Tutorial: Improve intent recognition with Language Understanding

10/13/2021 • 5 minutes to read •

Is this page helpful?

In this article

APPLIES TO: Composer v2.x

The **LUIS recognizer** is one of the recognizers available in Composer. It uses the Language Understanding (LUIS) service, which allows the bot understand the user's response and determine what to do next in a conversation flow.

Language Understanding (LUIS) is a service within Azure Cognitive Services that applies natural language processing to conversational text to predict meaning and extract relevant information.

This tutorial shows how to train the LUIS recognizer to capture the user's **intent** contained in a message. Then how to pass the information to the triggers, which defines how the bot will respond.

In this tutorial, you learn how to:

- ✓ Add the LUIS recognizer to your bot.
- ✓ Determine user intent and entities to generate helpful responses.

Prerequisites

- Complete the Create a weather bot tutorial
- Knowledge about Language Understanding concept article

- A LUIS account. If you don't have a LUIS account, you can sign up for one on the LUIS site.

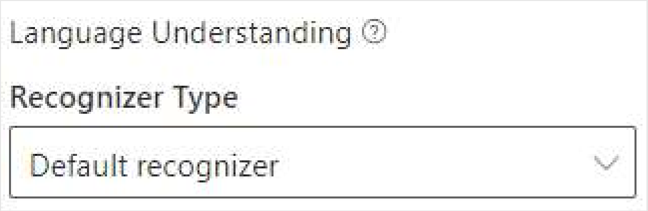
Update the recognizer type

1. Start Composer.

2. Select the **weather_bot** bot project from the **Recent** bot list on the homepage.

3. Select the main dialog **weather_bot** in the bot explorer.

4. In the properties pane, select **Default recognizer** from the **Recognizer Type** drop-down. The **Default recognizer** uses LUIS.



Language Understanding

Recognizer Type

Default recognizer

Add language understanding data and conditions

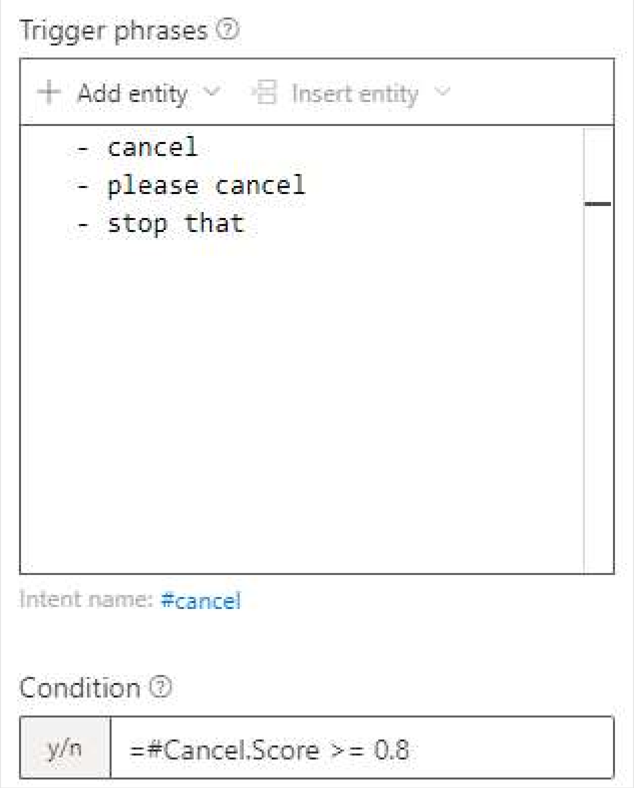
In this section, you will learn how to create three **Intent recognized** triggers using LUIS recognizer. You can ignore or delete the **Intent recognized** triggers you created using regular expressions in the Create a weather bot tutorial tutorial.

1. Select the **cancel** trigger you created using regular expressions. In the properties pane on the right, and enter the following **Trigger phrases**:

Language understanding Copy

```
- cancel
- please cancel
- stop that
```

2. In the **Condition** property from the drop-down list, select **Write an expression**. An = sign will appear. After the equal sign, enter *#Cancel.Score >= 0.8*.



```
- cancel
- please cancel
- stop that
```

=#Cancel.Score >= 0.8

This tells the bot not to fire the cancel trigger if the confidence score returned by LUIS is lower than 80%. LUIS is a machine learning based intent classifier and can return a variety of possible matches, so you will want to avoid low confidence results.

3. Now select the **help** trigger you created using regular expressions. In the properties pane on the right, and enter the following **Trigger phrases**:

Language understanding

Copy

```
- help
- I need help
- please help me
- can you help
```

4. In the **Condition** property from the drop-down list, select **Write an expression**. An = sign will appear. After the equal sign, enter *#Help.Score >= 0.5*.

5. Select the **weather** trigger. In the properties pane, enter the following **Trigger phrases**:

Language understanding

Copy

```
- get weather
- weather
- how is the weather
```

Configure a Language Understanding resource

You will need to create a Language Understanding (LUIS) resource in Azure. This resource will be used for authoring your language models. The following steps will guide you through creating a Language Understanding resource in Azure or sending a request to your Azure administrator to create a Language Understanding resource on your behalf.

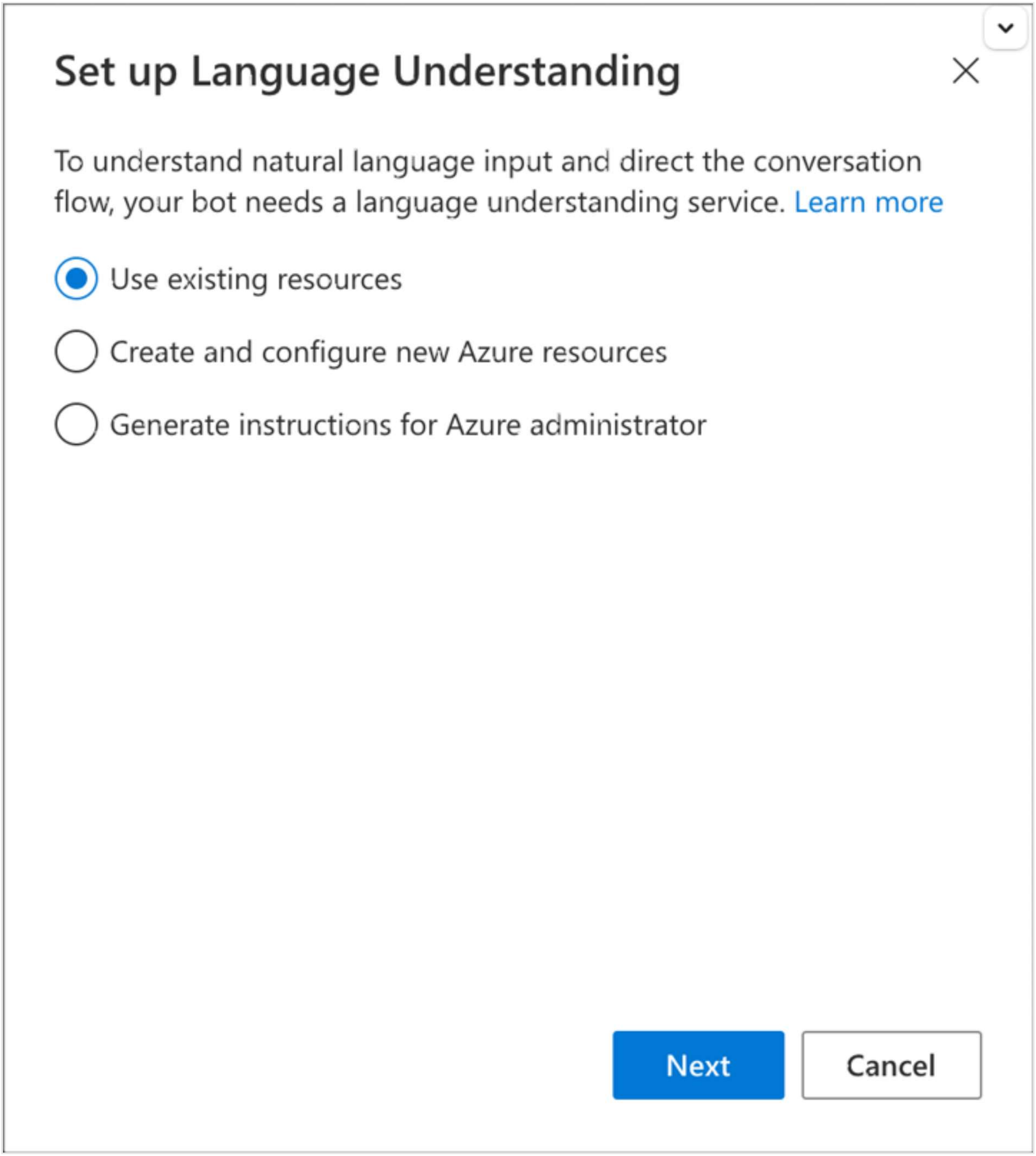
1. You will notice an error icon next to your **weather_bot** bot in the bot explorer. Select the icon, and then select **Fix in bot settings**.

   This will navigate you to the **Development resources** tab of the Configure section. Alternatively, you can select the wrench icon in the main menu and then select the **Development resources** tab.

2. You will see the field labeled **Language Understanding authoring key** field is highlighted with an error message.

> ⓘ **Note**
>
> The following steps will guide you through creating a new LUIS resource. If already you have a LUIS authoring resource and know the key, you can enter the key directly in this field and skip to the next section.

1. Select the button **Set up Language Understanding**. This window will appear:



Set up Language Understanding

To understand natural language input and direct the conversation flow, your bot needs a language understanding service. Learn more

● Use existing resources

○ Create and configure new Azure resources

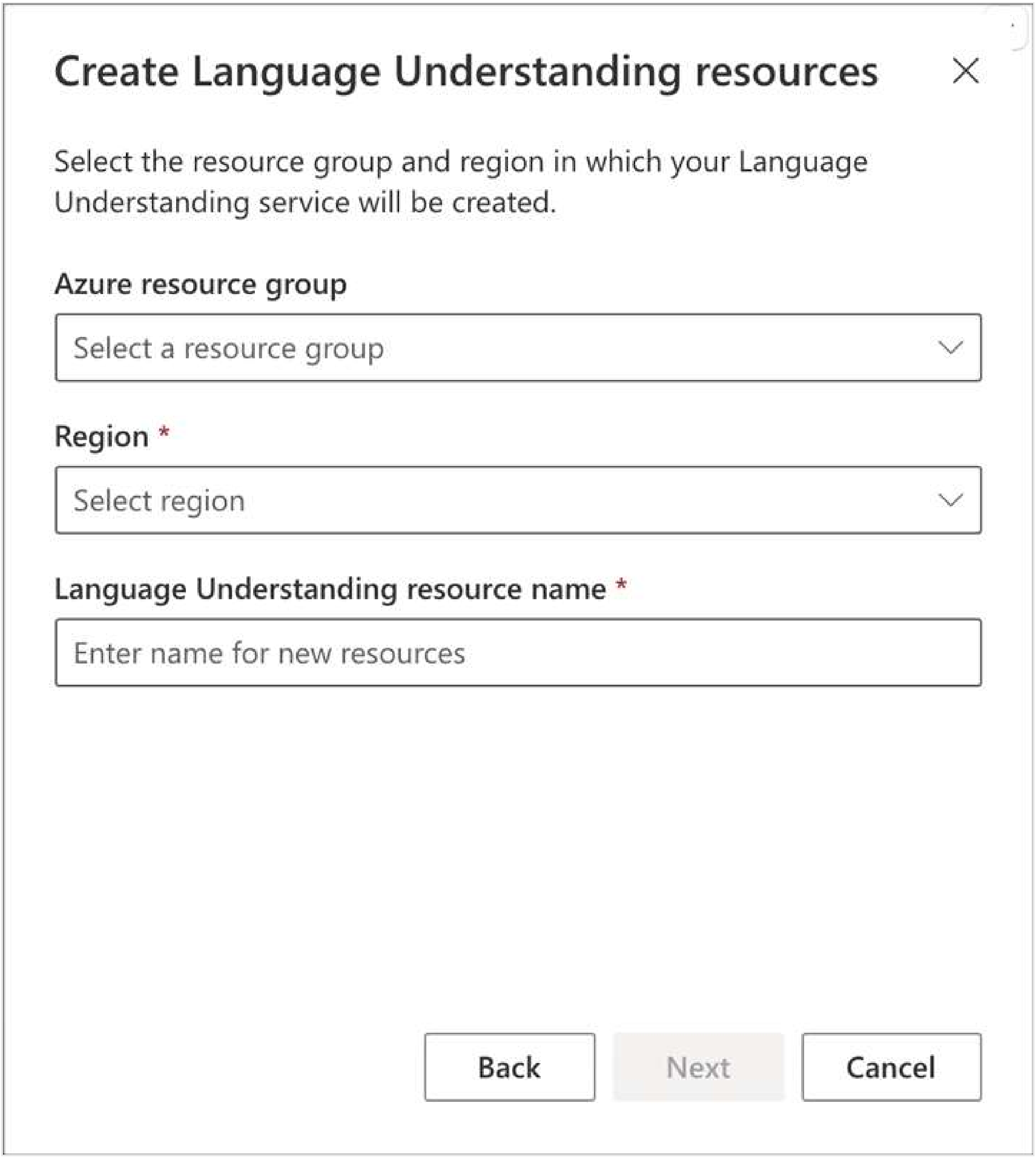
○ Generate instructions for Azure administrator

2. If you have an Azure account and have permissions to create new resources, select **Create and configure new Azure resources** and select **Next**.

> ⓘ Note
>
> If you need an Azure administrator to create Azure resources for you, select **Generate instructions for Azure administrator** to create a request you can send to your Azure administrator. You won't be able to continue without a LUIS key.

3. Follow the prompts for signing in to Azure.

4. Once you have signed in to Azure, you will land on the "Create Language Understanding resources" window:



Create Language Understanding resources

Select the resource group and region in which your Language Understanding service will be created.

Azure resource group

Select a resource group

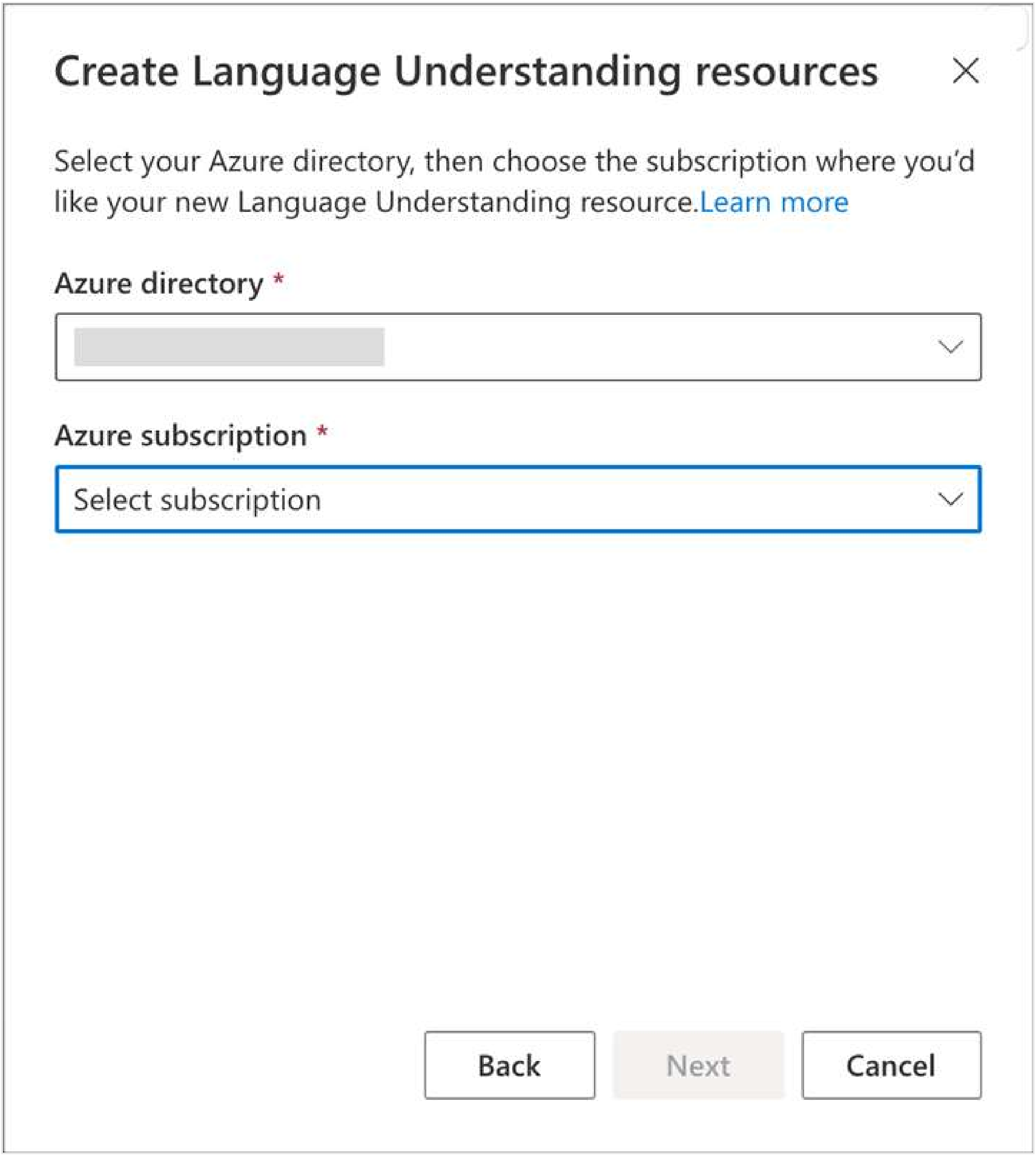
Region *

Select region

Language Understanding resource name *

Enter name for new resources

5. Select your subscription and then **Next**. The window will appear allowing you to name your resource group, choose a region, and name your resource:



Create Language Understanding resources

Select your Azure directory, then choose the subscription where you'd like your new Language Understanding resource.Learn more

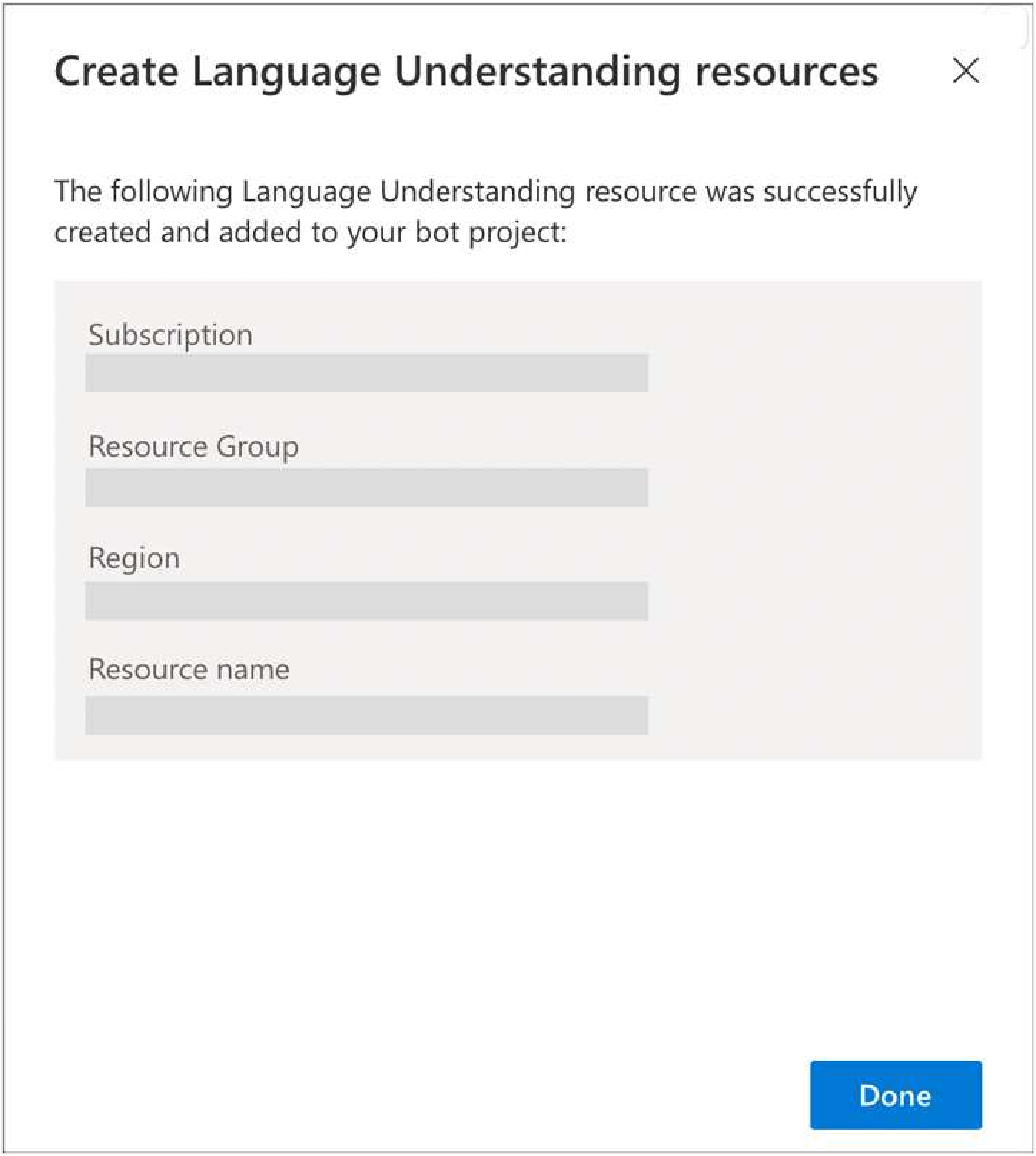
Azure directory *

Azure subscription *

Select subscription

6. Enter a name for your Azure resource group, select a region, and then name your Language Understanding resource. Select **Next**.

7. A window will appear, confirming the resource you created:



Create Language Understanding resources

The following Language Understanding resource was successfully created and added to your bot project:

Subscription

Resource Group

Region

Resource name

8. Select **Done**.

You will notice that the key for your Language Understanding resource has been copied into the field. You're now ready to test your bot!

1. Select **Start bot** on the top right, and then select **Open Web Chat**.

   With LUIS, you no longer have to type in exact regex patterns to trigger specific scenarios for your bot. Try phrases like:

   - "How is the weather"
   - "Weather please"

- "Cancel for me"
- "Can you help me?"

Using LUIS for entity extraction

You can use LUIS to recognize entities. An entity is a word or phrase extracted from the user's utterance that helps clarify their intent.

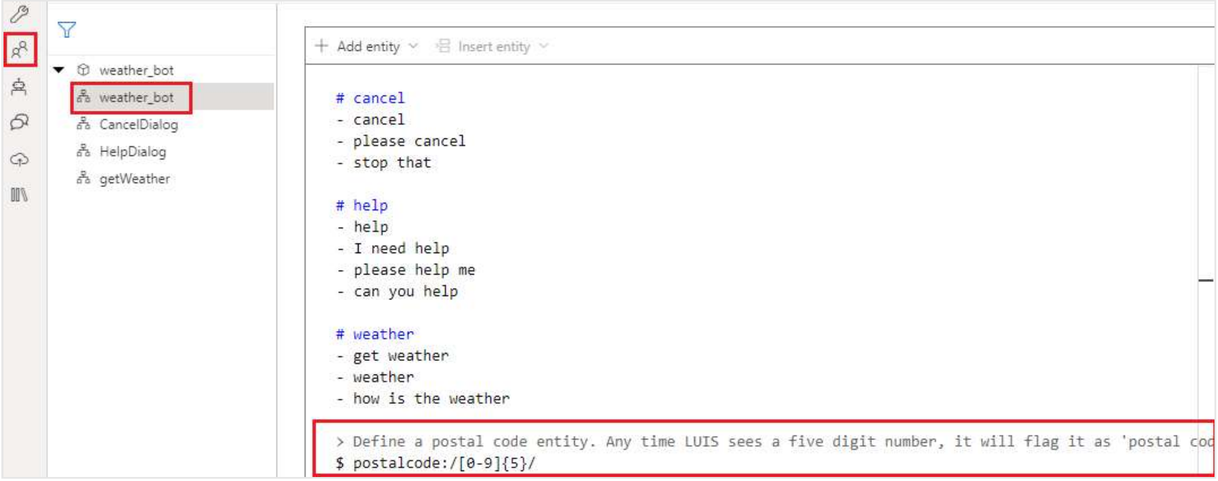
For example, the user could say "How is the weather in 98052?" Instead of prompting the user again for a zip code, your bot could respond with the weather. This is a very simple example of very powerful capabilities. For more information on how to use LUIS for entity extraction in Composer, read the How to define intent with entities article.

The first step is to add a regex entity extraction rule to the LUIS app.

1. Select **User Input** and then select the **weather_bot** dialog in the bot explorer. (Be sure to select the **weather_bot** dialog and not the **weather_bot** root bot.) Select **Show code** and in the **Language Understanding** editor add the following entity definition at the end of the LU content:

Language understanding Copy

```
> Define a postal code entity. Any time LUIS sees a five digit number, it
will flag it as 'postal code' entity.
$ postalcode:/[0-9]{5}/
```



```
# cancel
- cancel
- please cancel
- stop that

# help
- help
- I need help
- please help me
- can you help

# weather
- get weather
- weather
- how is the weather

> Define a postal code entity. Any time LUIS sees a five digit number, it will flag it as 'postal cod
$ postalcode:/[0-9]{5}/
```

The next step is to create an action in the **BeginDialog** trigger to set the `user.postalcode` property to the value of the `postalcode` entity.

2. Select the **getWeather** dialog in the bot explorer, then the **BeginDialog** trigger.

3. Select **+** in the authoring canvas to insert an action after the **Send a response** action (that has the prompt *Let's check the weather*). Then select **Set a property** from the **Manage Properties** menu.

4. In the properties pane, enter *user.postalcode* into the **Property** field and *=@postalcode* in the **Value** field. The `user.postalcode` property will now be set to the value of the `postalcode` entity. If the user enters a postal code as part of the message, they will no longer be prompted for it.

Your bot is now ready to test.

Test the bot

1. Select the **Restart Bot** button on the top right and wait for the LUIS application to finish updating your changes. Then select **Open Web Chat**.

   Now when you say "how is the weather in 98052", the bot will respond with the weather for that location instead of prompting you for a postal code.

Congratulations, you have completed tutorial and created a weather bot!

Recommended content

Skills and reuse of conversation logic between bots - Bot Framework Composer

Learn about extending bots, using other bots as skills in Bot Framework Composer.

How to create and test a local skill in Bot Framework Composer

Learn how to create a root bot and local skill in the same Composer session, and test their interaction locally, in the same Composer session.

Connect to a remote skill with Bot Framework Composer

Learn how to extend a user-facing bot by calling conversational logic provided by a separate skill bot.

Templates available in Bot Framework Composer

Composer provides a set of templates you can use to build your own conversational experience. The templates provide common conversational capabilities across a range of scenarios

Show more