Lab 6: Implement the Dialog

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. We'll 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 start by investigating the dialog and adding a good prompt for our chatbot.

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

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

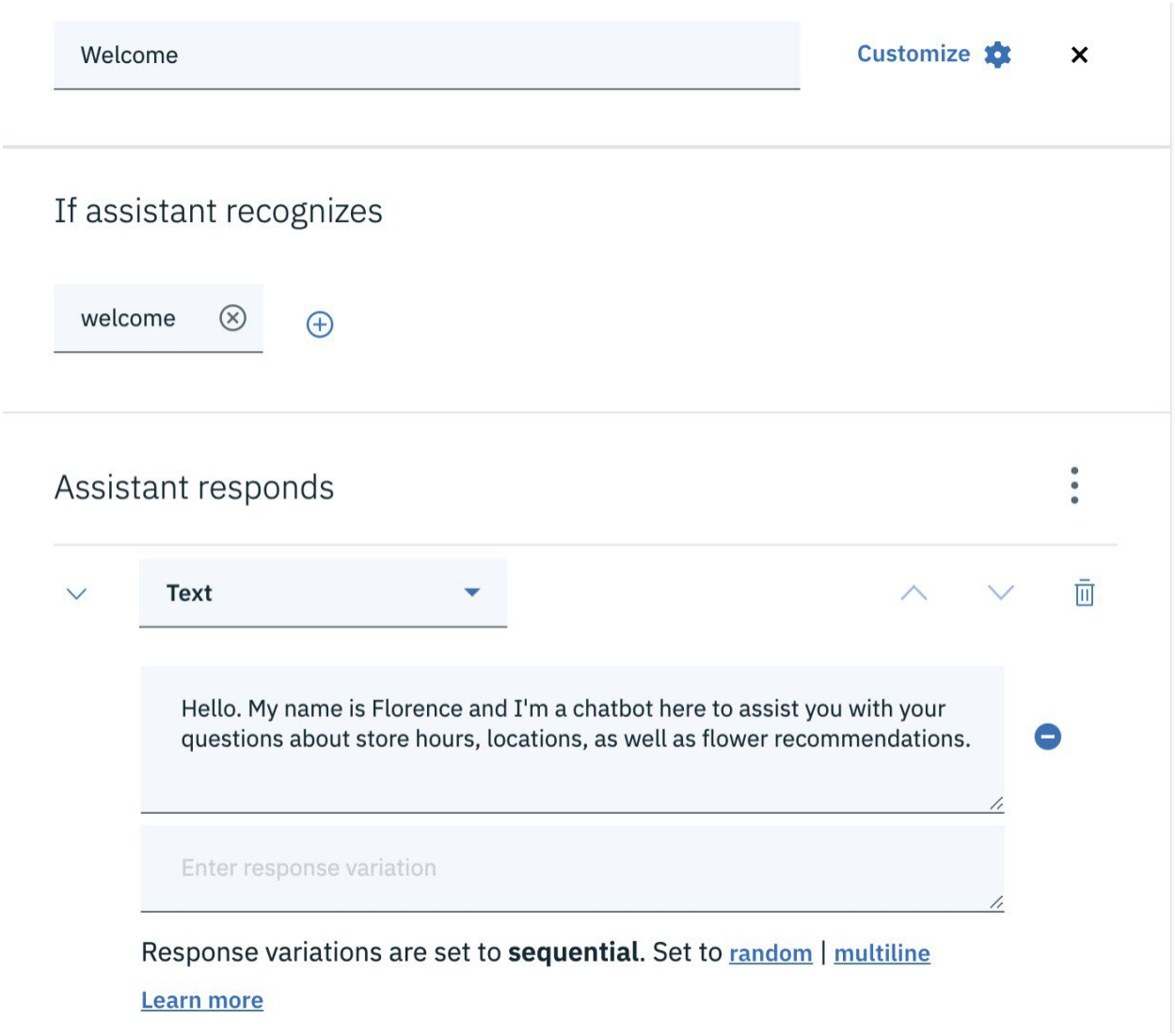
3. **Open the *Try it out* panel** and click on the *Clear* link at the top to start testing the chatbot from scratch.

The default prompt, "Hello, How Can I help you?" is actually not very user-friendly. **Let's change it.**

Close the *Try it out* panel, and then select the *Welcome* node. Here, you'll want to 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, as well as flower recommendations.
```

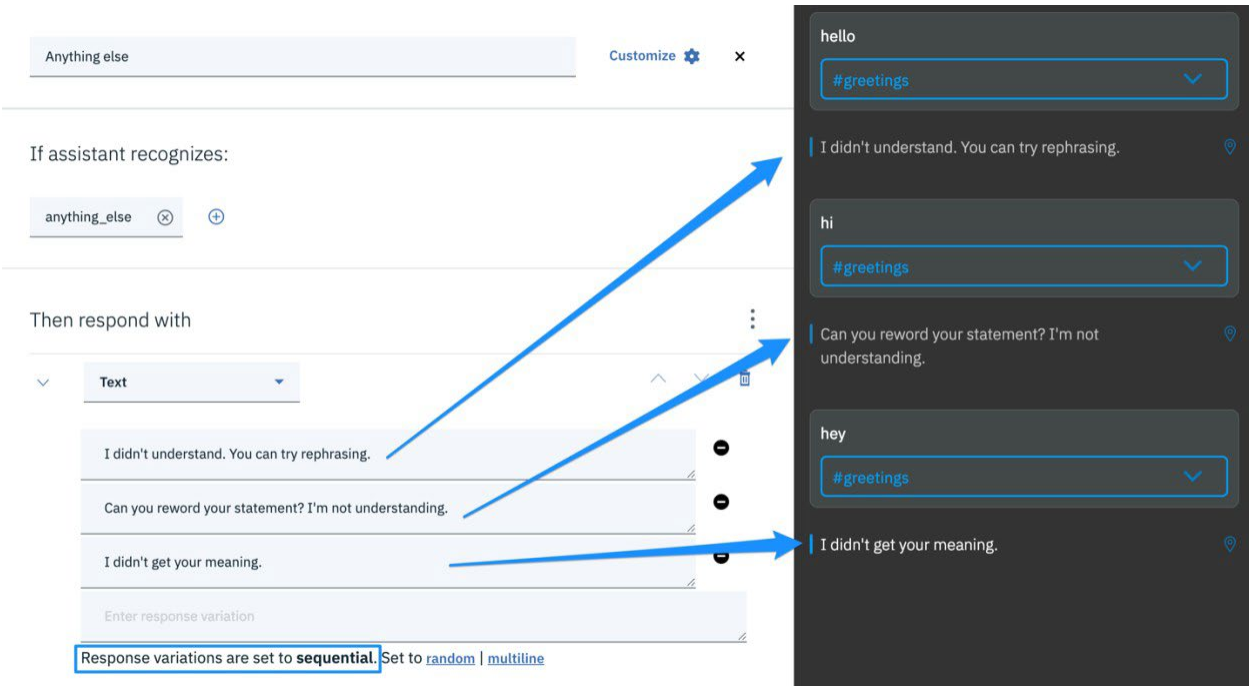
Change the name from Florence to whatever flower-inspired name you prefer to make it yours.



Open the *Try it out* panel and click the *Clear* link at the top once again. This time, you should see the much more informative prompt we specified.

4. Good. Now **try replying `hello`** in the *Try it out* panel. What happens? Watson recognized the right intent (i.e., `#greetings`) but didn'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. This is because 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 following response variation is provided to us, as shown below.



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

We want variation because 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 not as smart as 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. Variations are good in every other case.

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 that programmers call, "escaping").

Because we have this node's responses 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.

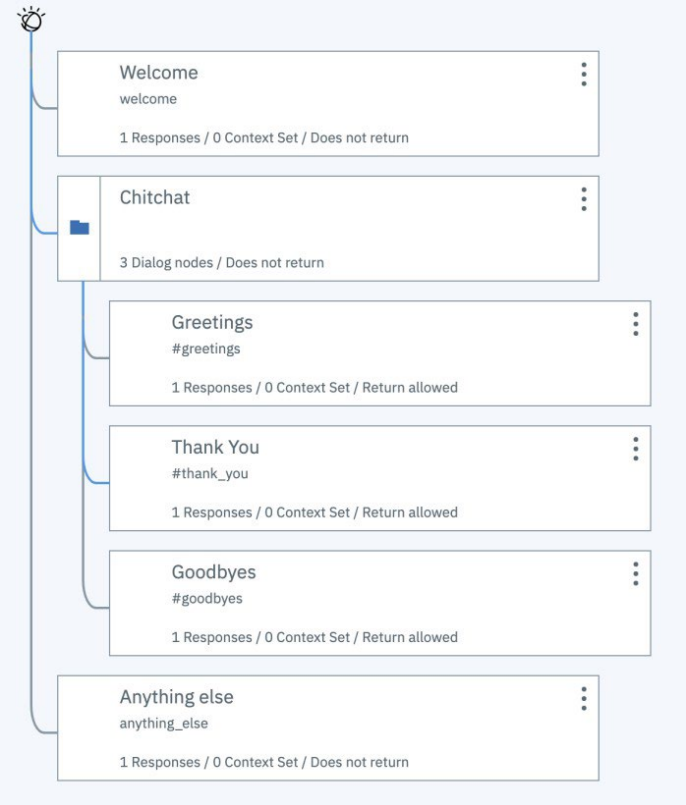
Exercise 2: Add Chit Chat nodes

As you know, 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 that are 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 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 do still 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 inside for now. The picture below shows the structure of the lab's end result.

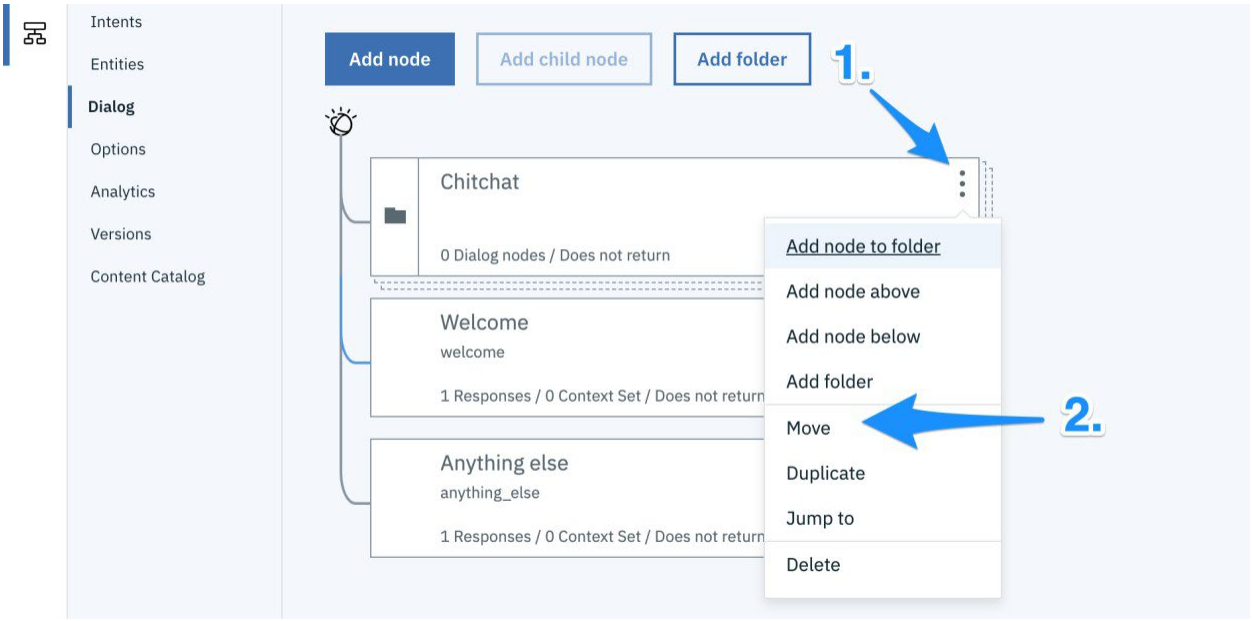


At any time, you'll be able to collapse or expand the folder by clicking on the folder icon next in the *Chitchat* folder.

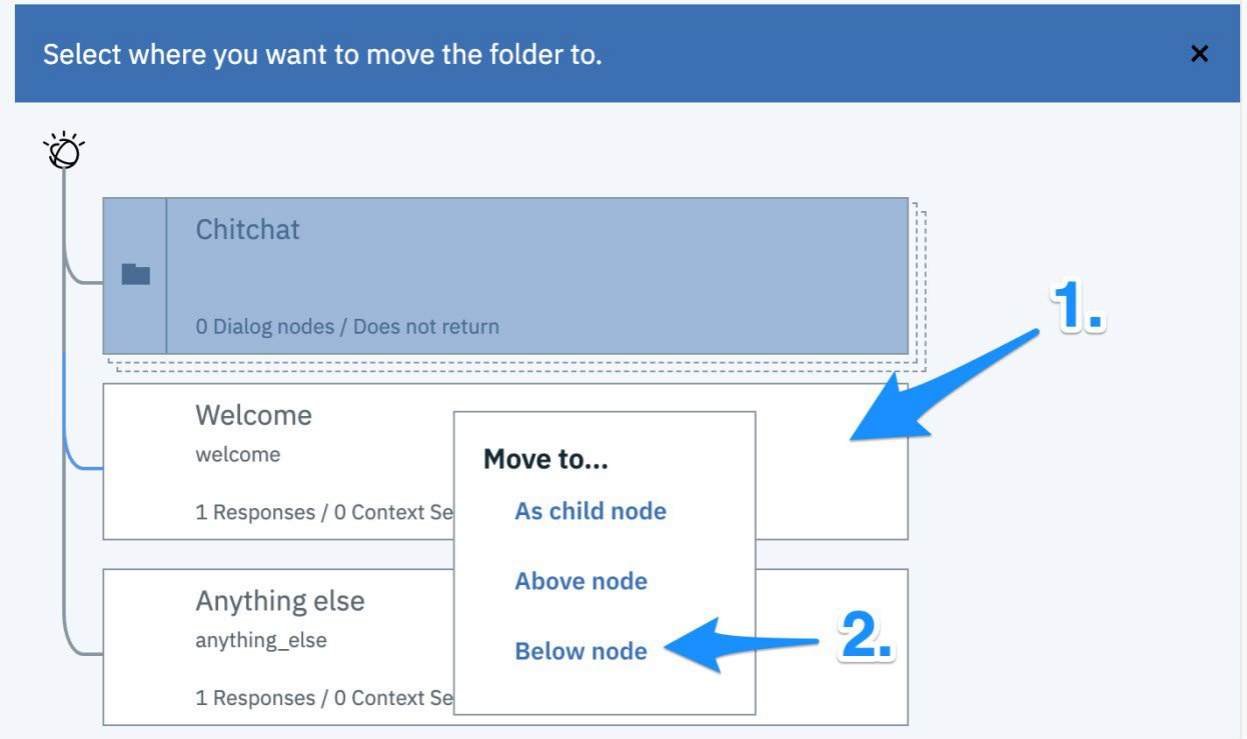
Follow these steps to implement this in your chatbot.

1. Click on the *Add folder* button. **Name the folder `Chitchat`.** You don't need to specify a condition for the folder, as the conditions of the child nodes will suffice.

2. We need to ensure that the folder is located between our prompt node (i.e., *Welcome*) and our fallback node (i.e., *Anything else*). So, if it was created above the *Welcome* node (or below *Anything else*), you'll want to click on the three vertical dot icon to the right of the folder and then click *Move*.

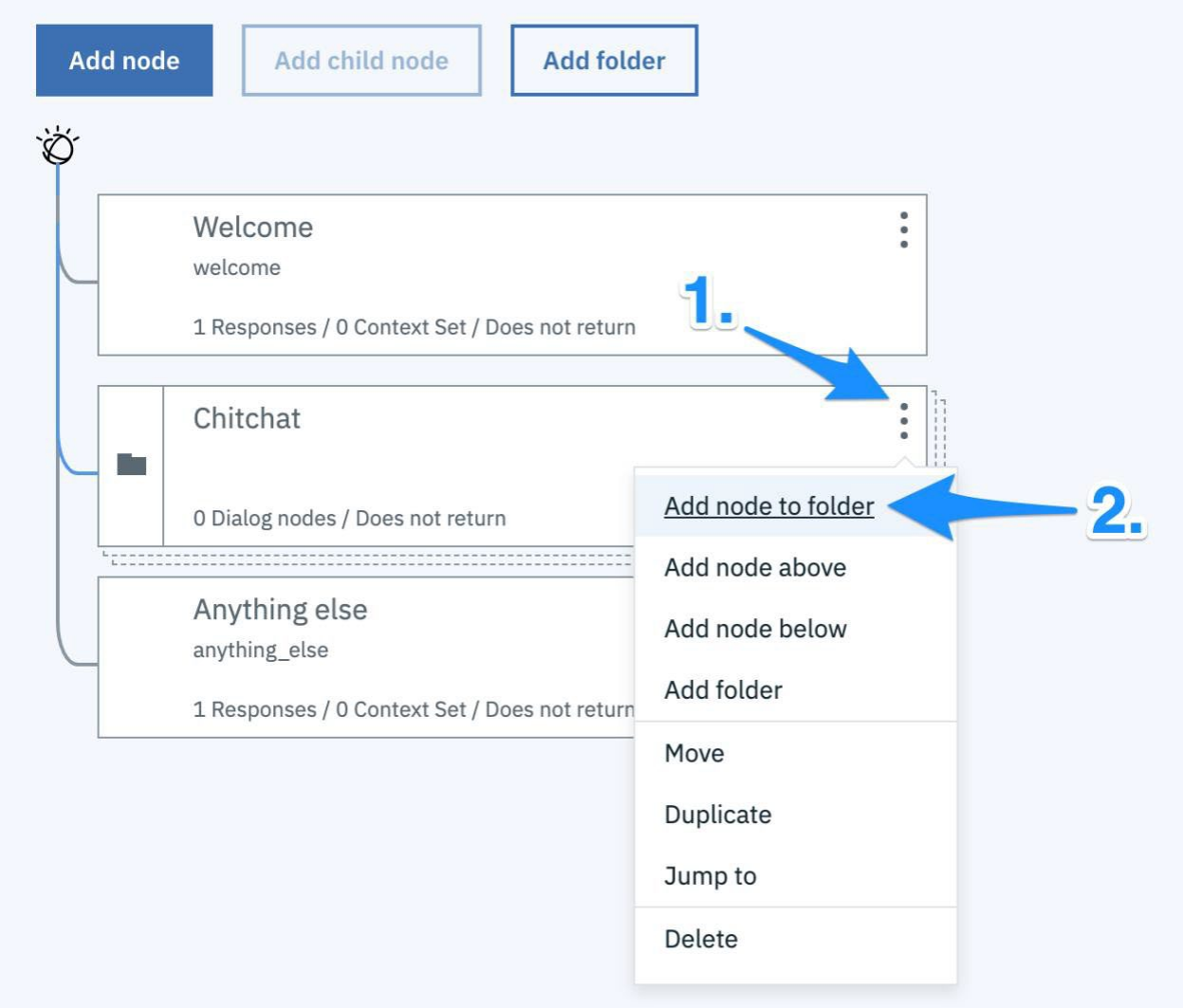


Next, you'll need to select the *Welcome* node, and click on *Below node*.



This will move the folder below *Welcome*. If it opened the folder for you, click on the X icon to close it and return to the dialog.

3. Select the options for the node (by clicking on the three vertical dot icon, sometimes called the more options or kebab menu) in the *Chitchat* folder, then **click *Add node to folder***, as shown in the screenshot below.



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 assistant 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:

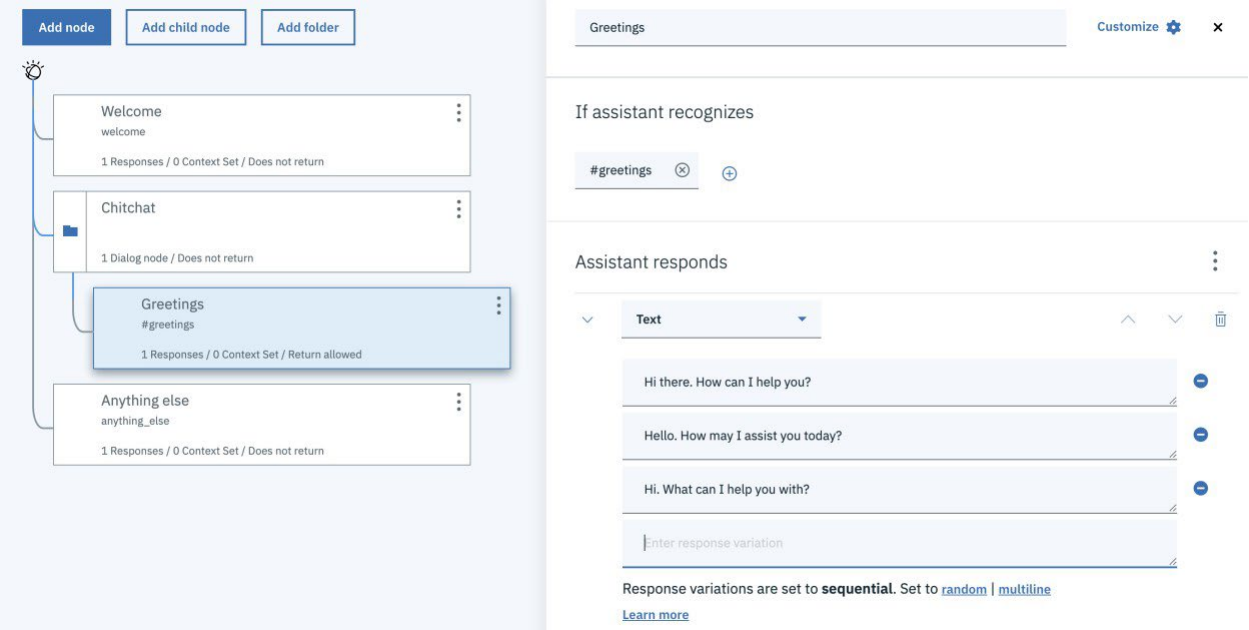
- `Hi there. How can I help you?`
- `Hello. How may I assist you today?`
- `Hi. What can I help you with?`

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 three variations of “how can I help you?” all at once.

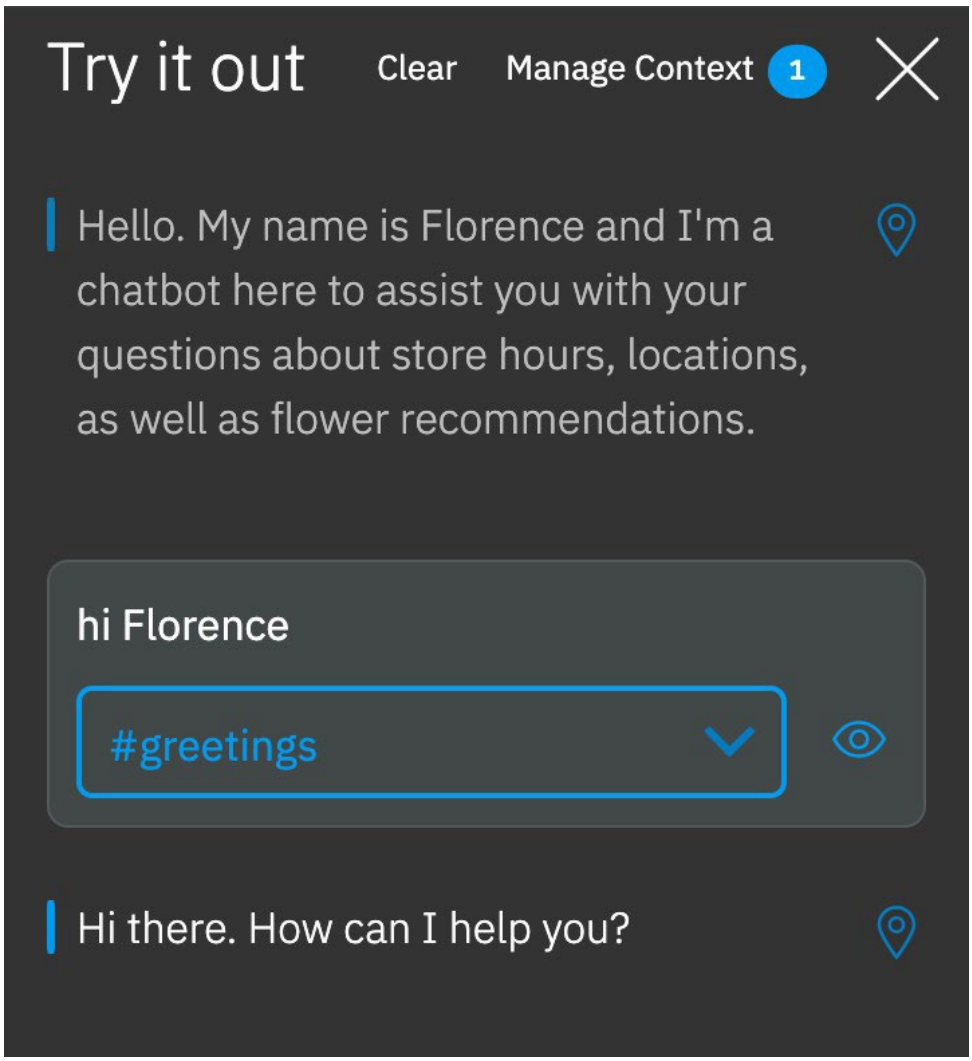
This is what the node will look like if you followed each step correctly.



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

8. Close the node and open the *Try it out* panel (if you closed it). Click the *Clear* link to start a new conversation. **Try to reply `hi Florence` to the chatbot prompt**.

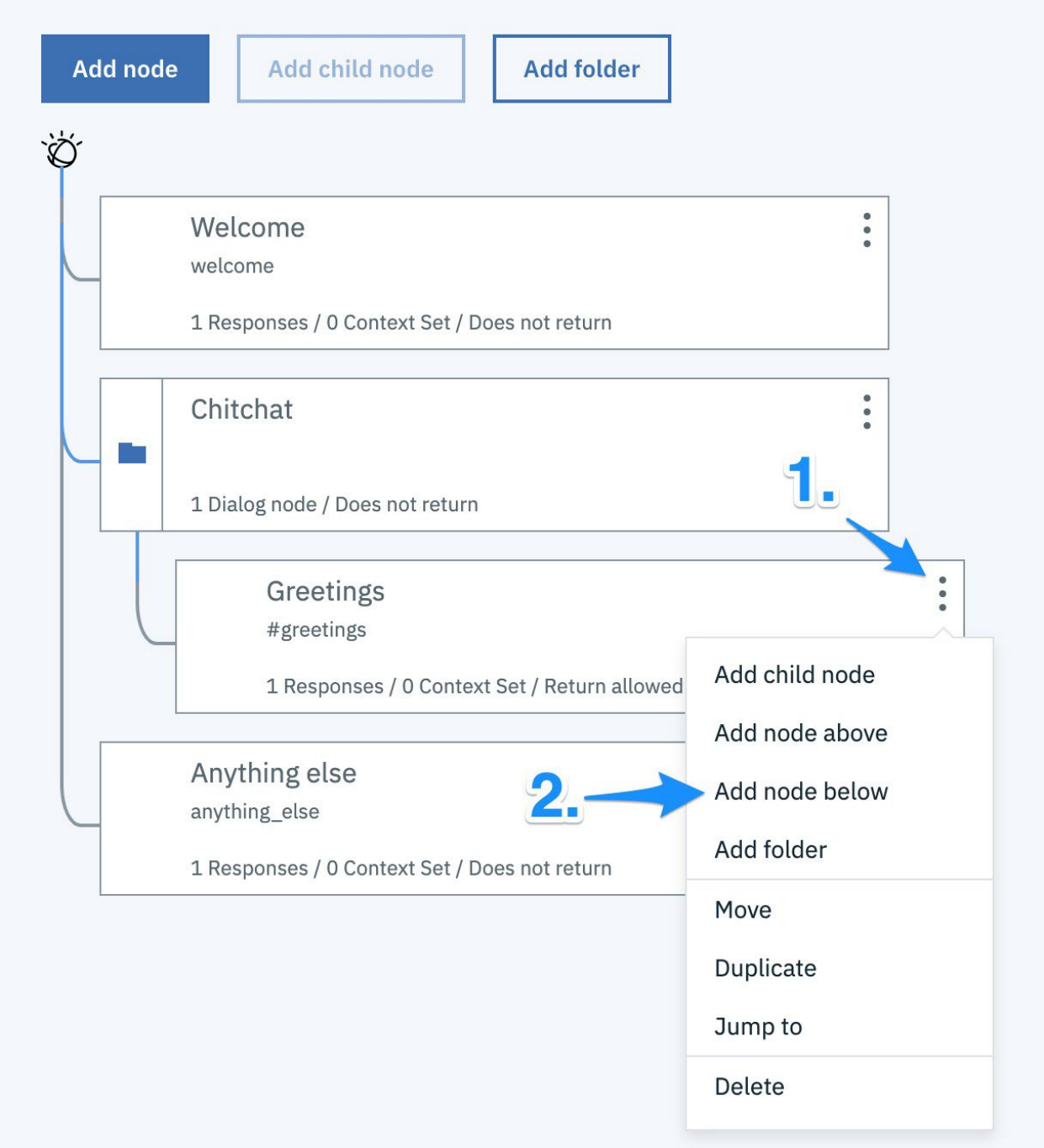
Congratulations if you see a proper response like the one in the image below! You just had your first conversation with our chatbot. It's not a complex interaction, but it's a good start.



You can now close the *Try It out* panel.

9. Now we must repeat the process to handle the scenario in which our user thanks us and the #thank_you intent is detected.

What we want to do is create a new node inside the *Chitchat* folder. **Click on the more options menu (the three vertical dot icon) for the Greetings node, then click *Add node below*.**



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).

It also makes sense to place them in a logical manner that is roughly equivalent to how a normal conversation would go. Greetings first, thanks in the middle, and goodbyes at the end.

Go ahead and make this node **handle the #thank_you intent**. You can name it whatever you like but Thank You will do. For the responses, you'll likely want something like:

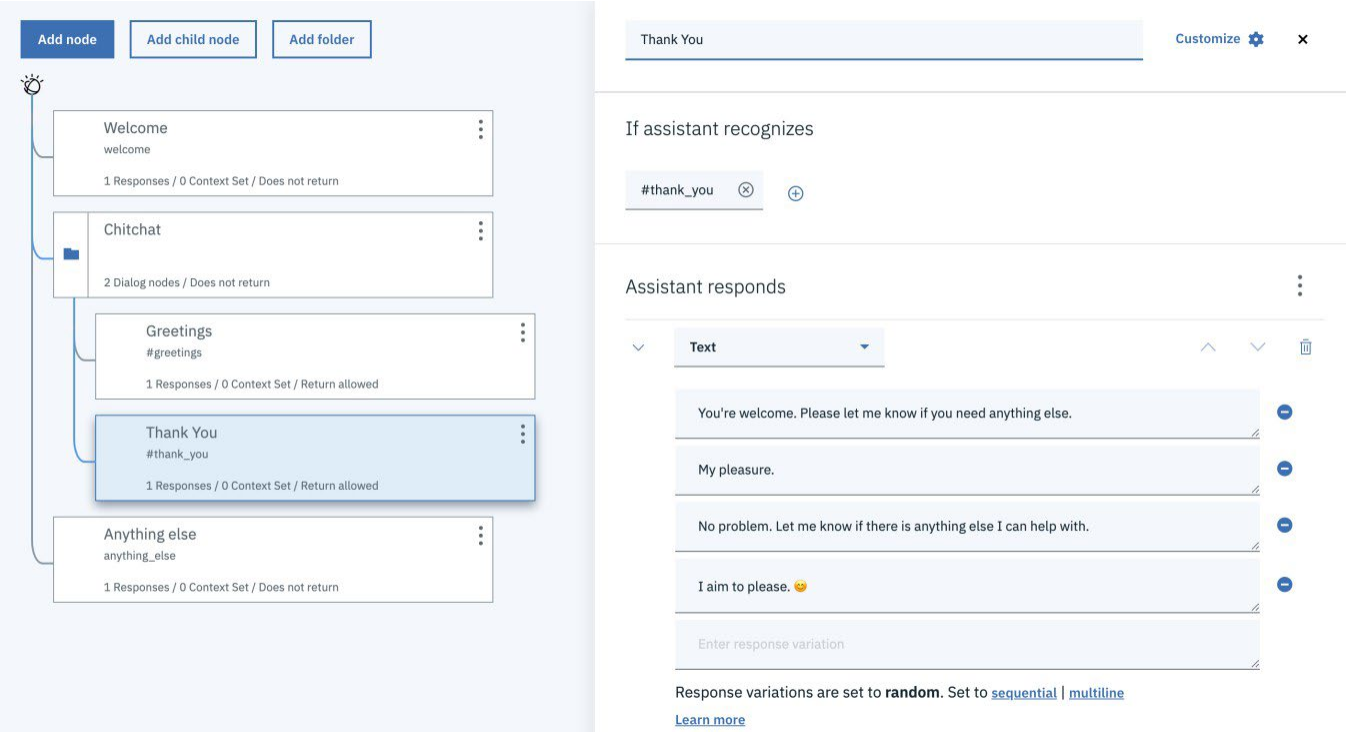
- `You're welcome. Please let me know if you need anything else.`
- `My pleasure.`
- `No problem. Let me know if there is anything else I can help with.`

You could get cheeky and add:

- `I aim to please.` 🙂

Of course, it depends on how much personality you'd like to inject into your chatbot. By the way, yes, emojis are supported.

Set the response variation to random by clicking on the *random* link.



10. **Repeat the process by adding a Goodbyes node** that handles the #goodbyes intent. This time make sure you use the *Thank You* node more options menu to select *Add node below*, since we want this third node to go below *Thank you*.

For now, you can use standard, polite goodbye responses such as:

- `Nice talking to you today.`
- `Have a nice day.`
- `Goodbye.`

(We'll improve these later on in the course.) Finally, **set their order to random.**

11. Start a new conversation in the *Try it out* panel and **test all three intents** to ensure you get a proper response in each case. As you can see below, Florence is coming along quite well.