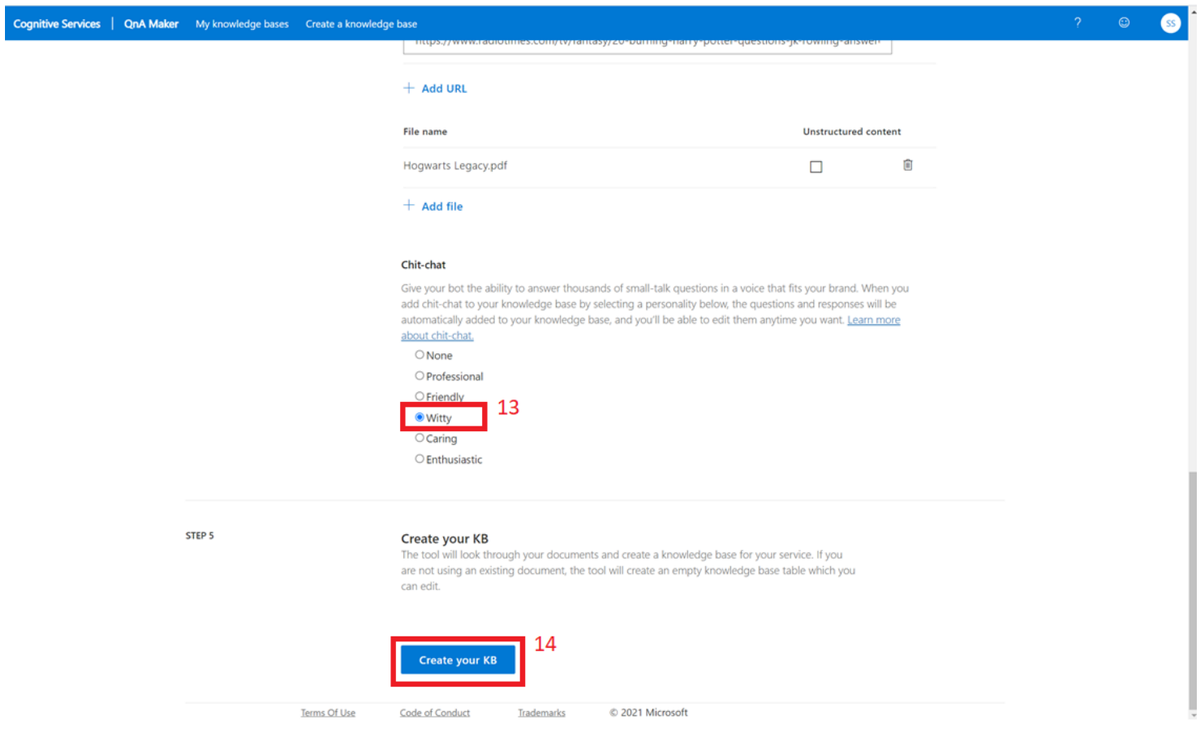
Provide the details for Text Analytics service as highlighted. Also tick the check box for Legal Terms & Responsible AI notice (step 1-3)

Click Review + Create (Step 4)

Verify the details and click create.
It will take a few seconds to minutes for your service to spin up.







Create a Knowledge Base

Once the service is deployed, go back to the QnA Maker portal and press refresh as shown in step 1 .

Select the AAD tenant, Subscription Name and QnA Maker resource that you just provisioned (step 2-4).

Select the Multiple language check box as shown in step 5.
This will allow you to build multiple Knowledges Bases in different languages under the same QnA Maker resource.
Each time you create a new Knowledge Base, you will be asked to choose the language for it.

Not selecting this will restrict you to build all the Knowledge Bases in the same language, in this particular QnA Maker resource.

In step 6, choose the language you want to build the Knowledge Base in (English in our workshop).

Give your KB a name of your choice (step 7).

Check 'Enable multi-turn extraction' (step 8).
When you select this option, QnA Maker extracts the hierarchy present in the document structure. The hierarchy is converted to follow up prompts and the root of the hierarchy serves as the parent QnA. In some documents the root of the hierarchy does not have content which could serve as an answer, you can provide the 'Default Answer Text' to be used as a substitute answer text to extract such hierarchies.

Keep clicking add URL as you keep adding more & more URL content to your knowledge base. Once you have added all URLs, you should have a view like step 10.

You may use the following URL links to build a Harry Potter fan base KB, that is used in this workshop :

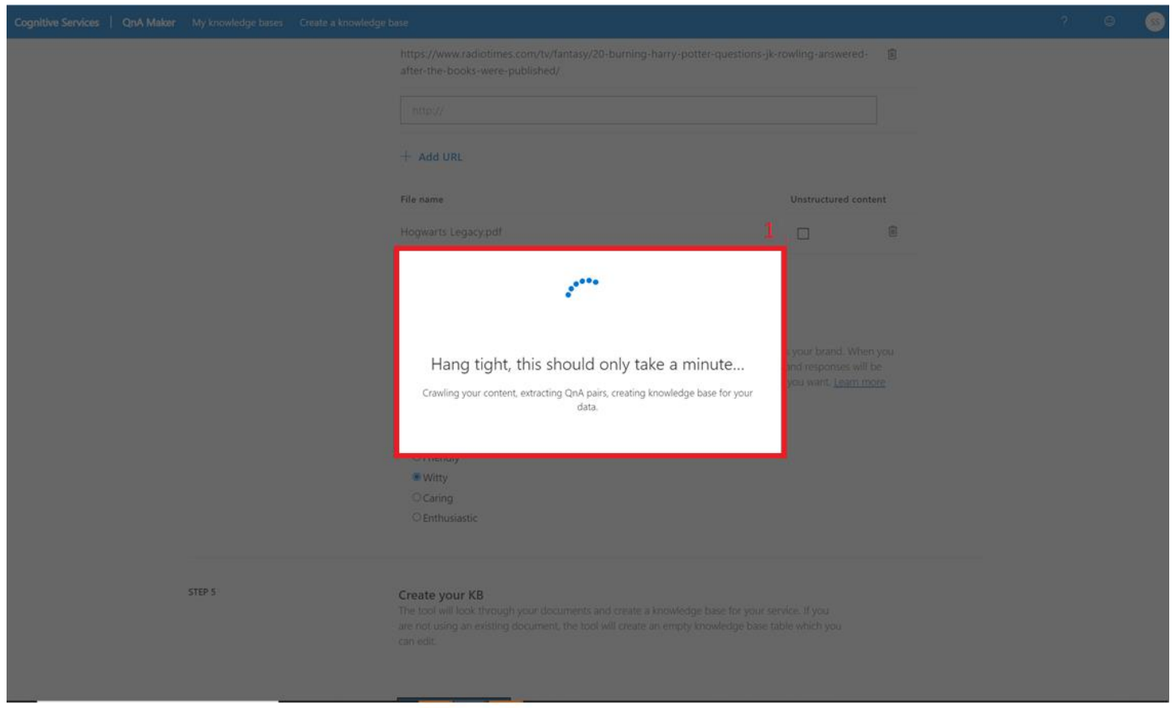
1. https://www.hp-lexicon.org/source/other-canon/jkr/jkr-com-desk/jkr-faq/
2. https://screenrant.com/harry-potter-questions-answers-afraid-ask/
3. https://ew.com/article/2000/07/21/harry-potter-faq/
4. https://www.thethings.com/15-common-harry-potter-questions-finally-answered/
5. https://www.radiotimes.com/tv/fantasy/20-burning-harry-potter-questions-jk-rowling-answered-after-the-books-were-published/

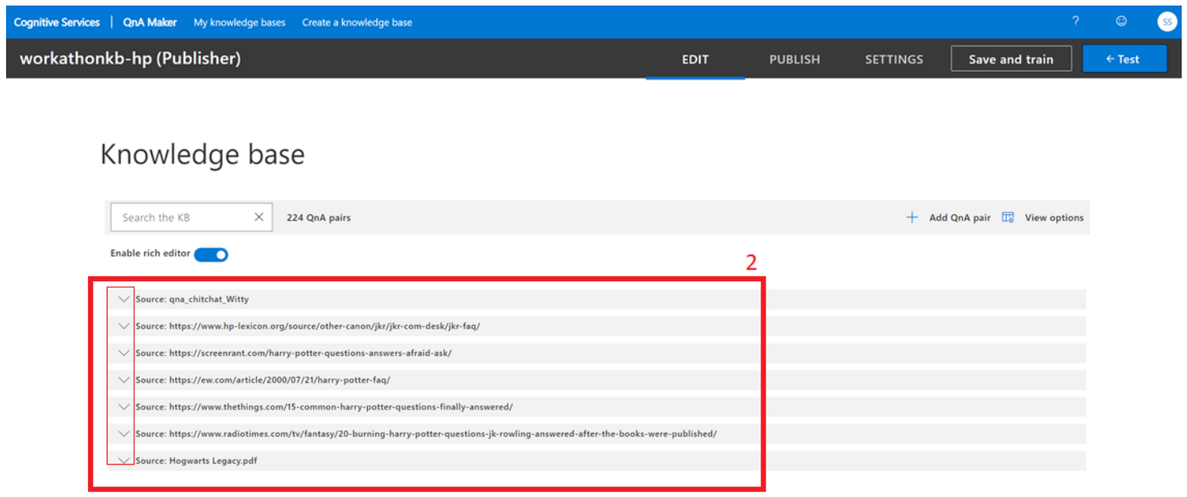
Alternatively, you may add files & documents as well. Select Add File (step 11) and browse to the required file location. You will see the uploaded files as shown in step 12.

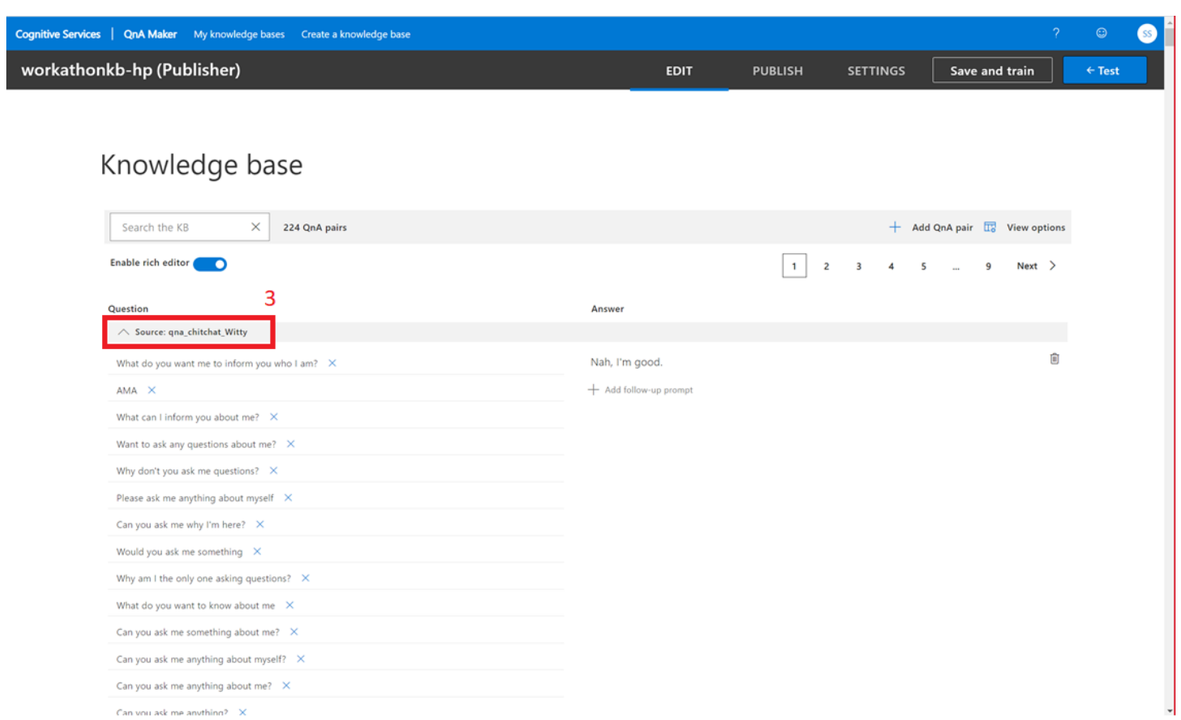
You may use the 'Hogwarts Legacy.pdf' file that you downloaded from the Data folder.

Chit-Chat : You can also give your bot a personality. This adds thousands of pre-built question answer pairs, to handle any miscellaneous inputs that the user might give. The chit-chat helps you define the tone and character for your bot.
Since this is a fun use case, we will be choosing Witty personality (step 13).

Hit Create you KB (step 14).





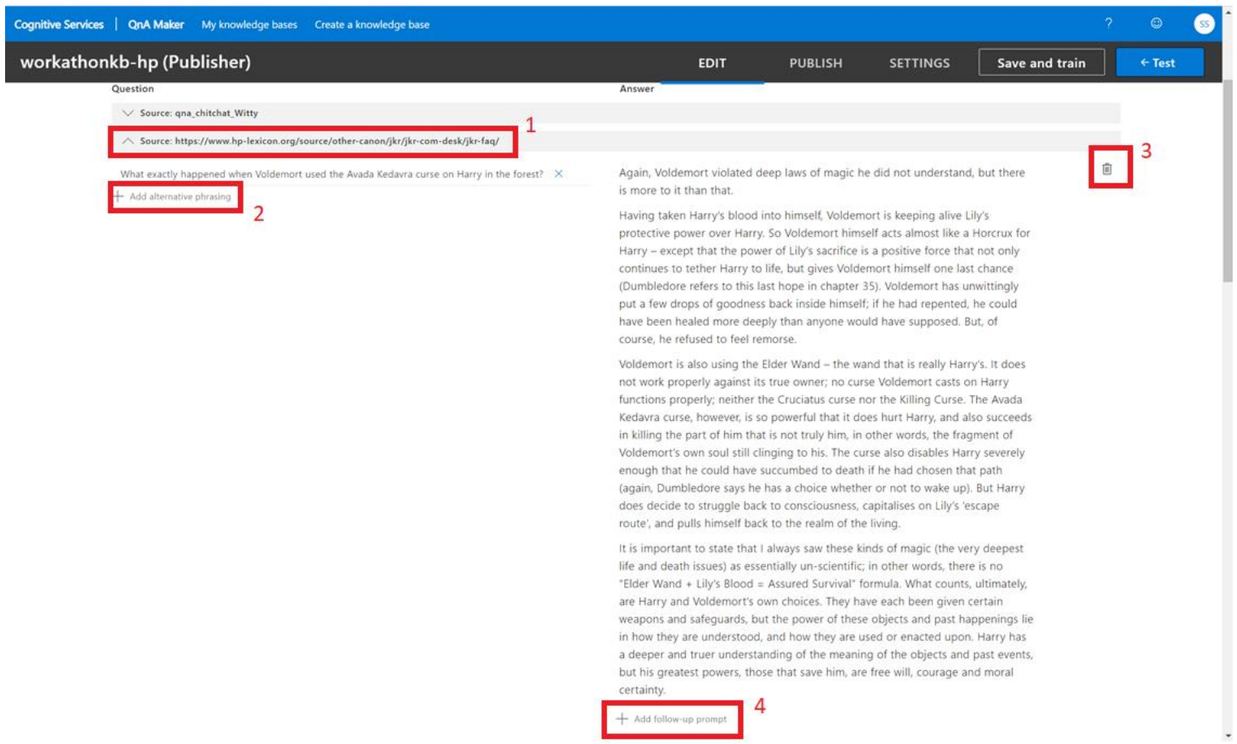


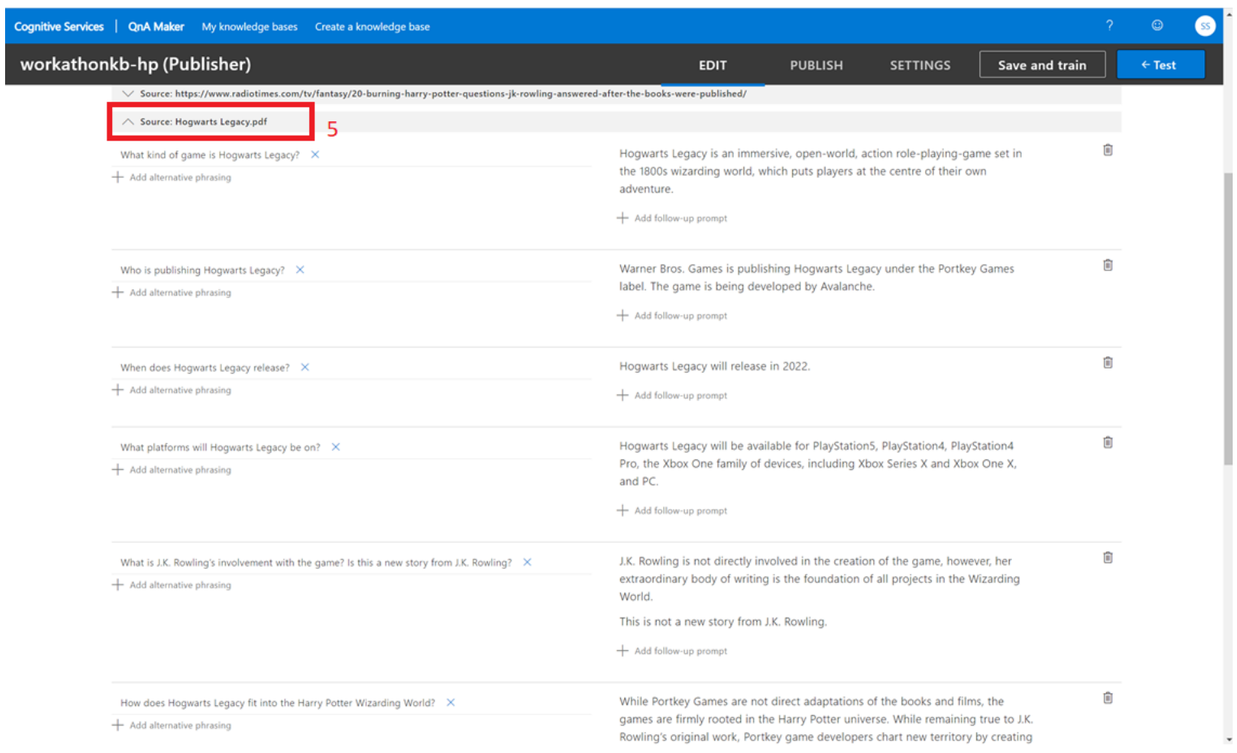
Exploring the Knowledge Base

1. The model will now extract question & answer pairs from all the input sources. This could take a while!

2. Once QnA pairs are extracted from all your input sources, if you collapse them all using the arrow button to the left of each source, you will be able to see all the sources you added while creating the KB & also the QnA pairs added because of the Chit-chat personality we chose (qna_chitchat_witty)

3. Expand the qna_chitchat_witty source, to observe the numerous question answer pairs added automatically by the model, to give the bot a witty personality.





Editing the extracted QnA Pairs

Expand any of the other input sources, to explore the options available to edit and further improve your knowledge base (step1).

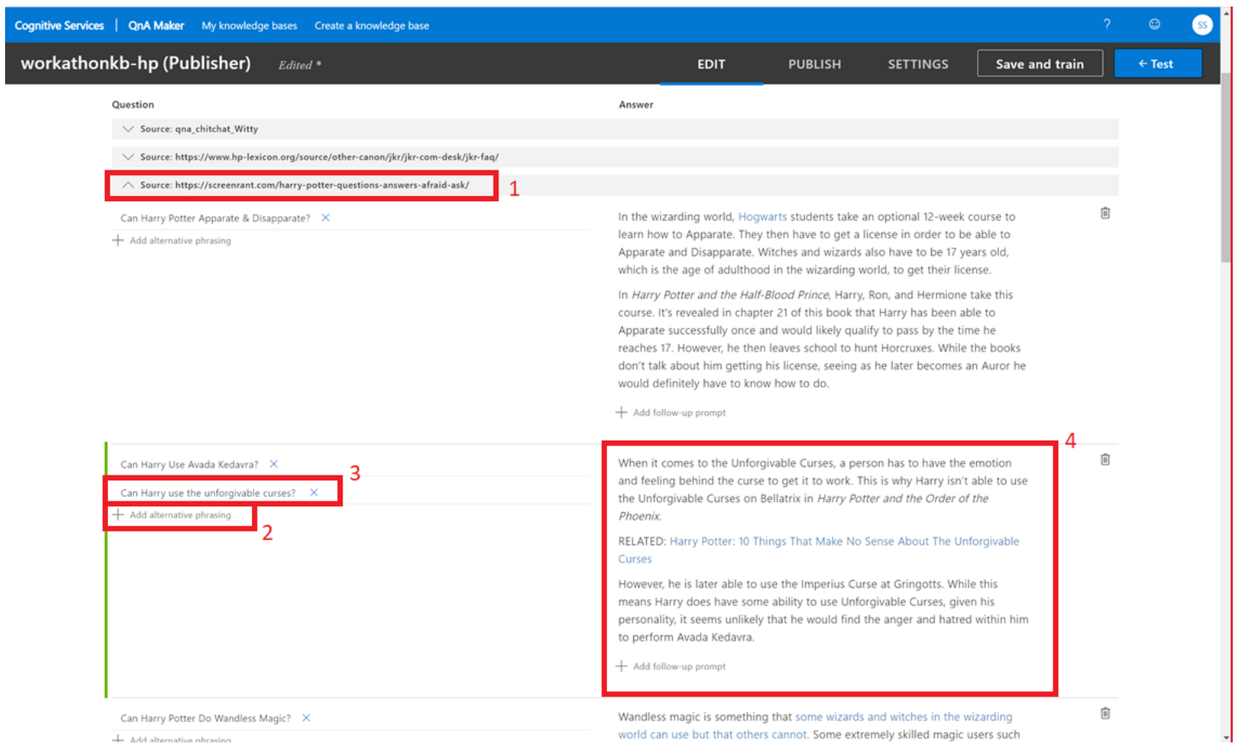
Alternative Phrasing : You can add alternate phrases for an input question, if you feel the user could ask the same question in a different way or using different words (step 2).
[The user does not have to input a 100% matching question to get a response. Even if the user types in a part of the question or makes a typing error, the model is intelligent enough to find the intent of the question and pick the right answer. However, adding alternate phrases will increase the accuracy and confidence of the returned answer.]

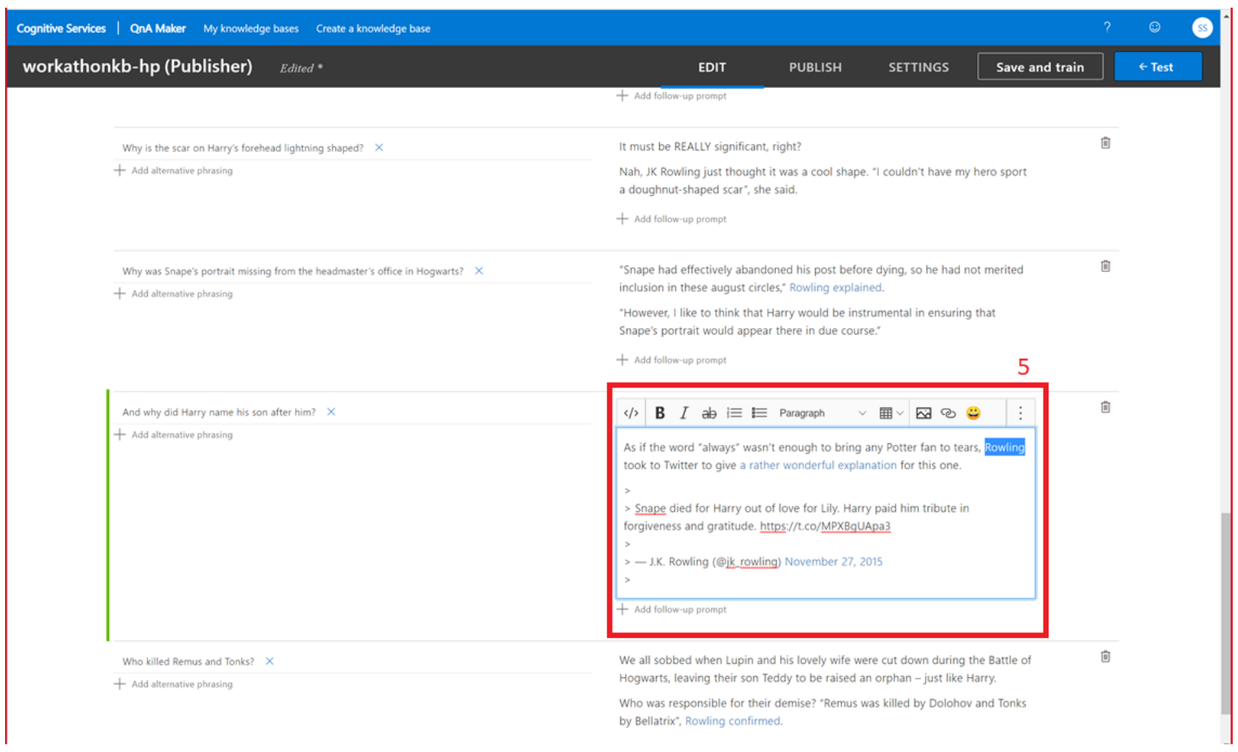
Deleting QnA pairs : If you feel a particular QnA pair is not needed, you can delete it as shown in step 3.

Add follow-up prompts : Using this feature, you can link QnA pairs. If you add follow-up prompts to a particular answer, post answering that question, the model will pose the follow-up questions to the user. This can help in keeping the conversation going.

Similarly, explore the QnA pairs extracted from the other sources.

As shown in step 5, notice how QnA pairs are extracted from documents and URLs alike.



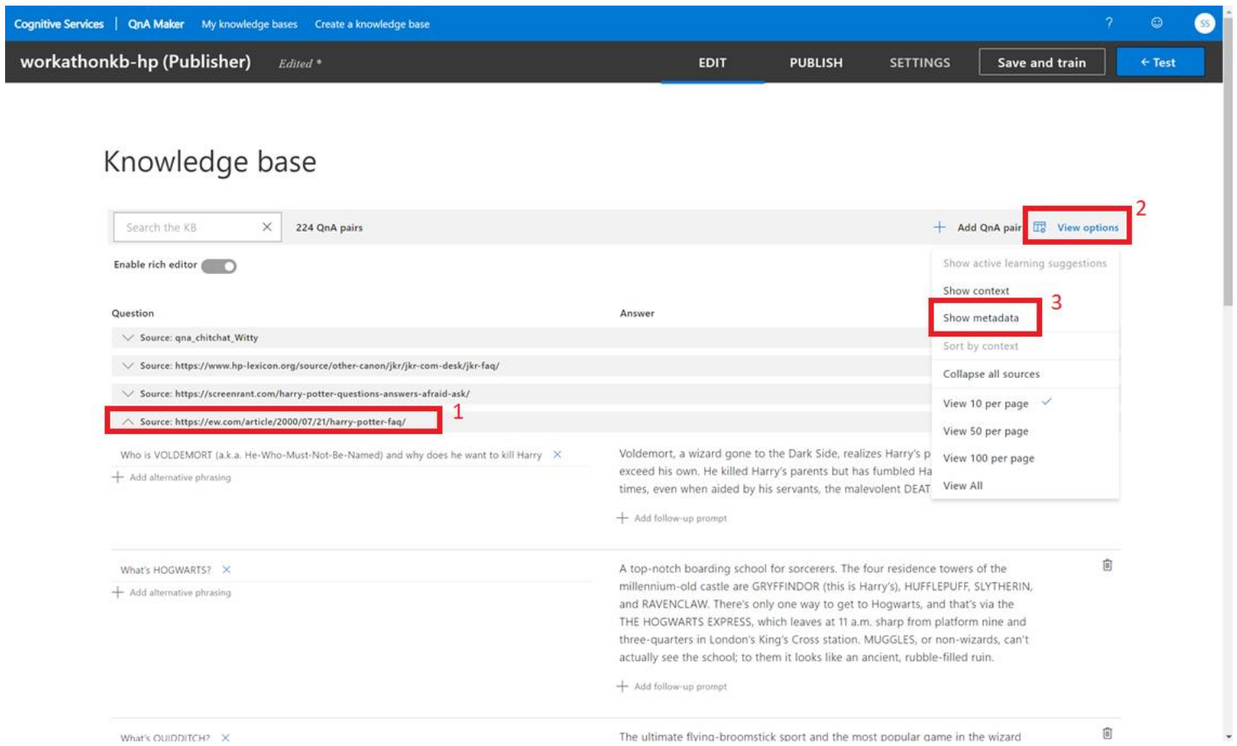


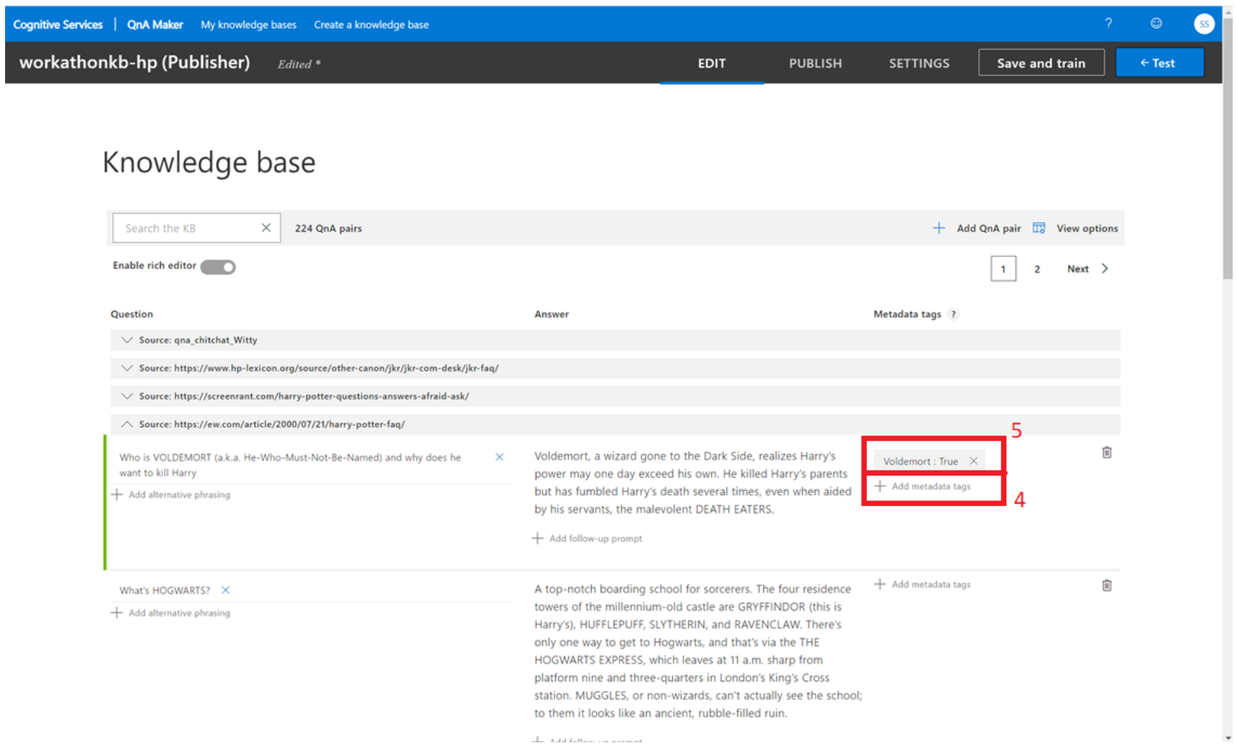
Add Alternate Phrasing

Steps 2,3 show how adding an alternate phrase will come up.
Questions similar to either of the two questions will now correspond to the answer as highlighted in step 4.

Editing responses

You can edit and beautify the response in several ways. (Step 5)
You could improve the aesthetics by using options like Bold, Italics, Strikethrough.
You could add bullet points, numbering, paragraphs, tables, images, links, emojis and a lot more to improve the answers!





Adding Tags & Metadata

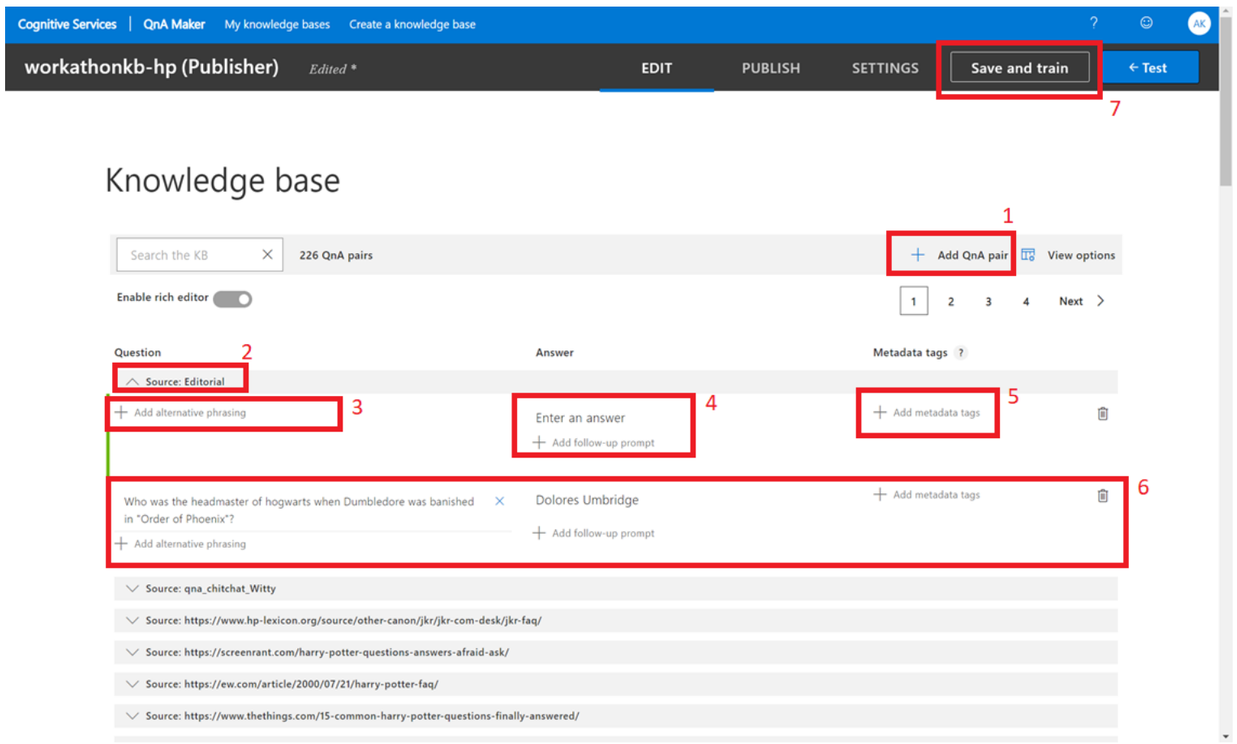
Adding metadata & tags allows you to filter the answers by these metadata tags and reduce the number of question answer pairs to be searched, thereby, increasing accuracy & confidence of your model.

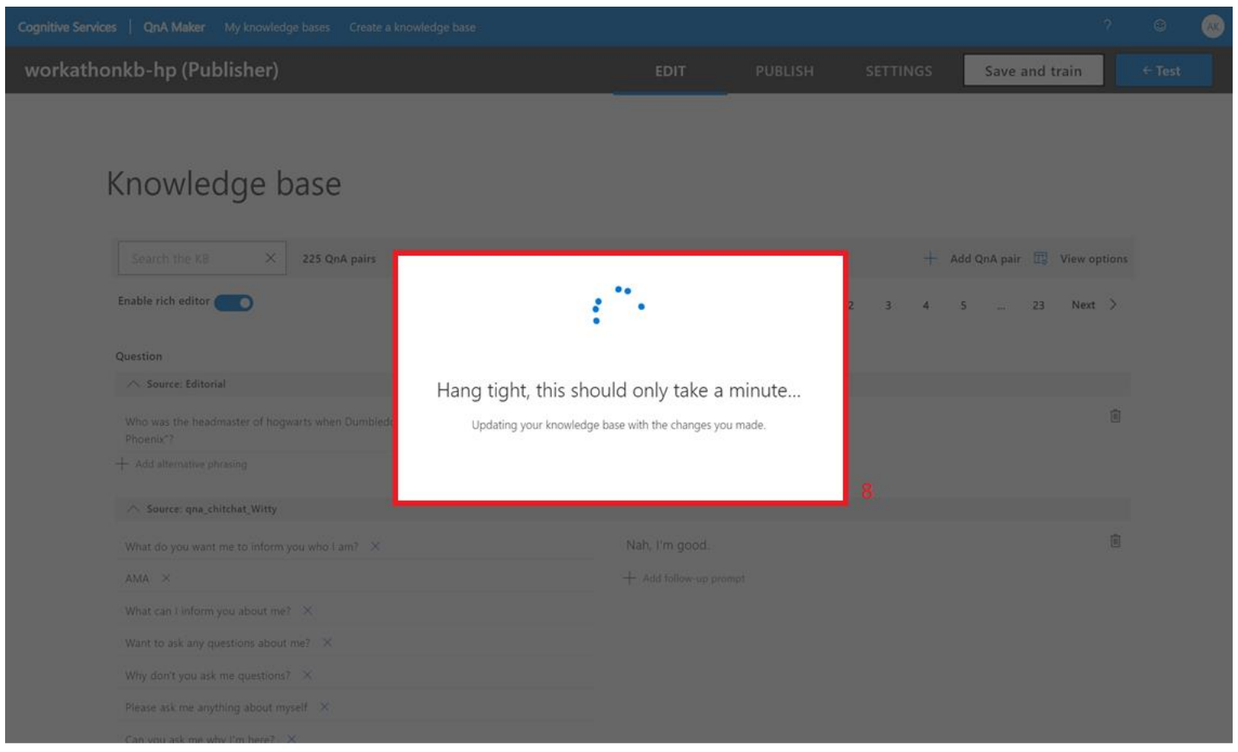
Metadata adds the ability for a client application to know it should not take all questions but instead to narrow down the results of a user query based on metadata tags. This is also useful in cases where the knowledge base answer can differ based on the metadata tag, even if the query is the same.

To add metadata tags, follow the steps as highlighted :

1. Navigate to the concerned source & question
2. Select view options
3. Select Show metadata

4. Add the key value pair for each tag
5. These will be visible as shown. You can also delete the tags if they aren't required.





Adding additional QnA Pairs

If required, you can also add additional custom QnA pairs directly in the Knowledge Base.

To add additional questions, follow the steps as highlighted :

1. Select Add QnA pair
2. This adds a section with the name as 'Source : Editorial'
3. Type in the question by clicking on Add alternate phrasing
4. Type in and beautify the corresponding answer
5. You can add metadata tags as added earlier

All functionality about adding alternate questions, follow-up prompts, metadata tags also apply to manually added questions.

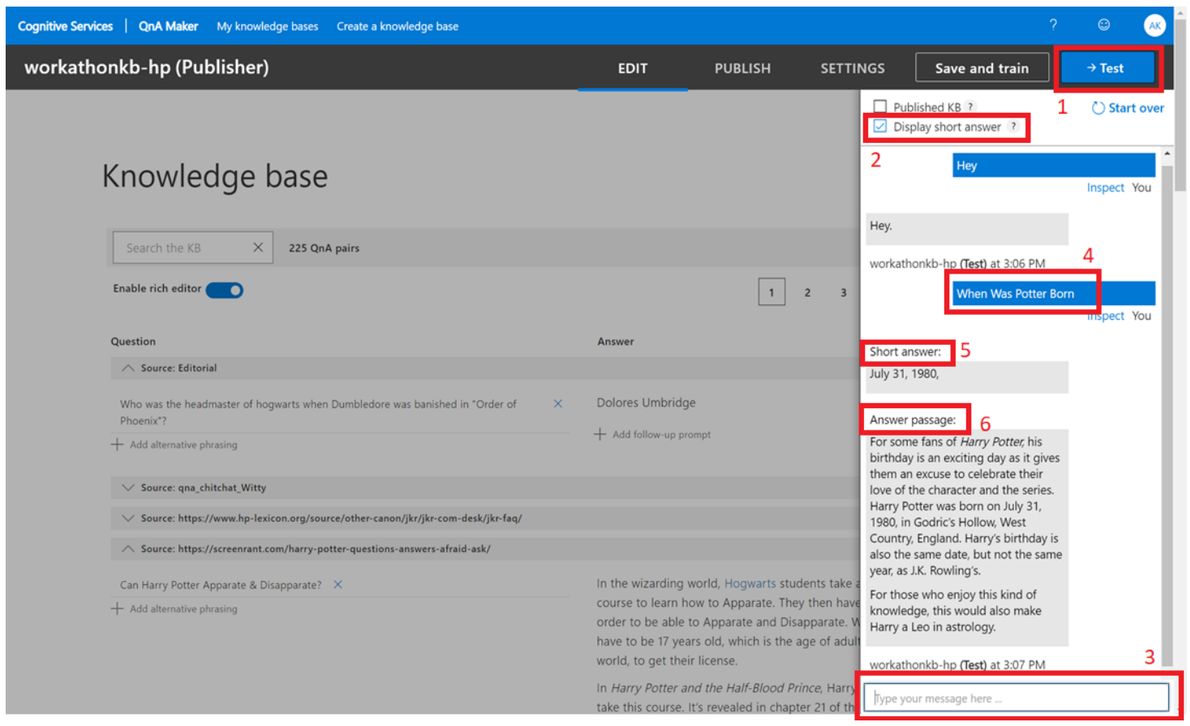
After you add a QnA pair, it will be visible as shown in step 6.

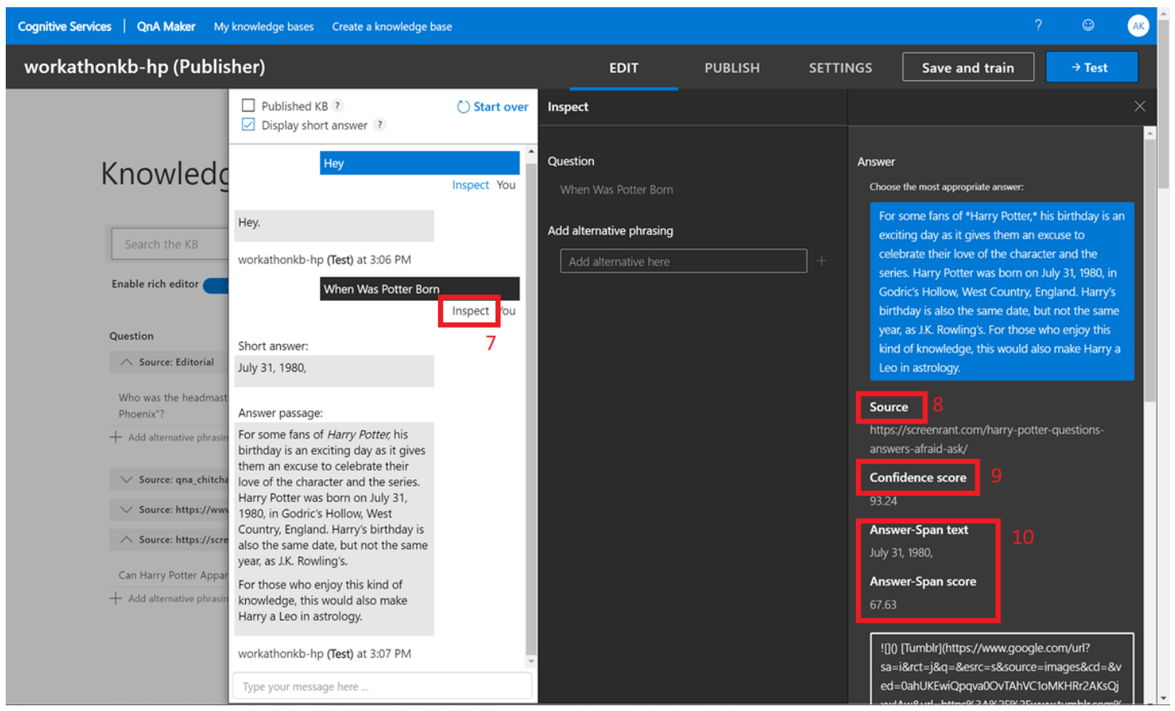
Training the model

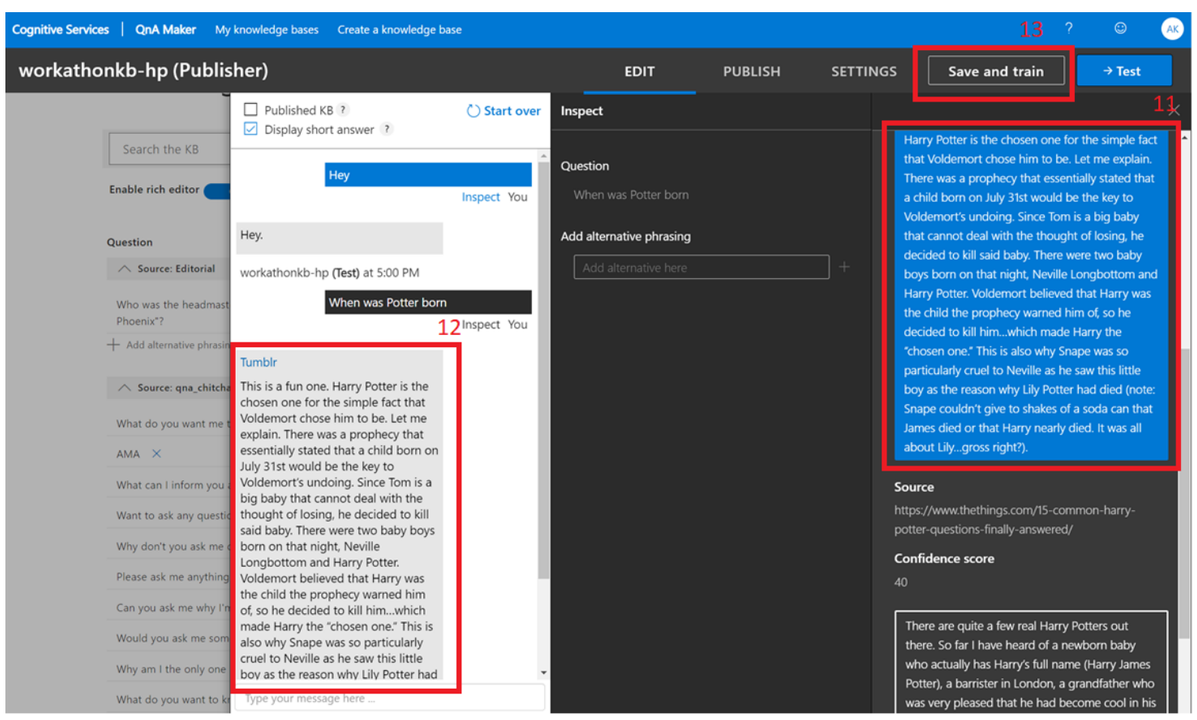
Once you have explored the QnA pairs, made the required changes and are satisfied with it all, select Save and Train (step 7).

This will take a while.

The model is now learning & training using the QnA pairs we provided. It is also indexing the terms, to provide results faster.







Testing the model

Once your model is trained, you can test it from within the portal itself. Click on the Test button to open the Test pane and perform a quick test (step 1).

Check the Display short answer option.
[**Significance** : If you have a long answer for a particular question, you have 2 choices. You can either extract the complete answer or you can enable short answers and have the model intelligently pick out the part from the answer that best caters to the question]

Type in a couple of questions to see how the bot responds (step 3).

Notice the response for the question 'When was Potter born'

The model returned the original answer as it is and since we had short answers enabled, it very intelligently picked up just the date from the answer paragraph, just like we humans would have! Hence the human parity in the QnA maker models.

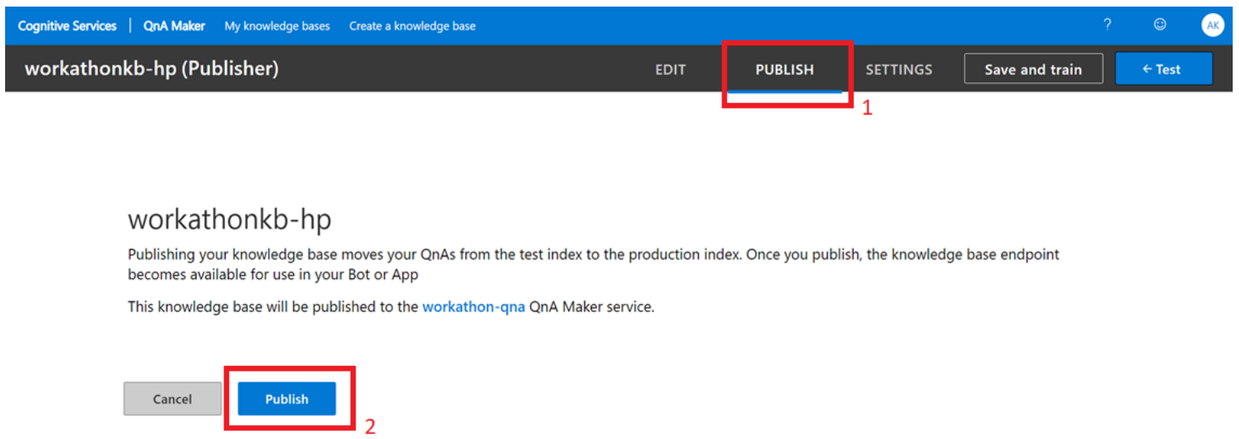
Inspecting the answers

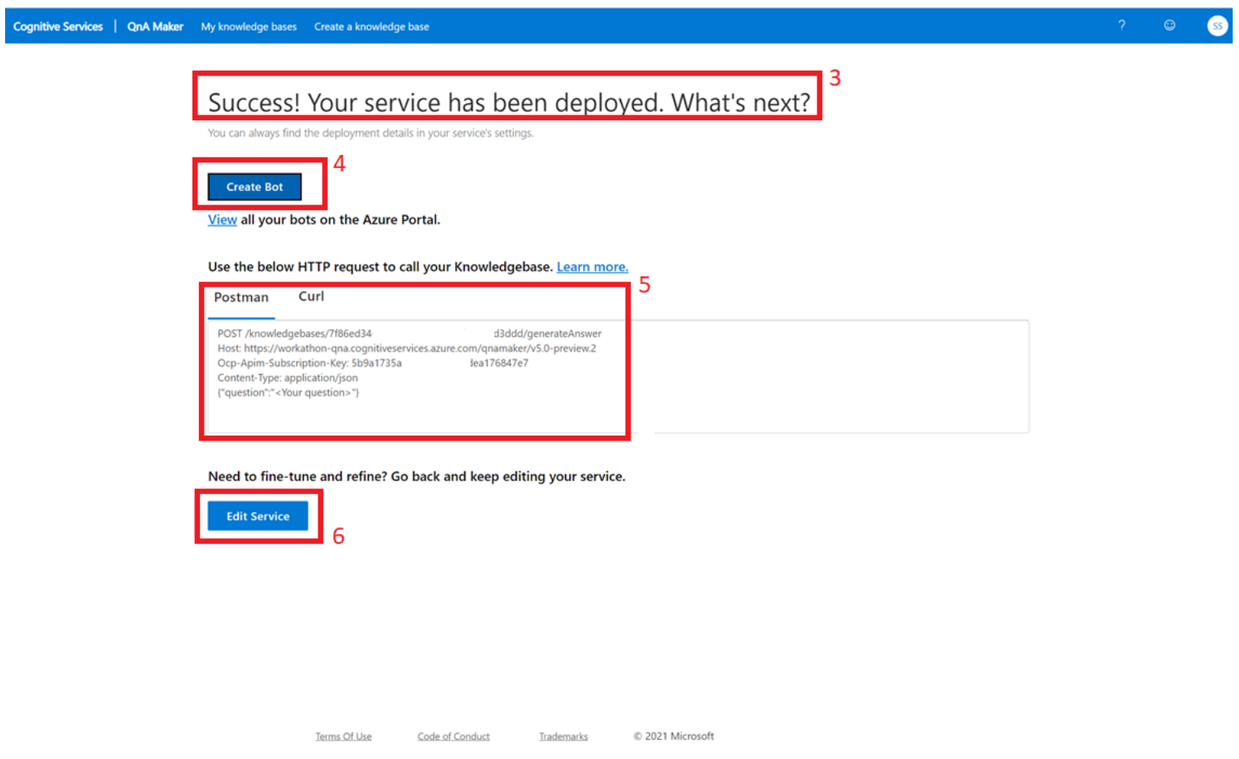
You can further inspect the responses and confidence score for each response by clicking on Inspect option below the question (step 7).

The inspection pane provides you with the source of the question (step 8), the confidence score of the full text (step 9) and the confidence score & text for the short answer (step 10).

If the answer is incorrect or displays a very low confidence score, you can choose the correct answer from the options that are there below in the pane (step 11).
This also changes the answer in the chat window (step 12).
Now, again Save & train (step 13) the model.

This will further tune the model and increase the accuracy of the outcome.





Publishing the model

Once you have trained & tested the model, it's time to publish the model, so we can leverage it in our applications.

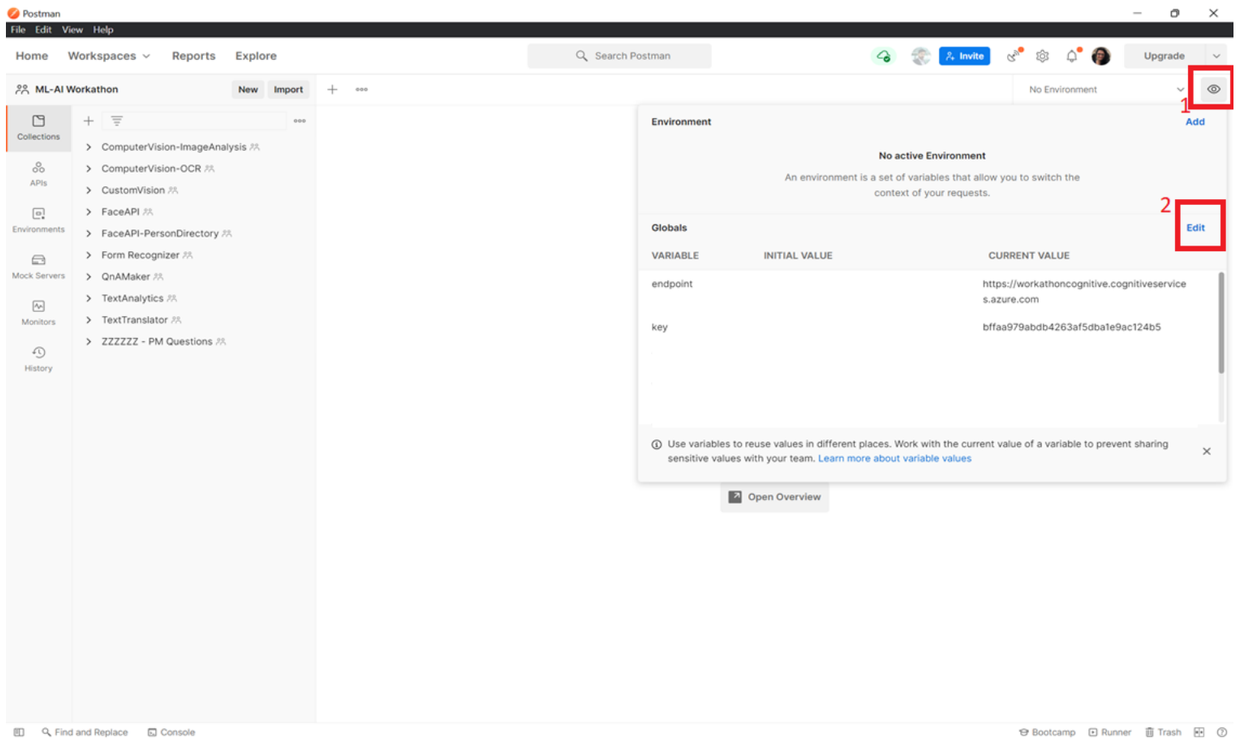
Go to the Publish tab (step 1) and click the Publish button (step 2).

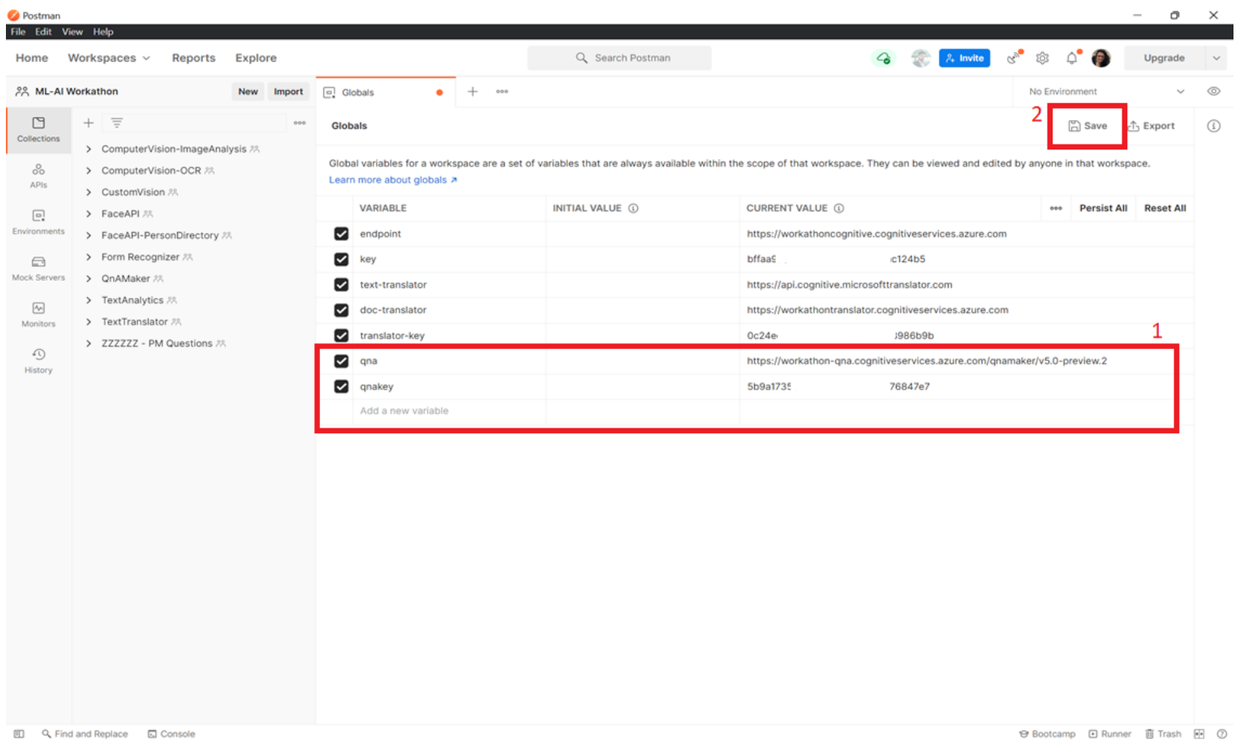
This might take a few seconds. Once published successfully, you will get a Success message as highlighted in step 3.

If you wish, you can Create a Bot (step 4) straight from here. You will be redirected to the Azure Portal and will have to provide some information in terms of the Bot name, Region, App service plan and so on.

In this workshop, we will be testing out the QnA maker endpoint in real time by making a REST API call in Postman.
As highlighted in step 5, copy the HOST, Post endpoint, Ocp-Apim-Subscription-key and Content-Type to a notepad.
We will be leveraging these values to make the REST API call.

Edit Service : Once you have published the model, you can further Edit and fine tune it (step 6).
This will take you back to the KB page. You can add questions, alternate phrases, edit answers, add metadata tags etc and retrain & republish the model.
This will not change the model endpoint. It will just change and improve the underlying KB. Thus, allowing you to keep improving your model without making any changes to your application or endpoint.





<u>Testing through REST API</u>

We have now switched the interface to Postman to explore the QnA maker model we just created. If you haven't downloaded the Postman client, you can use web version.

<u>Configure global variables in Postman</u>

Significance :

The global variables will be created once and leveraged time & again, in each API request that we make through Postman.

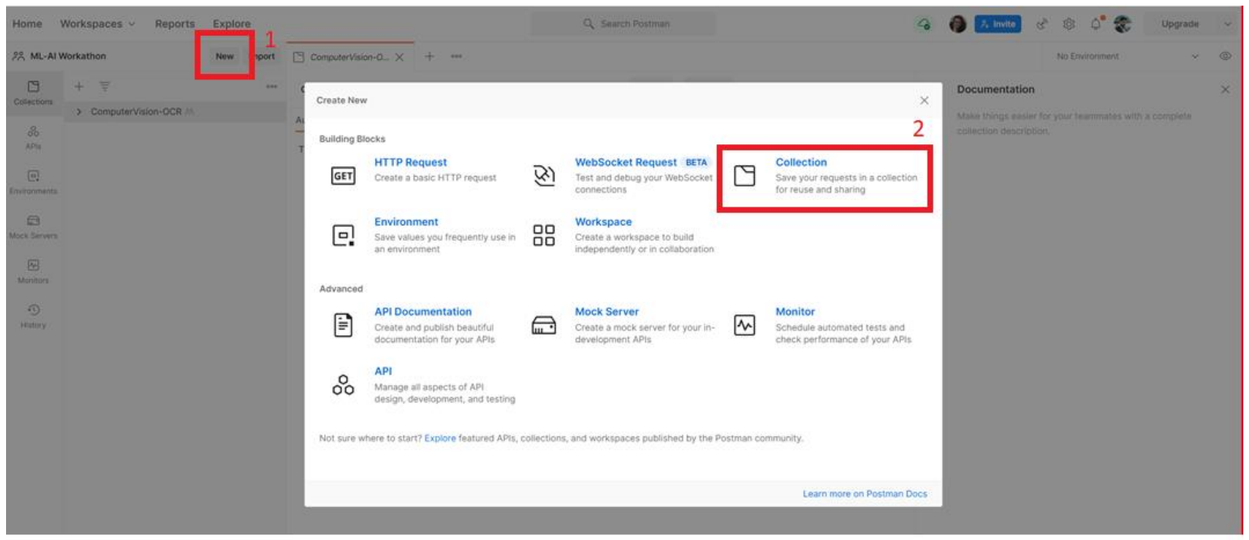
This way, you will not have to hard code the Endpoint & Key for every request you make, thereby, making it more secure. This will also save time and effort.

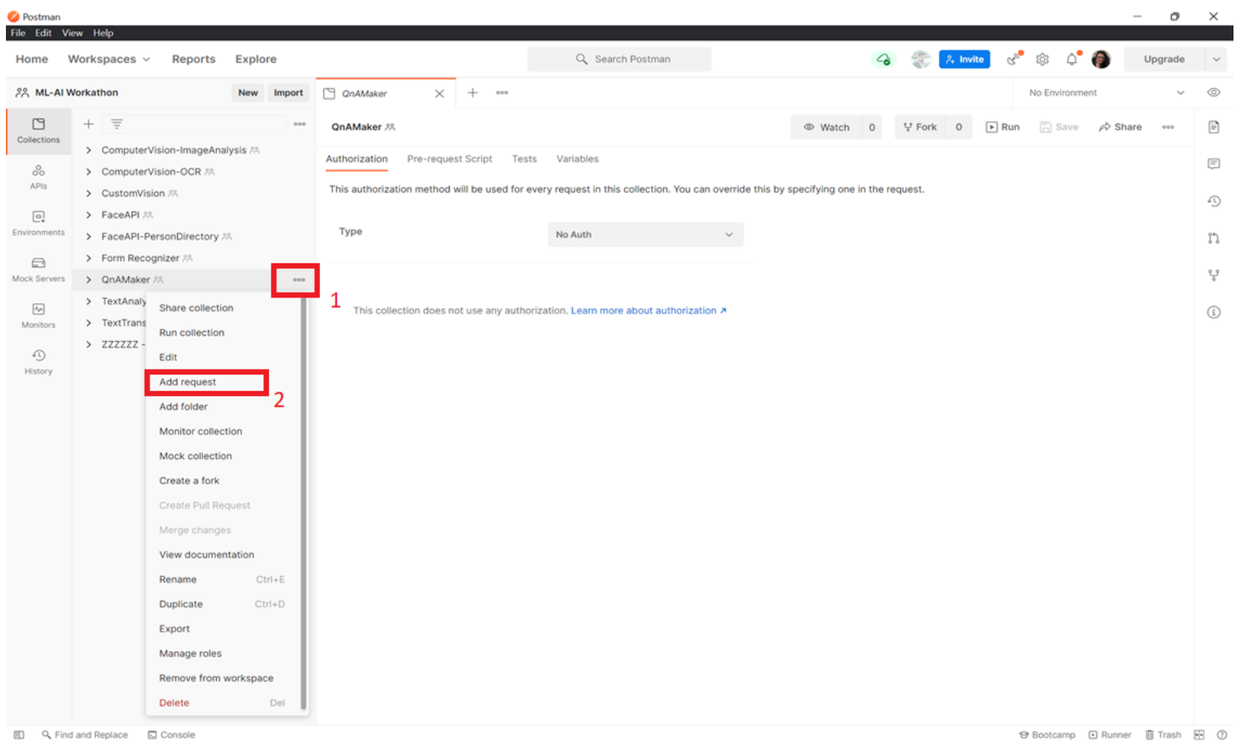
Follow step 1 & 2 to add global variables for :

1. qna – Paste the HOST endpoint copied in previous step
2. qnakey - Paste the Ocp-Apim-Subscription-key copied in previous step

Once you have added the global variables, they will appear in your global variables section, as shown in image 2.

Click Save.



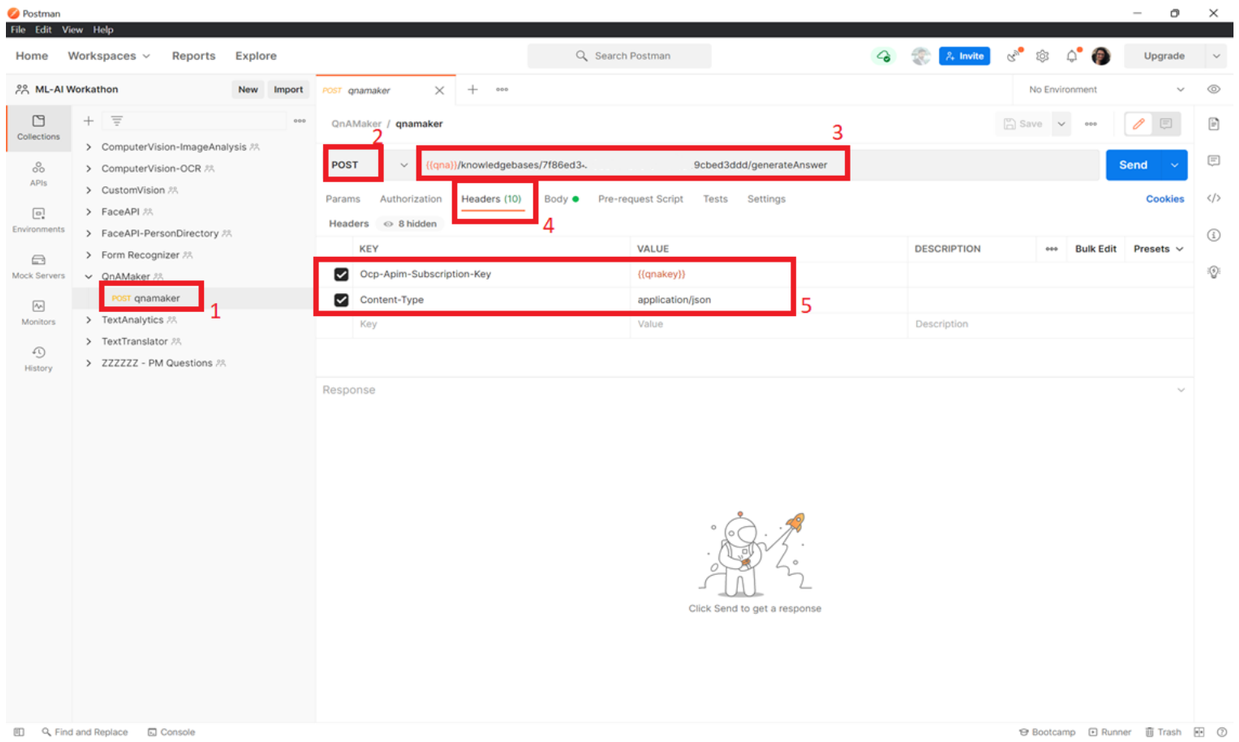


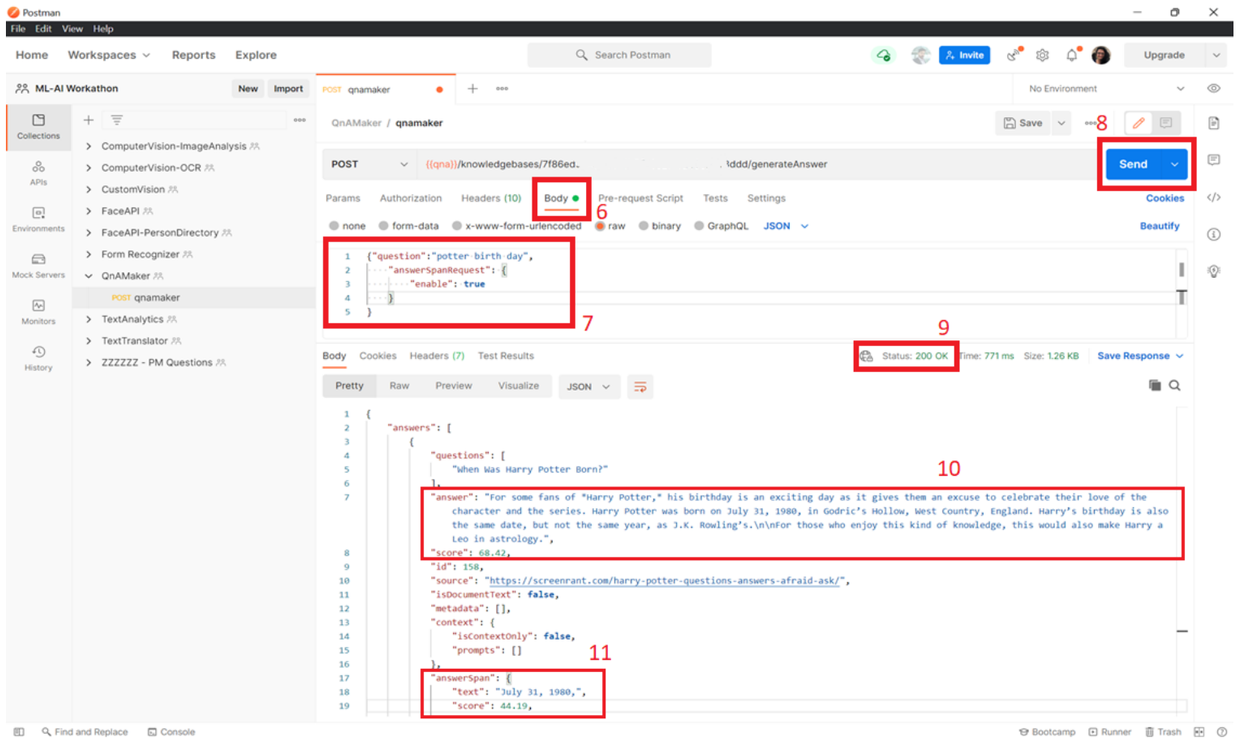
<u>CREATE NEW COLLECTION</u>

Open Postman > select New.
On the pop up select Collection.
Name the collection QnAMaker.

Collection is like a folder for managing the API call requests.

Once you have created the collection, follow steps 1 & 2, to create a new request.





Real time QnA Maker REST API Call

This request upon successful execution will return answer for the question asked, corresponding confidence score, source, short answer and other details.

URL : {{qna}}/knowledgebases/your-kb-id/generateAnswer
Replace the endpoint after *{{qna}}* with the post endpoint copied earlier or simply copy the kb-id from there and replace here

Headers :
Ocp-Apim-Subscription-Key : {{qnakey}}
Content-Type : application/json

Body :

```
{"question":"potter birth day",
   "answerSpanRequest": {
      "enable": true
   }
}
```

You should also try exploring with different questions basis your use case.

Significance of input & output

1. {{qna}} , {{qnakey}} : Values being picked from global variables

2. Ocp-Apim-Subscription-Key : This is the QnA Maker resource key, that will authenticate the request.
   Content-Type : This refers to the input type that you provide in the body, for eg application/json allows you to enter body text in JSON format.

3. Body :
   a. Question - this key takes as value the input question
   b. "answerSpanRequest": {"enable": true} – this enables the short answer feature

4. After you execute the call, observe the status returned, as shown in step 9. This should reflect 200 OK.

5. Observe the "answer" & "score" (step 10) and the "answerSpan" (step 11) and other values returned as part of the response.

Observer how the question we entered is quite different in verbiage than the question we trained out model on ("When Was Harry Potter Born"). Despite that, the model was able to deduce the intent and provide the correct answer.

Homework

1. Try adding follow-up prompts in the some of the answers while training the KB and test them out from the portal
2. Try creating a Bot after publishing the Knowledge Base, test it from Azure portal and publish it to any channel of your choice.

Additional recommended resources

Service Name	Category	Links
QnA Maker	Programming Language	C#, Go, JavaScript, Python, Ruby
	Tiers	Free, Basic, Standard (S1, S2, S3, S3HD)
	Pricing	https://azure.microsoft.com/en-us/pricing/details/cognitive-services/qna-maker/
	Limits	https://docs.microsoft.com/en-us/azure/cognitive-services/qnamaker/limits?tabs=v1#extraction-limits
	Language Support	https://docs.microsoft.com/en-us/azure/cognitive-services/qnamaker/overview/language-support?tabs=v1
	Sample Apps	Sample Application
	Regional Availability & Support	https://azure.microsoft.com/en-us/global-infrastructure/services/?products=cognitive-services&regions=all
	SLAs for Cognitive Services	https://azure.microsoft.com/en-in/support/legal/sla/cognitive-services/v1_1/
	Compliance & Certificates	https://azure.microsoft.com/en-us/support/legal/cognitive-services-compliance-and-privacy/
	Cognitive Services Updates	https://azure.microsoft.com/en-us/updates/?product=cognitive-services

Security best practices

1. Azure Cognitive Services security
2. Networking
3. Authentication
4. Key Management
5. Data loss prevention
6. Azure security baseline
7. Regulatory Compliance controls

Responsible AI being a part of best practices, we encourage you to read this.

QnA Maker Documentation

API & Error references