

AI Live 2018

Build a Microsoft Bot Framework bot with the Bot Builder SDK v4

Using Visual Studio 2017, and ASP.NET Core

Estimated time to complete: **30 Minutes**

This walkthrough will show you how you can configure your Bot to use QnA Maker. You'll need to have completed the previous lab.

Contents

What You Need

QnA Maker

Download the Starter Code

Configure QnA

What You Need

- The bot created in the previous labs working locally and published to Azure.
- Skype Desktop installed

QnA Maker

QnA Maker allows you to build, train and publish a simple question and answer bot based on FAQ URLs, structured documents, product manuals or editorial content in minutes. It allows you to go from FAQ to bot in minutes.

Download the Starter Code

Access the repo provided by your instructor.

1. Download the **qnamaker.zip** file.
2. Locate the ZIP file on your computer.
3. Right-click on it and select **Properties**.
4. Unblock the zip file.
5. Extract the contents to **C:\botlab**. You should have a folder **C:\botlab\qnamaker**. Continue to the next section.

Configure QnA

Next you'll create a QnA knowledge base.

1. In your web browser, navigate to <https://www.qnamaker.ai/>.
2. Click the **Sign In** link in the upper left hand .
3. Log in with the same account you're using in the Azure portal.
4. Click **Create a knowledge base**.
5. Click the **Create a QnA service** button in **Step 1**.
6. You'll be transferred to the Azure portal with the *Create QnA Maker* blade open.
7. Use the data in the following table to configure most of the information:

Item		Value	Notes
Name	labkb		A name for your knowledge base

Subscription	Your subscription	If necessary, select the Azure subscription you want to use if you have more than one.
Management pricing tier	F0	Select a pricing tier. You may need to change the location based on your subscription.
Location	West US	
Resource Group	Pick cs360bots .	Create a new resource group. It will be easier to clean up when you're done.
Search pricing tier	F	Click the View full pricing details link if you want to know more.
App name	A unique name. Consider qnabotabc99 where abc are your initials and 99 is a couple of numbers like the year you graduated, etc.	The unique URL name of the bot, no more than 35 characters in length.
Website Location	East US or East US 2	Select the geographic location for your resource group. Your location choice can be any location listed, though it's often best to choose a location closest to your customers. The location cannot be changed once the bot is created. For the lab at <i>Live! 360</i> , consider East US or East US 2 .
Application Insights	Off	

8. Once you've filled out the blade and checked your entries, click **Create**.

Wait for Azure to create your bot. Use the **Notifications** pane to monitor the status.

9. When your bot is ready, click the **Go to resource** button or navigate via the **Resource Groups** blade.
10. Leave the tab open and return to the QnA Maker site.
11. Refresh the page by clicking the link under **Step 2**.
12. Under **Step 2** fill out the settings until you can pick your QnA service.
13. Name your KB in **Step 3** to **qna**.
14. In **Step 4**, click Add file and locate the smartLightFAQ.tsv sample file in the CognitiveModels folder of your bot's source from the zip you extracted earlier.
15. It's up to you if you want to enable **Chit-chat**.
16. Click Create your KB under **Step 5**. Wait for it to finish.
17. Review the list of data and then try out the **Test** function.
18. When ready click **Publish**.
19. Click **Publish** again.
20. From the Postman box, save out the following values:
 - a. POST /knowledgebases/<your_knowledge_base_id>/generateAnswer
 - b. Host: https://<you_hostname>.azurewebsites.net/qnamaker
 - c. Authorization: EndpointKey <your_endpoint_key>

21. Return to Visual Studio and open the **qnamaker.bot** file.

22. Fill in the values in the JSON:

```
"type": "qna",  
"name": "QnABot",  
"KbId": "<YOUR_KNOWLEDGE_BASE_ID>",  
"Hostname": "https://<YOUR_HOSTNAME>.azurewebsites.net/qnamaker",  
"EndpointKey": "<YOUR_ENDPOINT_KEY>"
```

23. Save your changes.

24. **Build | Rebuild Solution** to build your bot.

25. Select **Debug | Start Debugging**.

26. Once your bot is running, start the **Bot Framework Emulator V4**.

27. Click the **Open Bot** button.

28. Navigate to your running bot's folder and open its **.bot** file.

29. Once connected, examine the code in **QnABot..cs** file and place a breakpoint or two in the **OnTurnAsync** method and run your bot.

30. Stop debugging when you've had enough.

Great job!