Working with Entities

Defining the Dialog

✓ **Video:** Putting it All Together
5 min

✓ **Video:** Building User-Friendly Chatbots
6 min

Reading: Lab 1: Implement the Dialog
45 min

Reading: Lab 2: Define Domain-Specific Intents
1h

Quiz: Module 4 Quiz: Dialog
5 questions

Reading: What's Next
1 min

With intents and entities under our belts, we can finally look at the third component: the dialog.

In fact, at this point, our chatbot can understand some intents and detect a few specific pieces of information thanks to entities. What we are missing is using this information to formulate appropriate responses to the user. Will do so in this module to create a simple, but useful chatbot. In this lab, we'll start by defining chit chat responses.

Exercise 1: Create a Dialog and improve the prompt

Let's kick things off by creating a dialog and a good prompt for our chatbot.

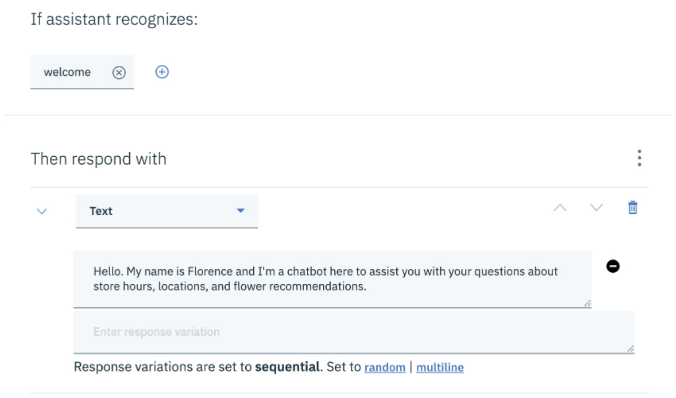
1. **Click on the *Dialog* section** of your skill.

2. It's empty at the moment. Go ahead and click the ***Create* button**.

3. Take a moment to **investigate the default *Welcome* and *Anything else* nodes** that were generated, by clicking on them.

4. **Open the *Try it out* panel** and click on the Clear link at the top to start testing the chatbot from scratch. Notice anything different this time around?

5. Yes, we have a prompt! Unfortunately, the default phrasing is not very user-friendly. **Let's change it.** Select the Welcome node and edit the response to say Hello. My name is Florence and I'm a chatbot here to assist you with your questions about store hours, locations, and flower recommendations. Change the name from Florence, to whatever flower-inspired name you prefer, to make it yours.


If assistant recognizes:

welcome

Then respond with

Text

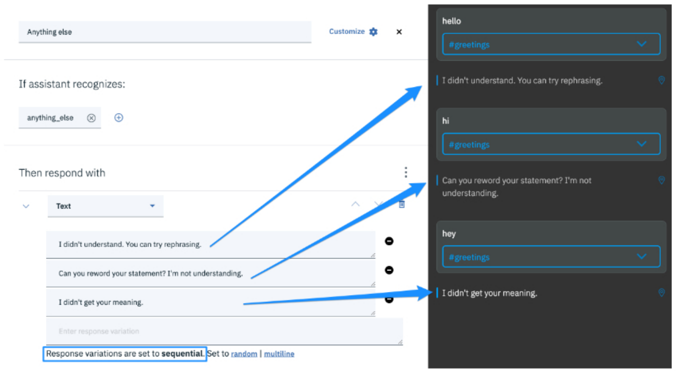
Hello. My name is Florence and I'm a chatbot here to assist you with your questions about store hours, locations, and flower recommendations.

Enter response variation

Response variations are set to **sequential.** Set to random | multiline

7. Now **try replying hello** in the Try it out panel. What happens? Watson recognized the right intent (i.e., #greetings) but doesn't have a node to handle greetings, so the fallback node Anything else was executed. We'll remedy this in the next exercise.

It's worth noting that if you enter a greeting (or anything at this point) multiple times, you'll get a different response each time. The reason for this is that the Anything else node has three response variations by default. Furthermore, these are set in sequential mode. So every time we hit this node, the next response variation is provided to us, as shown below.


Anything else

If assistant recognizes:

anything_else

Then respond with

Text

I didn't understand. You can try rephrasing.

Can you reword your statement? I'm not understanding.

I didn't get your meaning.

Enter response variation

Response variations are set to **sequential.** Set to random | multiline

hello
#greetings
I didn't understand. You can try rephrasing.

hi
#greetings
Can you reword your statement? I'm not understanding.

hey
#greetings
I didn't get your meaning.

If we didn't care about the specific order, we could set this to random and a random variation would be provided each time.

The reason why we want variation is that we don't want to robotically say to the user, *I don't understand* every time the chatbot fails to handle the user input with an appropriate node. It gets old fast and makes our chatbot come across as dumber that it could be.

For nodes that are unlikely to be hit multiple times within a conversation, it's okay to have a single response with no variations. In every other case, variations are good to have.

You might wonder whether you should prefer sequential or random for your variations. Sequential works well when you plan to leverage your knowledge that the node was hit multiple times to provide a better response to the user. Random when the variation that is given to the user, doesn't matter.

Go ahead and **add a fourth variation** to the *Anything else* node, with the following text: It looks like we are not quite getting each other today. Would you like to talk to a human agent instead? If so, please contact us at 555-123-4567 or email us at support\@example.org. The \ before the @ is needed to display the special character (something programmers call, "escaping").

Because we have this set in sequential mode, we ensure that this escalation of the sort is only given as an option after we failed to understand the user four times in the same conversation. If this was set to random, we'd risk escalating the very first time we don't understand the user, which is typically not what we want.

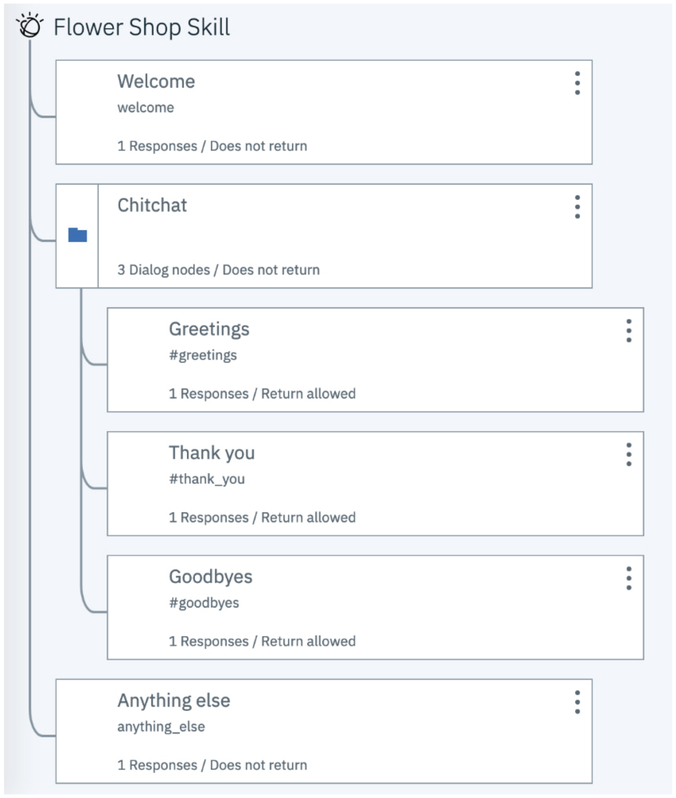
Exercise 2: Add Chit Chat nodes

We have three chit chat intents, #greetings, #thank_you, and #goodbyes. We now need to have nodes that specify what response we want to give the user when such intents are detected.

We have a couple of strategies possible here. We could create three nodes, one for each of these intents. This is the most common and simple approach. The other option would be to create a single node for chit chat that uses multiple responses (essentially, conditional responses), attaching a condition to each response.

I would recommend that you stick to the traditional way as it's more flexible. It allows us to add more chit chat nodes down the line, as well as making the chit chat logic more complex if needed.

We want to keep things organized, separating small talk from domain-specific nodes. So we'll create a folder for chit chat, and we'll create three nodes in it for now. The picture below shows the end result. At any time, you'll be able to collapse or expand the folder by clicking on the blue folder icon.



Flower Shop Skill

Welcome
welcome
1 Responses / Does not return

Chitchat
3 Dialog nodes / Does not return

Greetings
#greetings
1 Responses / Return allowed

Thank you
#thank_you
1 Responses / Return allowed

Goodbyes
#goodbyes
1 Responses / Return allowed

Anything else
anything_else
1 Responses / Does not return

Follow these steps to add it to your chatbot:

1. Select the Welcome node and then **click the *Add folder* button.** This will create a folder underneath the selected node (i.e., Welcome).

2. **Name the folder Chitchat.** You don't need to specify a condition for the folder, as the conditions of the children nodes will suffice.

3. With the Chitchat folder selected, **click the *Add child* node button**. This will create an empty child node within the folder.

4. **Name this node Greetings.** We want it to be executed when the #greetings intent is detected, so under If bot recognizes: **enter the #greetings intent.** Autocomplete will help you find the intent (not that useful here, but quite handy in complex chatbots with many intents).

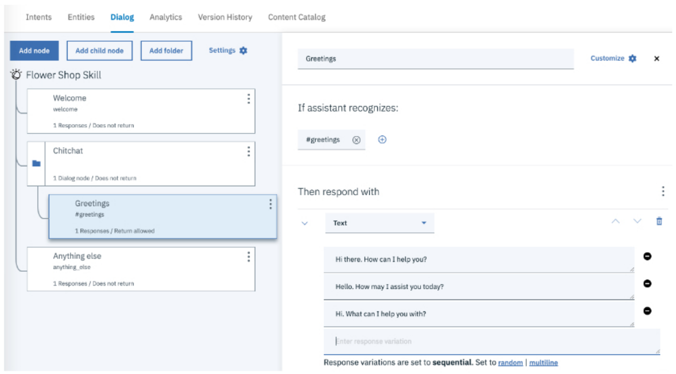
It's worth noting that you can make the condition of a node as complex or as simple as you'd like. You can use ||(or its alias OR) and && (or its alias AND) to make the condition more complex. We don't want this here, but if you wanted to execute a node if the intent detected was either #greetings or #goodbyes we could simply type #greetings OR #goodbyes in the node condition.

5. **Enter a few appropriate responses**. The scenario we are handling here is one in which we already greeted the user with our prompt, and they replied with a greeting. So we should greet them back without repeating the prompt verbatim.

Enter a few responses to offer some variation if we get a greeting-happy user. Examples could be Hi there. How can I help you?, Hello. How may I assist you today?, Hi. What can I help you with?

Normally, I would advise against open-ended questions such as how can I help you, but since we already qualified the scope of the chatbot in our prompt, we can get away with it here.

6. You can leave the response variations set to sequential or set them to random if you prefer. The third option, multiline is not applicable here, as it would provide a response over multiple lines using each response you wrote as its own line, de facto asking the user what is essentially the same question three times at once. 😄 This is what the node will look like.



Flower Shop Skill

Welcome
welcome
1 Responses / Does not return

Chitchat
1 Dialog node / Does not return

Greetings
#greetings
1 Responses / Return allowed

Anything else
anything_else
1 Responses / Does not return

Greetings

If assistant recognizes:

#greetings

Then respond with

Text

Hi there. How can I help you?

Hello. How may I assist you today?

Hi. What can I help you with?

Enter response variation

Response variations are set to **sequential.** Set to random | multiline

7. The *And finally* section at the bottom of the node defines what happens after this node has been executed and a response was given to the user. In the case of this node, after we responded to the user, we expect them to enter some more questions, so you can also **leave *Wait for user input***as the final action for this node.

8. Open the *Try it out* panel (if you closed it) and click the Clear link to start a new conversation. **Try to reply hi to the chatbot prompt**. Congratulations, you just had your first conversation with our chatbot. It's not a complex interaction, but it's a start. You can now close this panel.

9. With the Greetings node selected, **click on the *Add node* button**. This will create an empty peer node below Greetings

The order of these chit chat nodes is not that important because they are all simple nodes with independent intents. However, the order can matter in more complex scenarios (as we'll see in a moment) and it makes sense to place them in a logical manner that is roughly equivalent to how a conversation would go. Greetings first, thank yous in the middle, and goodbyes at the end.

Go ahead and make this node **handle the #thank_you intent**. For the responses, you'll likely want something like:

You're welcome. Please let me know if you need anything else.

My pleasure.

No problem. Let me know if there is anything else I can help with.

You could get chicky, and add:

I aim to please. 😊

Depending, of course, on how much personality you'd like to inject in your chatbot. BTW, yes, emojis are supported. **Set the response variation to Random** by clicking on the *Random* link.

10. With the *Thank you* node selected, **add a Goodbyes node** which will handle the #goodbyes intent. You can use standard polite goodbye responses such as:

Nice talking to you today.

Have a nice day.

Goodbye.

11. Start a new conversation in the *Try it out* panel and **test all three intents** to ensure you get an appropriate response in each case.

Mark as completed