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 create a bot that use LUIS.

Contents

LUIS

Language Understanding (LUIS) is a machine learning-based service to build natural language into apps, bots, and IoT devices. It allows you to quickly create enterprise-ready, custom models that continuously improve and can make your bots more user friendly.

Create new Bot for LUIS

You'll now create a new bot to work with LUIS. The sample code will show you how to use many bot features including dialogs, LUIS, and more.

1. Return to the Azure portal.
2. Like before, click the **Create a resource** link in the upper left-hand portion of the portal home page.
3. In the list of items on the **Azure Marketplace** blade, click **AI + Machine Learning**.
4. Then, select **Web App Bot**.
   The Azure Portal will open a Web App Bot blade that you need to fill in with the requested information to create your bot.
5. Use the data in the following table to configure most of the information:

Item	Value	Notes
Bot name	Your bot's display name. Consider **luisbotabc99** where **abc** are your initials and **99** is a couple of numbers like the year you graduated, etc.	The display name for your bot that appears in channels and directories. Your bot's name can only have the following characters: **-**, **a-z**, **A-Z**, **0-9**, and **_**
Subscription	Your subscription	If necessary, select the Azure subscription you want to use if you have more than one.
Resource Group	Click **Create New** and provide a name like **cs360bots**.	Create a new resource group. It will be easier to clean up when you're done.
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 *Live! 360*, consider **East US** or **East US 2**.
Pricing tier	**F0**	Select a pricing tier. You may update the pricing tier at any time.
App name	A unique name. Consider **luisbotabc99** 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.
Bot template	**Basic Bot C#**	Choose **SDK v4** and **C#**
LUIS App Location	**West US**	Only US location to date.
App service plan/Location	Select your Bot App Service plan created earlier.	
Azure Storage	A new Azure storage account	Create a new data storage account.
Application Insights	**Off**	
Microsoft App ID and password	Auto create App ID and password	

6. 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.

7. When your bot is ready, click the **Go to resource** button or navigate via the **Resource Groups** blade.
8. Make sure your bot works. Click **Test in Web Chat** under the **Bot Management** heading.
9. Once the page loads, type **Help.**
   The bot will provide some help an adaptive card and some messages.
10. Now type **Hello**.
    The bot will now ask you a few questions. Follow along. Continue once the bot says it's happy to meet you.
    Now you'll download the source code so you can examine what's going on.
11. Back under **Bot Management**, click **Build**.
12. Click the **Download Bot source code** button. It will take a minute or two to create the zip file.
13. When it's ready, click the **Download bot Source code** that appears. Save the zip to your computer.
14. Locate the ZIP file on your computer.
15. Right-click on it and select **Properties**.
16. Unblock the zip file.
17. Extract the contents to **C:\botlab\luisbot\**.
18. Switch to Visual Studio 2017.
19. Open the downloaded solution. It will be called **BasicBot**.
20. Wait for the package restore process to complete.
21. Select **Build | Rebuild Solution**.
22. In the *Solution Explorer* window, double-click on the **Properties** node.
23. Change the **Target Framework** to **.NET Core 2.1**.
24. Save your work via **File | Save All**.

25. Select **Build | Rebuild Solution**.
26. In the *Solution Explorer*, right-click on the **Solution** node and choose **Manage NuGet Packages for Solution.**
27. In the *NuGet - Solution* window, if **Updates** is showing a number next to it, select the **Updates** link.
28. Click **Select all packages** followed by the **Update** button and complete the NuGet package update process.
29. Save your work via **File | Save All**.
30. Open **appsettings.json**.
31. Replace the long comment string *value* for **botfilePath** with the name of your **.bot** file.
32. To get the secrete you need to return to the Azure portal click the **All App service settings** command near the bottom of the menu of commands for your bot.
33. Click **Application Settings** from the menu on the left under **Settings**.
34. In the main pane, scroll down to find the Application setting section.
35. Copy the value from **botFileSecret** to the clipboard.
36. Return to Visual Studio.
37. Place the value in the **appsettings.json** file.
38. Save your changes. Keep the value on the clipboard.
39. Select **Debug | Start Debugging**.
40. Once your bot is running, start the **Bot Framework Emulator V4**.
41. Click the **Open Bot** button.
42. Navigate to your running bot's folder and open its **.bot** file.
43. When prompted, provide the secret for your bot file.
44. Once connected, examine the code in **BasicBot.cs** file and place a breakpoint or two in the **OnTurnAsync** method and run your bot.
45. Stop debugging when you've had enough.
46. In your web browser, connect up to https://www.luis.ai/home.
47. Click the **Login / Sign up** button.
48. Log in with the same account you're using in the Azure portal.
49. You'll notice you have an app with has a mangled name with your bot name and some extra digits.
50. Click it.
51. Examine the Intents and the rest of the app.
52. Return to the bot and in the **BasicBot.cs** file, change the message on line **189** (the message "Let me try to provide some help") to something else.
53. Build and test your changes in the debugger and emulator.
54. Next, in order to publish your changes back, you'll need to grab your Publish Profile from the App Settings Overview page. Return to the Azure portal.
55. It should still be on the Application settings page. Click the **Overview** menu item.
56. Click **Get publish profile** and save the file to your computer.
57. Open the saved file with Notepad.
58. Search for **userPWD**.
59. Copy the value inside quotes to the clipboard.
60. Return to Visual Studio.
61. Right-click on your project node and select Publish.
62. Click the Publish button.
63. When prompted for the password paste it in.
64. Once the publication is complete, test your bot from the Azure Portal's web chat.

Manually Configure LUIS

You can also manually create a LUIS App as well. Here's an example if you want to try.

1. In your web browser, connect up to https://www.luis.ai/home.
2. Click the **Login / Sign up** button.
3. Log in with the same account you're using in the Azure portal.
4. Next click **Create new app**
5. In the *Create new app* dialog, set the **Name** to **Support**.
6. Leave the **Culture** as **English**.
7. Optionally enter a **Description** like **App to offer support and help** and click **Done**.
   Wait a few seconds for it to be created and the left menu pane to open.
8. If necessary, select **Intents** from the left-hand menu pane.
9. Next select **Add prebuilt domain intent**.
10. In the *Add prebuild domain intents* dialog, type Help in the *Search* field.
11. Select **Utilities.Help** and then click **Done**.
12. In the upper right, click the **Train** button.
    Wait for the training to complete.
13. Click the **Test** button.
14. In the *Test* panel type **I need help with my computer** and press Enter.
15. Try a few more phrases noting the score.
16. When ready, you could publish your app and use it from a bot.