

# Introduction to Artificial Intelligence

## Lab 4 – Getting Started with Bots

### Overview

Bots are digital agents that users can interact with through chat sessions, social media platforms, email, or other channels. At its simplest, a bot receives and responds appropriately to messages from users. Bots can be very simple, with a limited set of well-defined interactions, or they can leverage AI capabilities, such as those provided by the Cognitive Services APIs, to respond intelligently to a comprehensive range of inputs.

In this lab, you will create simple Bots by using the Microsoft Bot Framework.

### What You'll Need

To complete this lab, you will need the following:

- A Microsoft account (for example, an *outlook.com*, *live.com*, or *hotmail.com* address)
- A Microsoft Azure subscription
- A Windows, Linux, or Mac OS X computer
- The lab files for this course

**Note:** To set up the required environment for the lab, follow the instructions in the [setup guide](#) for this course.

### Publishing a QnA Service

Bots are often used to answer frequently asked questions. You can train a QnA service with a knowledge base, and then use a Bot to enable users to ask questions that can be answered from the knowledge base.

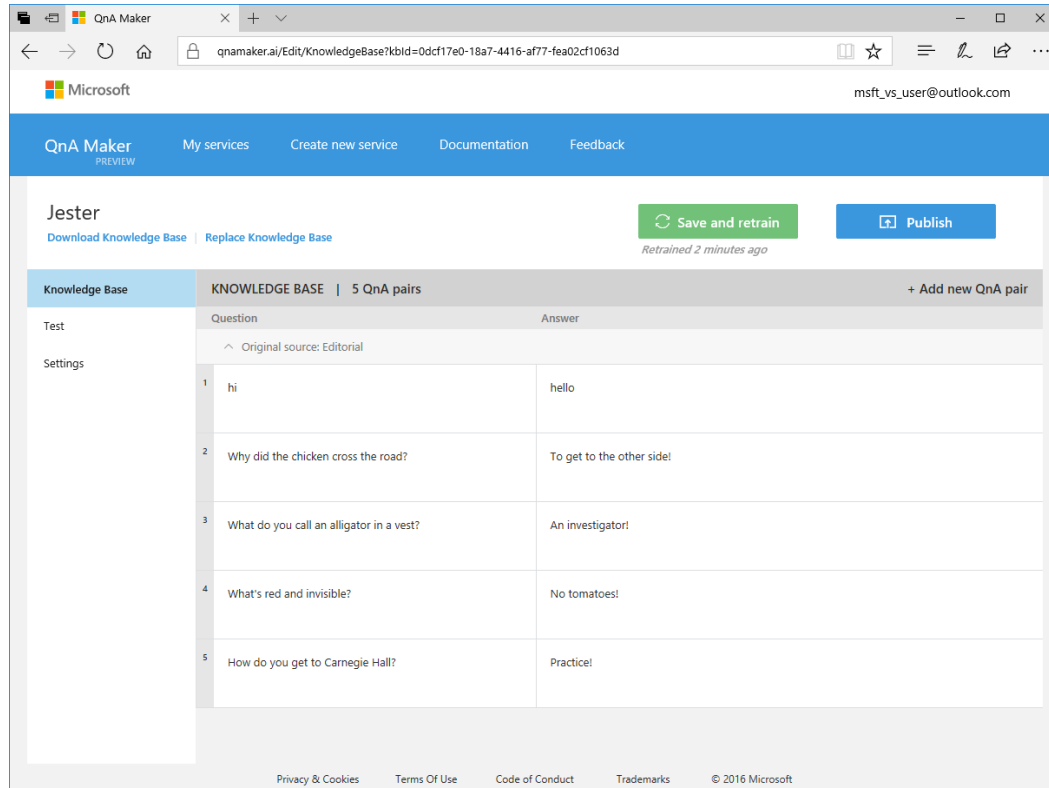
#### Create a QnA Service

1. Open a new browser tab, and navigate to <https://qnamaker.ai>.
2. Sign in using the Microsoft account associated with your Azure subscription. If this is the first time you have signed into the QnA Maker site, you may need to grant the app permission to access your info and agree to the terms and privacy statement (use the links to view these).
3. After you have signed in, click **Create new service**.
4. In the **Creating a QnA service** page, enter a service name of your choice. Then, without specifying a URL or uploading a file, click **Create**.
5. When your Q&A service has been created, the knowledge base consists of a single question and answer pair, in which the question *hi* corresponds to the answer *hello*.

- Click **Add new QnA pair** to add a row to the knowledge base table, and then in the new blank **Question** cell, enter *Why did the chicken cross the road?*, and in the **Answer** cell type *To get to the other side!*
- Repeat the previous step to add the following question and answer pairs:

Question	Answer
What do you call an alligator in a vest?	An investigator!
What's red and invisible?	No tomatoes!
How do you get to Carnegie Hall?	Practice!

- After you've added these question and answer pairs, ensure that your service looks like this:

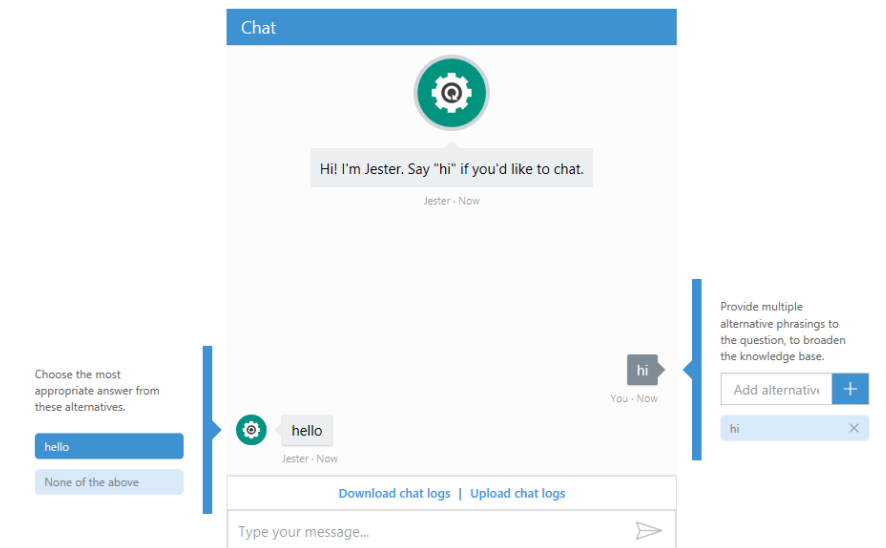


- Click **Save and retrain**.

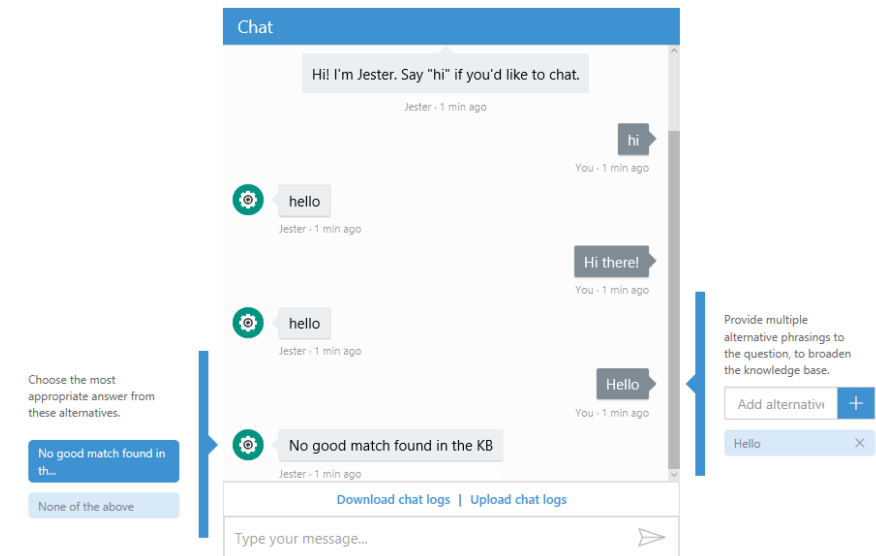
## Test the QnA Service

Now that you've defined a knowledge base for your QnA service, you can test it.

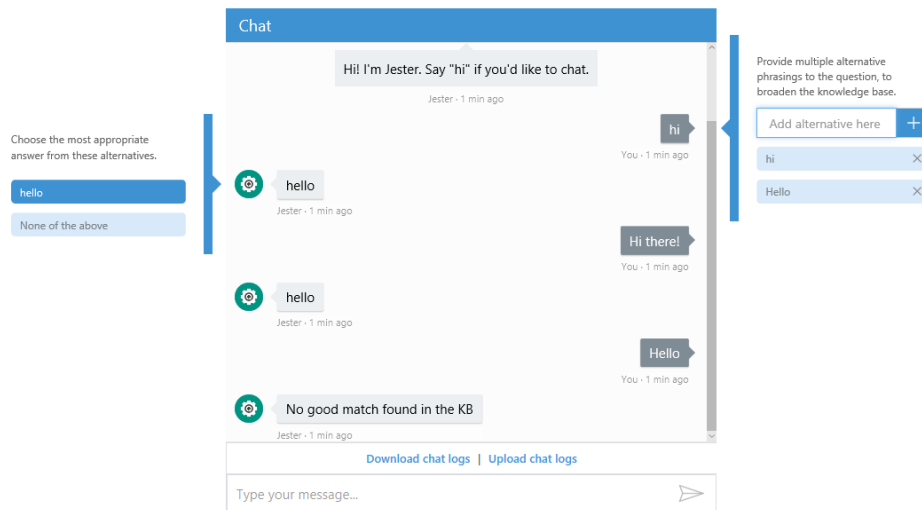
- In the QnA Maker page for your service, click **Test**. This displays a chat interface in which you can test responses to questions. The initial exchange of *hi* and *hello* has been completed as an example:



2. At the bottom of the chat window, enter the message *hi there!*, and note that the response *hello* is returned. *Hi there!* has been interpreted as *hi*.
3. Enter the message *Hello*, and notice that the message No good match found in the KB is returned. Clearly *Hello* has not been interpreted as *hi*:



4. On the right side of the chat window, next to the entry for *Hello*, note the option to provide multiple alternative phrasings to the question.
5. In the chat window, click the original *hi* message, and note that you can now add an alternative phrasing for this.
6. Add *Hello* as an alternative phrasing for *hi*:



7. Click **Save and retrain** to update the knowledge base.
8. After the knowledge base has been retrained, enter the message *Hello*, and note that the response *hello* is returned. *Hello* has now been interpreted as *hi*.
9. Enter the message *Why did the chicken cross the road?*, and verify that the answer *To get to the other side!* is returned.

## Publish the QnA Service

After you've refined the knowledge base for your service, you can publish it so that it can be consumed by a bot.

1. In the QnA Maker page for your service, click **Publish**
2. On the **Review your changes** page, click **Publish**.
3. After your service has been deployed, note the sample HTTP request that can be used to call your service from a bot.

## Creating a Bot

You can use your preferred development tool to build bots, and you can use the Bot Framework Emulator to test them on your local computer. Additionally, you can get started creating bots directly in the Azure portal by using the Azure Bot Service; which is the approach we'll take in this lab.

In this exercise, you'll create a simple bot that uses the QnA service you created in the previous exercise to answer user questions.

## Create a Bot Service

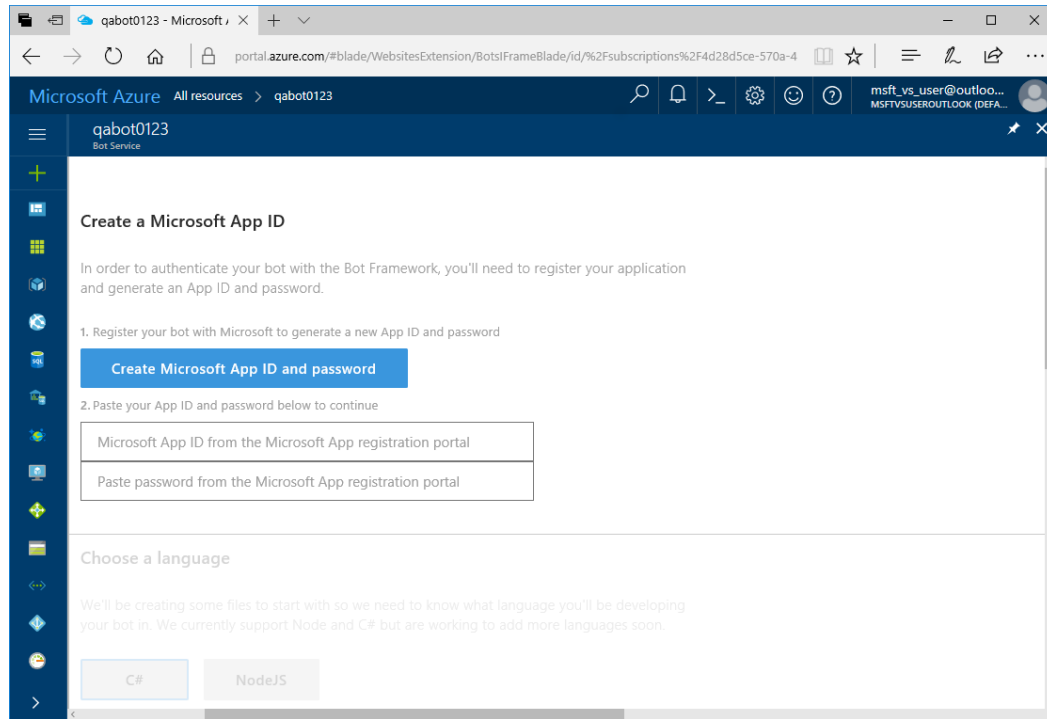
To start the development of a bot, you need to provision an Azure Bot Service in your Azure subscription.

1. In a web browser, navigate to <http://portal.azure.com>, and if prompted, sign in using the Microsoft account that is associated with your Azure subscription.
2. In the Microsoft Azure portal, click **New**. Then in the **Data + Analytics** menu, click **Bot Service**.
3. In the **Bot Service** blade, enter the following details and then click **Create**.
  - **App name:** Enter a unique name for your bot – this name will determine the fully qualified internet endpoint for your bot in the format `app_name.azurewebsites.net`.
  - **Subscription:** Choose your Azure subscription.

- **Resource Group:** Use the same resource group you used in the previous labs (or create a new one if you did not complete the previous labs).
  - **Location:** Choose region where you want to host your bot.
4. Wait for the bot to be deployed, and then in the **All resources** blade, click your bot.

You have created the service for your bot, but before you can implement it, you must authenticate it with the Bot Framework by assigning an app ID and password.

5. In the blade for your bot, click **Create Microsoft App ID and Password** as shown here:



6. When prompted, sign in using the Microsoft account associated with your Azure subscription.
7. In the Application Registration Portal, the App name for your bot and the App ID that has been generated for it are shown:

Microsoft Application Registration Portal Docs Feedback

## Generate App ID and password

App name  
qabot0123

App ID  
adf3684a-74cf-42d4-9304-03917cf4dae1

Generate an app password to continue

English Contact us Terms of use Privacy statement © Microsoft 2017

8. Click **Generate an app password to continue**. This generates a password in a dialog box as shown here:

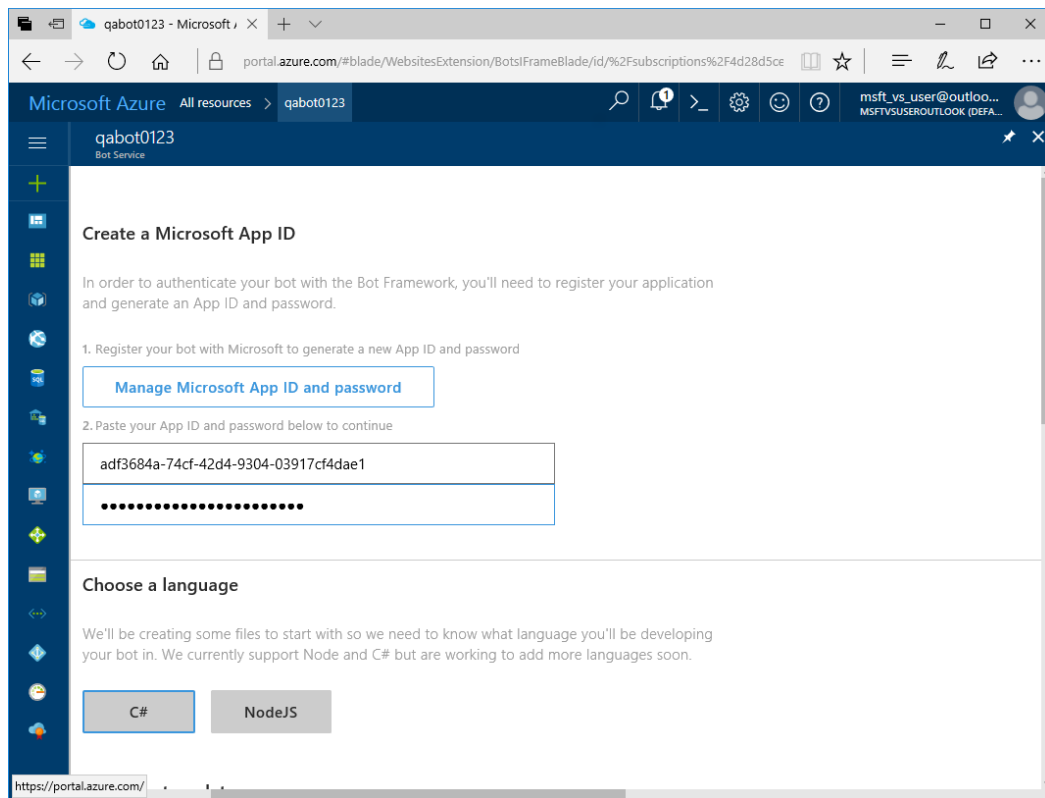
New password generated

Copy this password now, this is the only time when it will be displayed. Please store it securely and paste it into your bot configuration file.

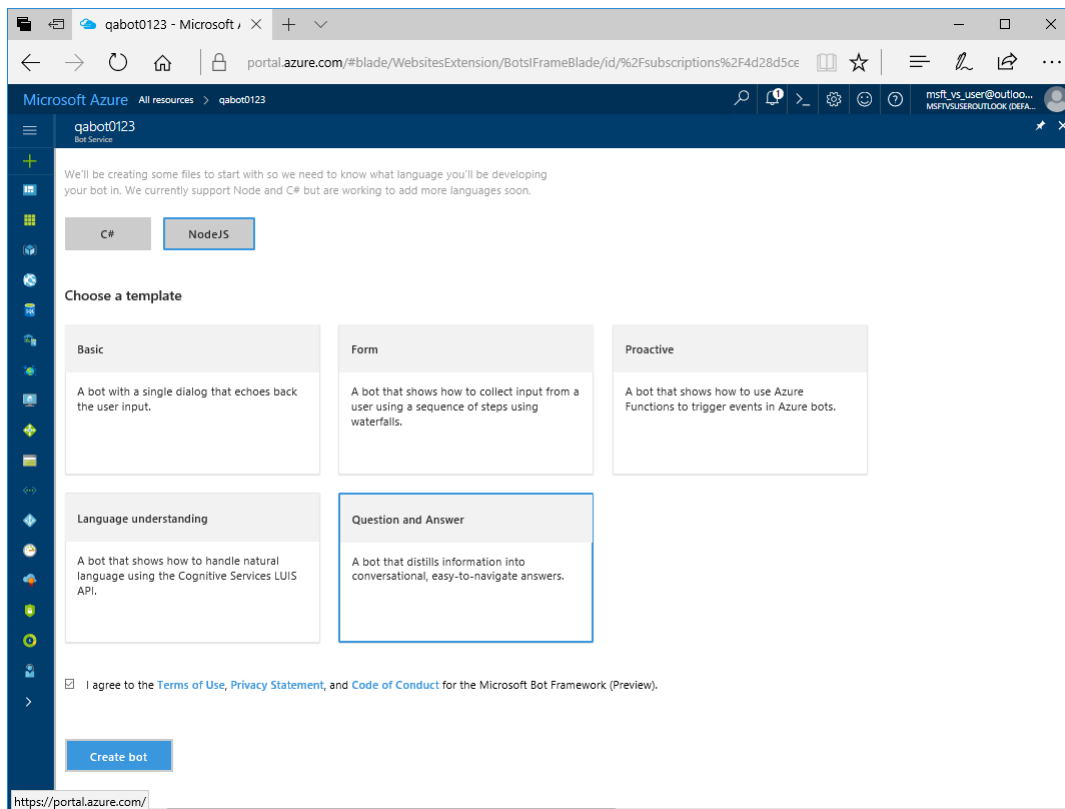
SXXLbXwqzZVZYRy1GoHgcYs

Ok

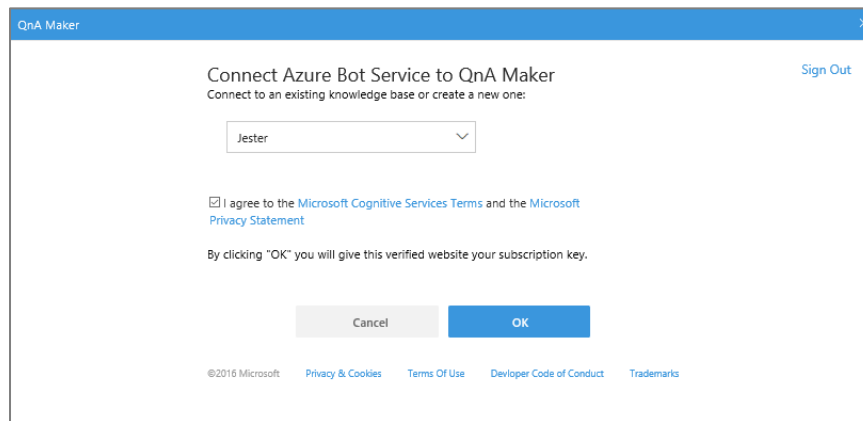
9. Copy the password to the clipboard – note that after closing this dialog box, you will not be able to retrieve the password. Only after you have copied the password, click **OK**.
10. In the Application Registration Portal, note that your password is obscured for security reasons:
11. Click **Finish and go back to Bot Framework**. This closes the Application Registration Portal.
12. Back in the Azure portal, in the blade for your bot, paste the password you copied under the App ID that has been filled in for you:



13. Scroll down if necessary, and select the language you prefer (we won't be doing any custom coding in this lab, but you may want to examine the code that is generated). Then select the **Question and Answer** template, agree to the terms of use, privacy statement, and code of conduct (use the links provided to review these), and click **Create bot**.



14. In the **Connect Azure Bot Service to QnA Maker** dialog box, select your QnA service, agree to the terms and privacy statement (use the links to review these), and click **OK**.



It may take a few minutes for your bot to be created and deployed.

## Explore Your Bot

Your bot is implemented as a project that contains multiple code files.

1. View the files generated for your bot – these will vary depending on the programming language you chose.
2. Note that you can edit the code files directly in the Azure portal, or you can configure *continuous integration* to synchronize your bot files with a code repository – common source code management systems such as Visual Studio Team Services and GitHub are supported as well as cloud storage locations such as OneDrive.
3. Click the **Channels** tab, and note the channels through which users can interact with your bot.
4. Click the **Analytics** tab, and note that metrics about bot usage are displayed here.
5. Click the **Settings** tab, and note the configuration settings with which you can manage your bot.

## Test the bot

The code that has been generated implements a working bot. You can test it to see it in action.

1. At the top of the blade for your bot, click **Test**. This opens a chat window.
2. At the bottom of the **Chat** pane, type *Hello*. After a short interval, the bot will be initialized, and the message *hello* will be displayed.
3. Enter the message *What do you call an alligator in a vest?*, and verify that the answer *An investigator!* is returned.
4. Enter the message *A chicken crossed the road. Why?*, and verify that the answer *To get to the other side!* is returned – despite the question not being exactly the same as the format in the knowledge base.
5. Click **Test** again to hide the chat window.
6. Close the blade for your bot.

## Summary

In this lab, you have created a simple bot that uses a QnA service to answer user questions. The Microsoft Bot Framework provides the application “scaffolding” you need to create bots that accept and respond to user input. You can use the Bot Framework to implement bots in .NET, Node.js, or any



application platform that can exchange JSON messages over HTTP through REST interfaces. The [Bot Builder SDK](#) provides libraries that you can use to implement bot functionality using the Bot Framework.

To learn more about developing bots – including more sophisticated bots that take advantage of cognitive services, see <https://dev.botframework.com>.