|  |
| --- |
| DATA 650 |
| Mr. Cold Chatbot Walkthrough |
| Using Watson Conversation Service on IBM Cloud |

|  |
| --- |
| In this walkthrough, you will build a Mr. Cold chatbot that provides home treatment recommendation based on specified cold symptom. |



Contents

[Introduction 2](#_Toc505329733)

[Instructions 2](#_Toc505329734)

[Adding the Conversation Service 2](#_Toc505329735)

[Workspace 4](#_Toc505329736)

[Intents 5](#_Toc505329737)

[Adding an Intent Manually 6](#_Toc505329738)

[Importing the intents from a CSV file. 7](#_Toc505329739)

[Entities 9](#_Toc505329740)

[Creating Entities Manually 9](#_Toc505329741)

[Bulk Uploading the Entities from CSV File. 11](#_Toc505329742)

[Dialog 13](#_Toc505329743)

[Try it out 18](#_Toc505329744)

[Using Context Variables 19](#_Toc505329745)

[Using Slots 21](#_Toc505329746)

[Managing Handlers 23](#_Toc505329747)

[Downloading the Workspace as JSON file 25](#_Toc505329748)

[Troubleshooting 25](#_Toc505329749)

# Introduction

In this exercise, you will use Watson Conversation service on IBM Cloud to build a Mr. Cold chatbot that provides the treatment recommendations based on specified cold symptoms. If a user does not have any symptoms, the chatbot checks if an end-user had an annual flu shot.

Figure 1 shows the high-level dialog flow. The implementation of some conversation scenarios is part of the ungraded exercises.

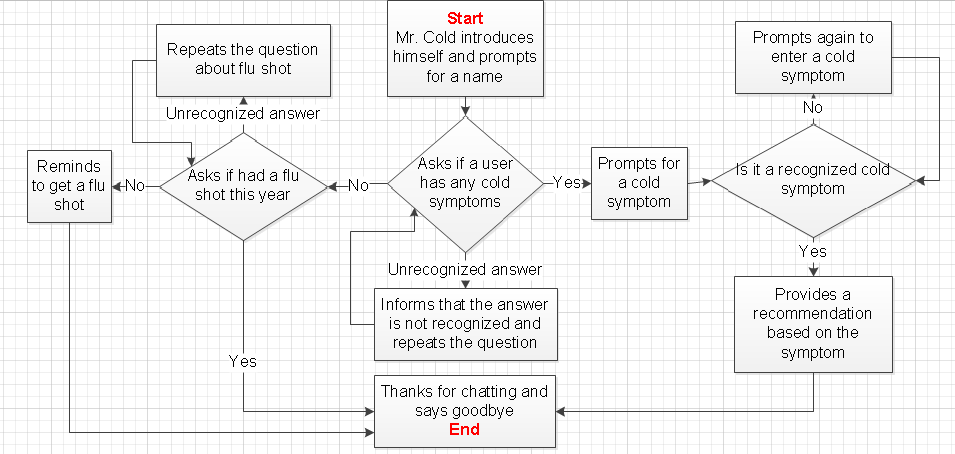


Figure 1: Dialog Flow

# Instructions

Before your start working on this assignment make sure that

* You have an active IBM Cloud account
* Completed the required readings in the course content
* Download intents.csv and entities.csv files from assignment folder

The walk through includes the ungraded exercises that need to be done in the same order as they appear in the document. Avoid proceeding to the next page until an exercise is complete. Although you do not need to turn in the answers, all exercises are **required** to get the maximum benefit from the walkthrough.

# Adding the Conversation Service

Login to your IBM Cloud account and click on Catalog at the top menu bar.



Figure 2: Click on Catalog

Select Watson in the left panel, and click on conversation in the right panel.

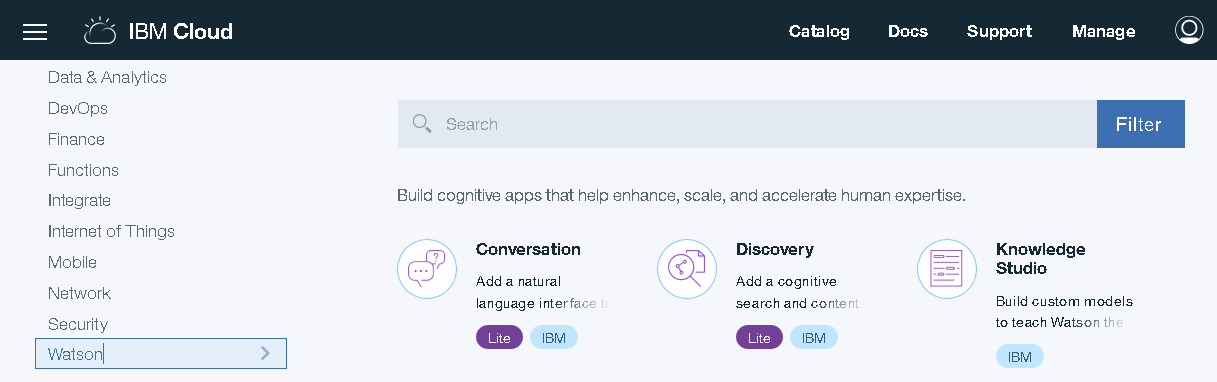


Figure 3: Click on Conversation service

The conversation service page will open. Click on Create to add a service to your dashboard.

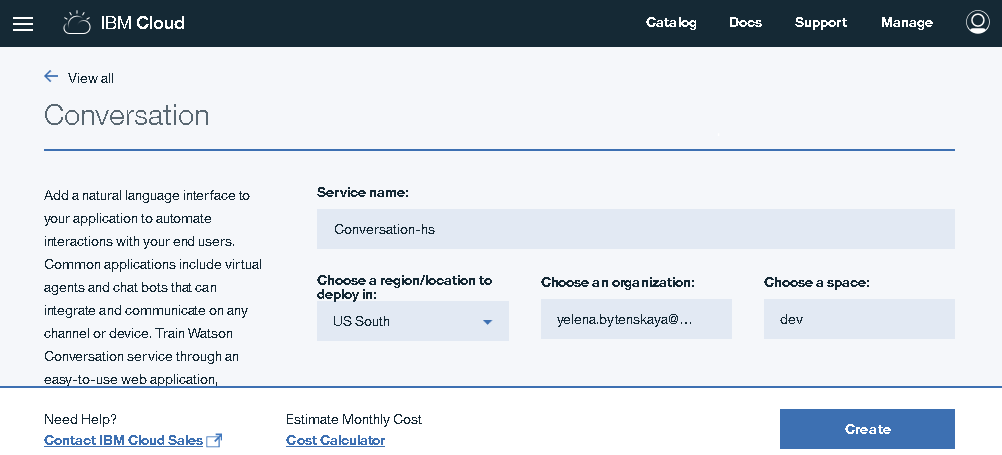


Figure 4: Conversation service create page

The Conversation service launch page will open. Make sure that Manage tab is selected in the left panel. Click on launch tool button to open an application.

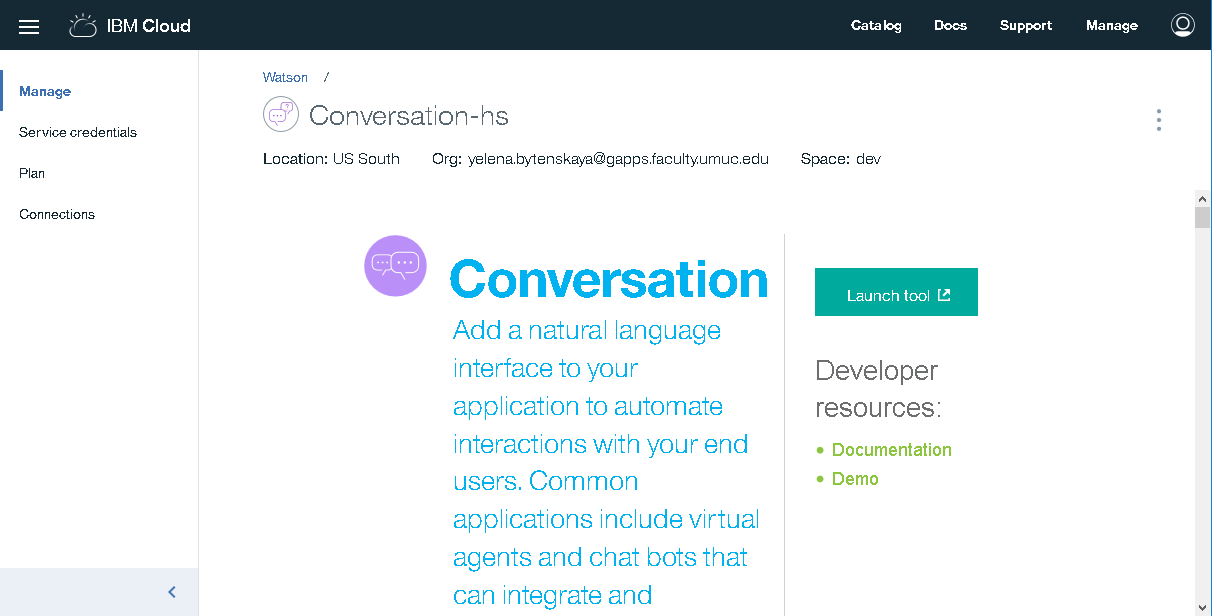
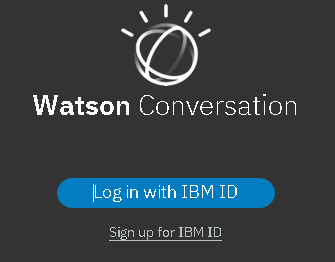


Figure 5: Conversation Service Launch Page

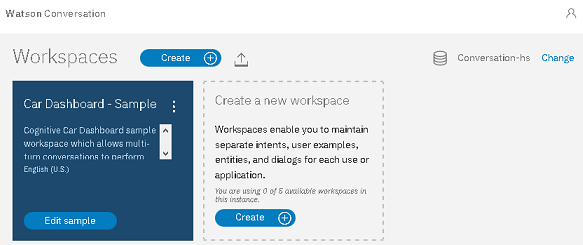


If prompted, click on Login with IBM ID button

Figure 6: Log in with IBM ID

You will be redirected to the workspaces page on Figure 7. The first tile is an editable Sample Car Dashboard application. Check the demo at the following web page.

https://conversation-demo.ng.bluemix.net/



Click to create new workspace

Sample Car Dashboard application

Figure 7: Workspaces Page

# Workspace

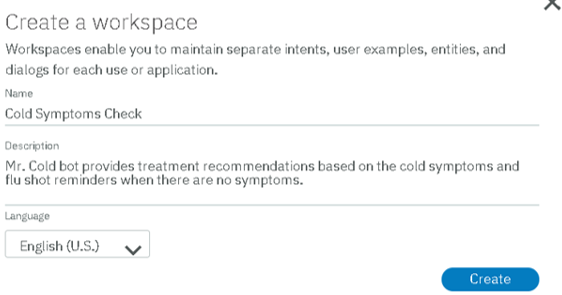
A workspace is a container where the chatbot conversation dialog tree model is built and where the model elements are defined. You may create up to 5 workspaces.

Click on create button in the second tile on Figure 7 to open the pop-up window on Figure 8.

Name the workspace *Cold Symptoms Check,* and enter the following description.

*Mr. Cold bot provides treatment recommendations based on the cold symptoms and flu shot reminders when there are no symptoms.*

Click create to continue.

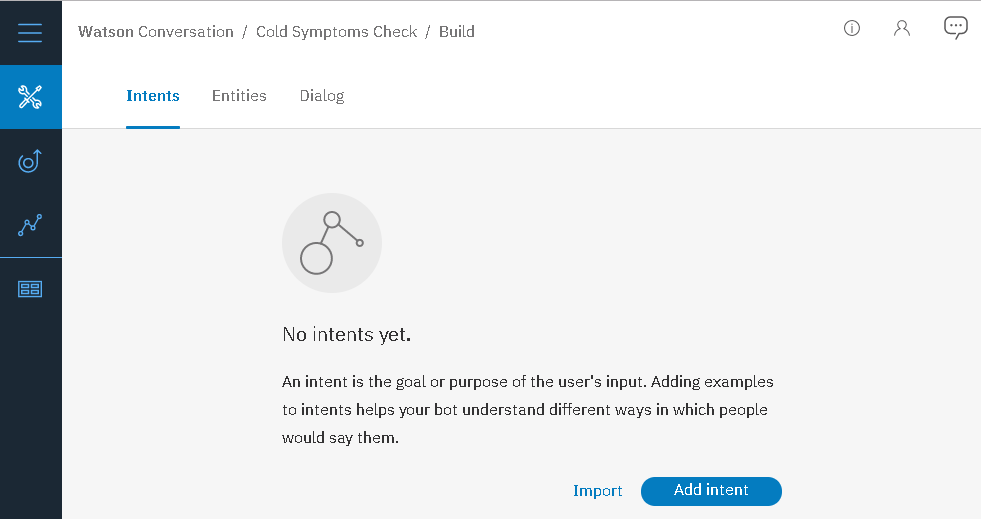


Workspace description

Workspace name

Figure 8: Create Workspace Window

You will be redirected to the intents tab. The No intents message is displayed since the new workspace does not have any intents.



Click if you need to return to the workspaces page

Click to add an intent manually

Click to open the help page

Click to bulk upload the intents from CSV file

Figure 9: Intents Tab

# Intents

An intent is a specific purpose of your input during the dialog. For example, it could be a request or a question that you may want to ask the bot. It could be an answer you enter at the prompt. In this dialog, the intents include answering if you have cold symptoms, and saying goodbye.

An example has the same meaning as an intent. For instance, bye, have a nice day, and see you later have the same meaning as goodbye.

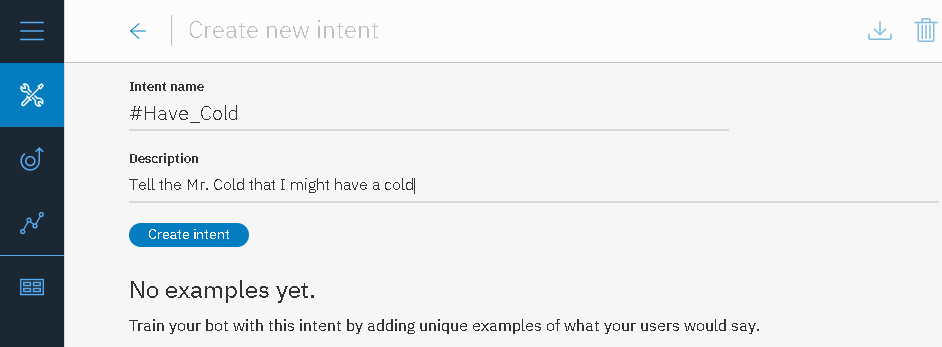
We may add intents manually or we may import intents from a CSV file.

Important rules

* The intent name starts with a # (hash). It’s included automatically when you create new intent.
* The spaces are not allowed in the intent name. An example may contain spaces.
* The intent names in a workspace must be unique. An intent cannot have duplicate examples.

## Adding an Intent Manually

Click on Add Intent button on the intents tab to open the create new intent interface.



Note: The hash tag in the intent name is included automatically when you create an intent

1. Name the intent Have\_Cold.
2. Enter the intent description Tell Mr. Cold that I might have a cold
3. Click on Create intent

Figure 10: Create New Intent

You will see a Have\_Cold added message.

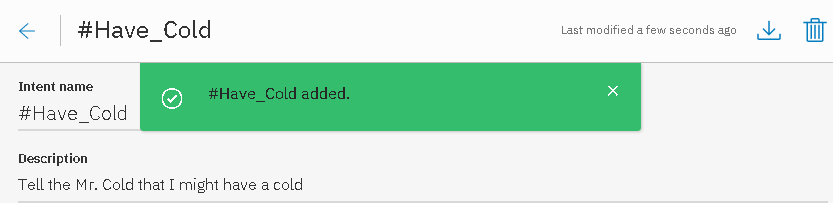
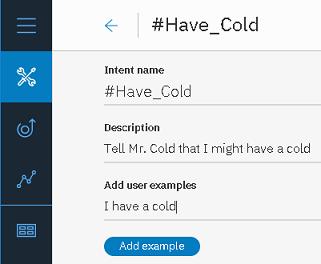


Figure 11: Intent Added Confirmation

Follow the steps on Figure 12 to add an example to #Have\_Cold intent.

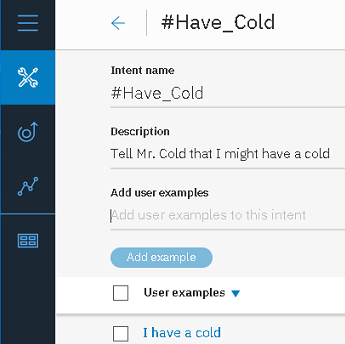


1. Enter I have a cold under Add user examples.
2. Click Add Example to continue.

**An example** is a potential user input that matches an intent.

Figure 12: Add example

We may continue adding examples, or we may return to the intents page.



Intent name

Intent description

.

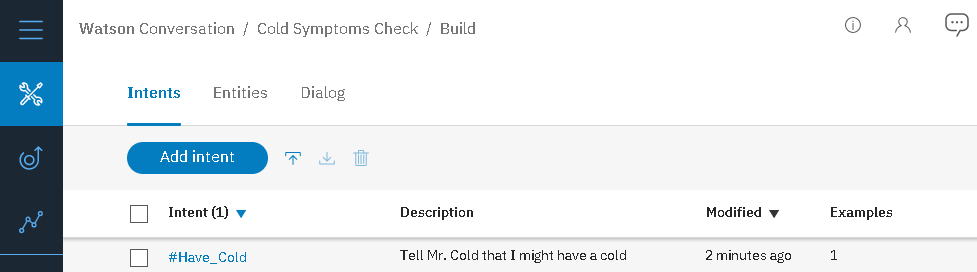
An example we just added is in the User examples section.

To return to the intents page, click on an arrow at the top of the screen next to the intent name.

.

Figure 13: User examples

The intent tab on Figure 14 displays an information on the intent we just added, including the intent name, intent description, and number of examples.

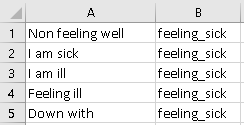


Note – if you need to return to intent details page, click on the intent name in the first column.

Figure 14: Intents tab displays Have\_Cold intent information

## Importing the intents from a CSV file.

Figure 15 shows the first 5 lines in the intents.csv file.



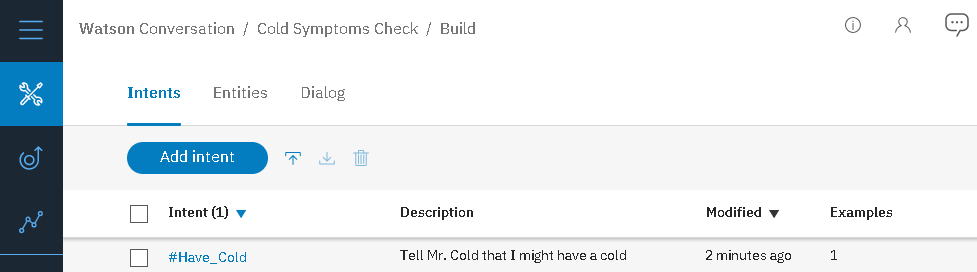
Column B is an intent name

Column A is an example of intent in column B

An intent name cannot contain spaces.

Figure 15: Partial content of the intents.csv file

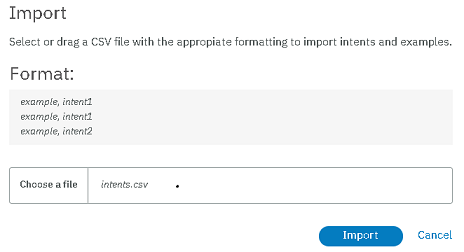
Click on an up arrow next to add intent button to open the import window.



Upload the intents data

Figure 16: Click to open import window.

Click on choose a file, and find the intents.csv file. Click on import to continue.

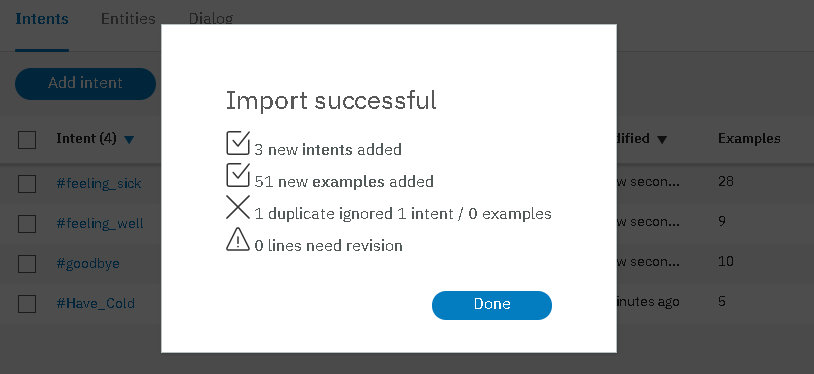


Column B in intents.csv file

Column A in intents.csv file

Figure 17: Upload the intents.csv file

The import successful message on Figure 18 will display. The message specifies the number of new entities and number of new examples.



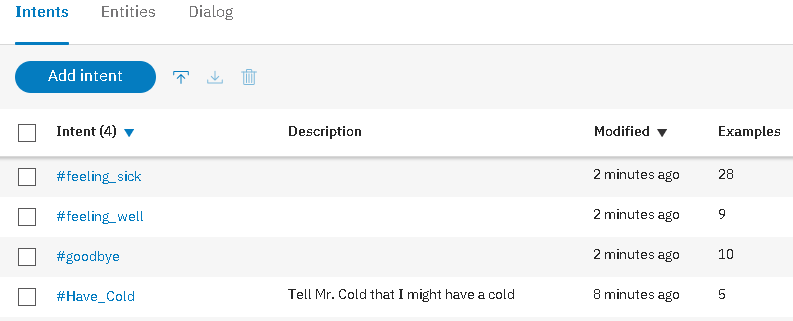
Click done to continue

If an intent with the same name did not exist, a new intent is created

If an intent with the same name already existed, the new examples are added to an existing intent. An example is ignored if the corresponding intent already has the same example.

Figure 18: Import successful

Figure 19 shows an updated intents page. Intents feeling\_sick, feeling\_well, and goodbye are new intents. The intent Have\_Cold was updated with new examples.



The intents can be sorted by name or by modification date. Click on an arrow to sort.

.

Figure 19: Intents tab with 4 intents

You may click on intent name in the left column to view the examples.

# Entities

**Ungraded Exercise 1**

Click on #Have\_Cold intent. What new examples were added when you imported the intents.csv file?

The entities are useful for collecting more detailed information in the conversation, especially when the details determine the bot’s response. For instance, the Mr. Cold bot suggests the treatment options are based on the entered cold symptom. We use the cold symptom entity to differentiate between the symptoms.

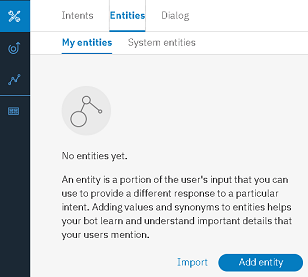
An entity contains a set of values. The specific cold symptoms are the values in the cold symptom entity. In addition, we may specify the synonyms for each symptom. The synonyms are optional.

Important rules

* An entity name starts with an @
* An entity name cannot contain spaces
* The entity names in a workspace must be unique

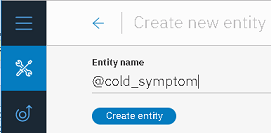
We may create entities manually or we may import the CSV file.

## Creating Entities Manually



1. Click on Entities tab.
2. Make sure that My entities subtab is active.
3. Click on add entity button.

Figure 20: Empty My entities page



1. Enter cold\_symptom into an entity name.
2. Click create entity to continue.

Figure 21: Enter the entity name

You will see a @cold\_symptom added confirmation.

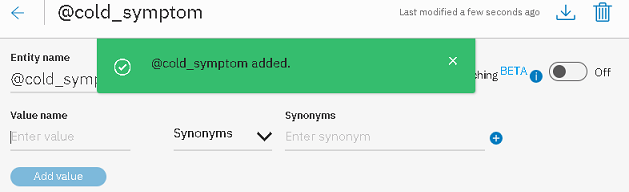


Figure 22: @cold\_symptom added confirmation

Enter cough into a value name and cough up mucus into the synonyms.



Figure 23: Add a Value with a synonym

Click on a + next to the synonym you just entered to add another synonym.

Enter dry cough and click add value to continue

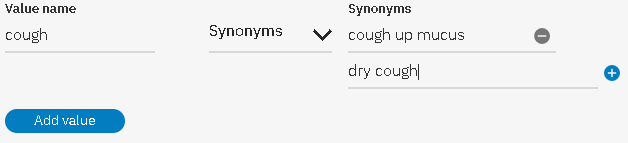
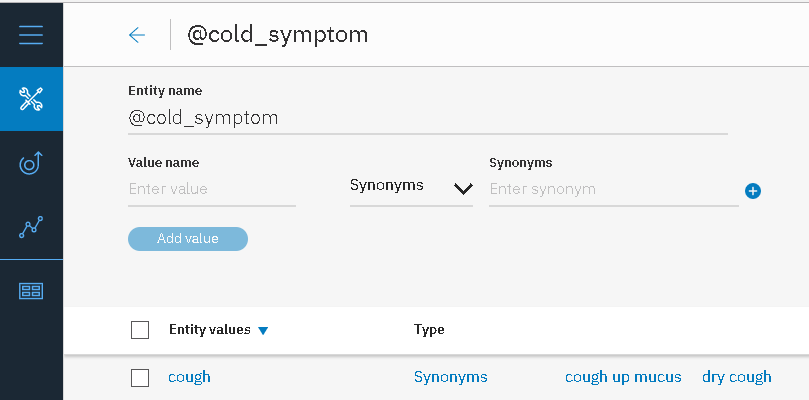


Figure 24: Enter the second synonym

The entity value cough and 2 synonyms will appear in the entity values section of the page.

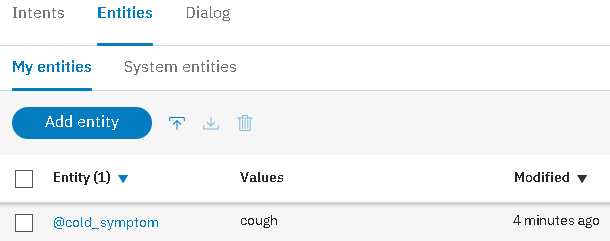


To return to entities page, click on this arrow

Figure 25: Entity values section

Click on an arrow at the top of the page on *Figure 25* to return to the Entities tab.

The “no entities yet” message was replaced with the @cold\_symptom entity information, including name and values.



Import from file

Figure 26: @cold\_symptom entity is listed on an Entities tab

## Bulk Uploading the Entities from CSV File.

*Figure 27* shows the first 5 rows in the entities.csv file that we are going to import.

Column A is an entity name. Colum B is the value of an entity in column A. The remaining columns are the synonyms for a value in column B.

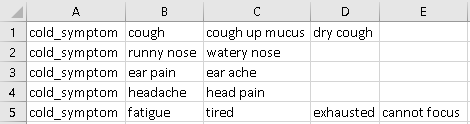
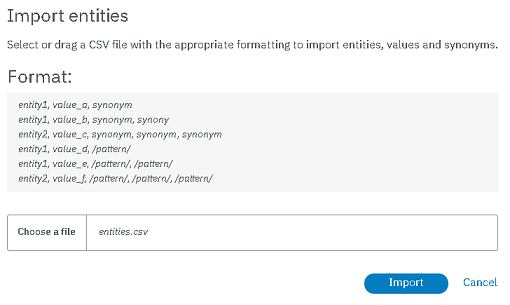


Figure 27: first 5 rows in entities.csv file

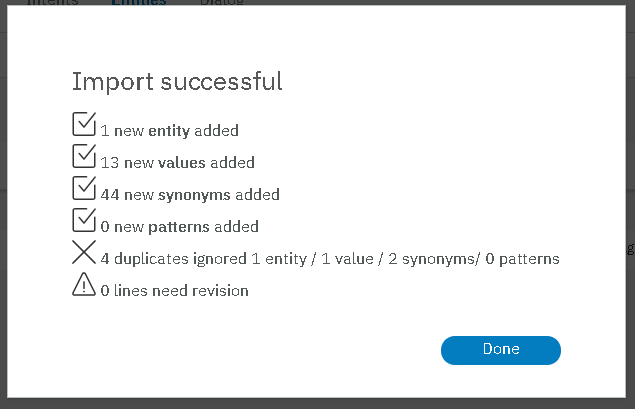
Click on an up arrow next to add entity button on *Figure 26* to open the import window on *Figure 28*,.



1. Click on choose a file, and find the entities.csv file.
2. Click Import.

Figure 28: Import entities window

The import confirmation will display. Click Done to continue.



If an entity with the same name existed, the import did not create another entity with the same name.

Instead, the new values and new synonyms are added to the existing entity.

Figure 29: Import Successful Confirmation

*Figure 30* shows the entity names and partial list of values. To view the synonyms for each value, click on the entity name in the first column.

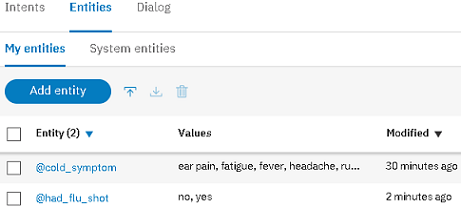


Figure 30: My entities tab after entities.csv file import

**Ungraded Exercise 2**

Click on @cold\_symptom entity name in the first column. How many values does an entity have? What synonyms are associated with the value fever? What are the main reasons for specifying the synonyms?

**Ungraded Exercise 3**

What are the differences between user entities and system entitles? Navigate to the entities tab, and click on system entities subtab. What system entities are available, and when would you use them?

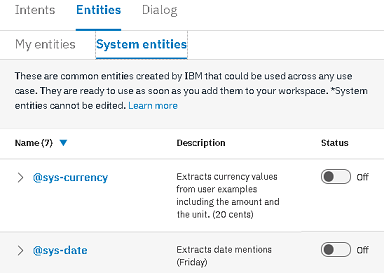
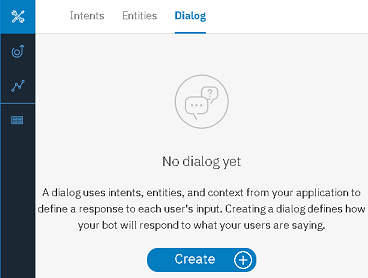


Figure 31: Navigate to System Entities Page

# Dialog

The dialog uses intents, entities, and context variables to define an interaction with the bot. It’s represented as a hierarchy of nodes, and an execution starts from the top node.

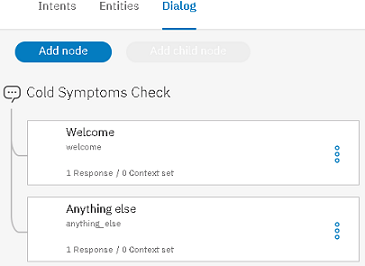


Make sure that the dialog tab is selected.

Click on Crate button to start building a dialog.

Figure 32: The dialog tab

This is the workspace name

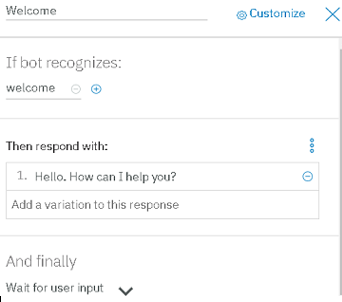


The welcome node is always executed when the dialog starts.

The anything else node is executed when the user’s input does not match the condition in any nodes above. A user is prompted to paraphrase the response.

Figure 33: Dialog with only Welcome node and anything else node

Click on a Welcome node to check the configuration explained on Figure 34.



Node Name

Condition that triggers the node execution.

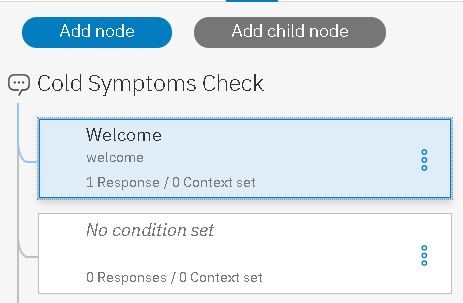
The welcome condition means that the dialog execution starts from this node.

The response is a message displayed to a user. It could be a prompt for information or an answer to user’s question.

Since the node does not have any child nodes, once the user provides the input, all nodes will be reevaluated starting from the top.

Figure 34: Welcome Node Configuration

Next, we add a node to handle the goodbye intent.



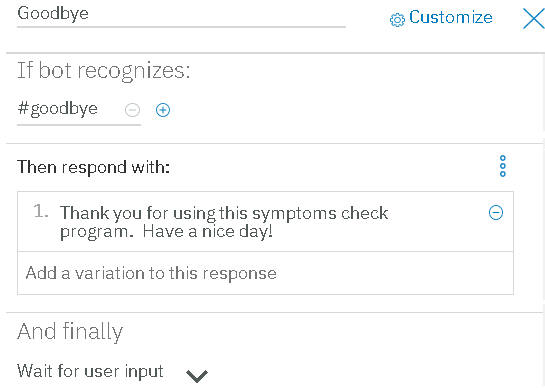
Select welcome node.

Click on Add Node to add a new node below it.

Since we have not configured the node, you see “No condition set” instead of the node name.

Figure 35: Add node under welcome node

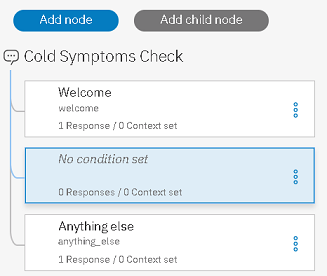
Click on a node you just added to configure the name, entry condition, and response.



1. Name the node Goodbye
2. We want the node to be executed when the #goodbye intent is recognized.
3. Enter a response “Thank you for using this symptoms check program. Have a nice day.”
4. Add the response variations to allow different responses when the node is executed.
5. Keep Wait for user input as a final action.

Figure 36: Goodbye node configuration

Next, we add a node to handle the not feeling well intent. Click on a welcome node, and click add node.

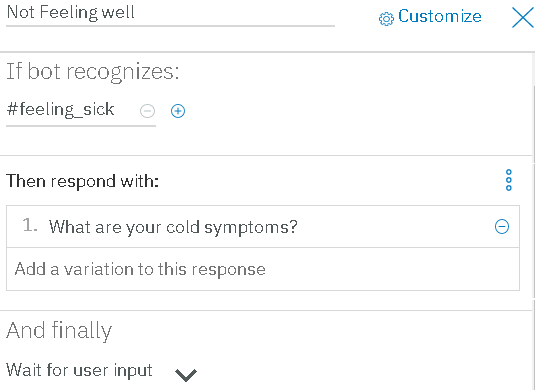


The new node we just added under the welcome node.

Double click on the node to configure it.

Figure 37: Add another node under Welcome node.

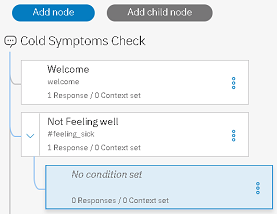
Follow Figure 38 to configure the Not Feeling well node that you just added.



1. Name the node Not Feeling well.
2. This node is triggered by the #feeling\_sick intent
3. Enter a response “What are your cold symptoms”
4. The finally should be set to wait for user input because the next bot’s action depends on response.

Figure 38: Not Feeling well node configuration

We add a new node to handle the recommendation based on specified symptom. To make sure that the recommendation is handled once the user specifies the symptom, the new node should be added as a child of not feeling well node.



The child have a cold node

Select on Not Feeling well node and click on add child node.

Figure 39: Add a Have a Cold child node to the Not feeling well node

Click on a node you just added to configure it.

1. Name the node Have a Cold
2. The node is triggered by either #Have\_Cold intent or by any value of the @cold\_symptom entity.
3. Click on Customize

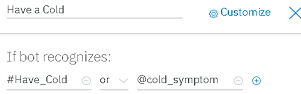
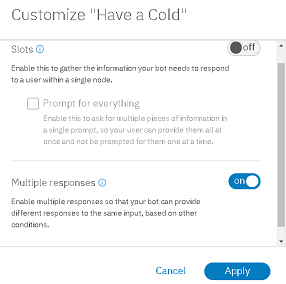


Figure 40: Have a Cold node name and condition



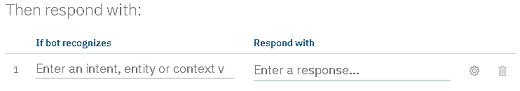
Since the response depends on the specified cold symptom, we need to enable the multiple responses.

Turn on the multiple responses.

Click apply

Figure 41: Enable multiple responses

The respond with configuration section now has 2 columns. We specify the condition in the first column and the response in the second column. Alternatively, we may click on a gear icon on Figure 42 to open the response configuration window on Figure 43.



Click to open response configuration

Figure 42: Click on Gear to configure response 1

If the cold symptom is the shortness of breath, a user should consider going to emergency room.

1. Enter

@cold\_symptom:(shortness of breath)

Since the entity value shortness of breath contains spaces, it’s specified in parenthesis. The : is a separator between an entity name and a value.

1. Enter the response Consider going to emergency room as soon as possible.
2. Click Save

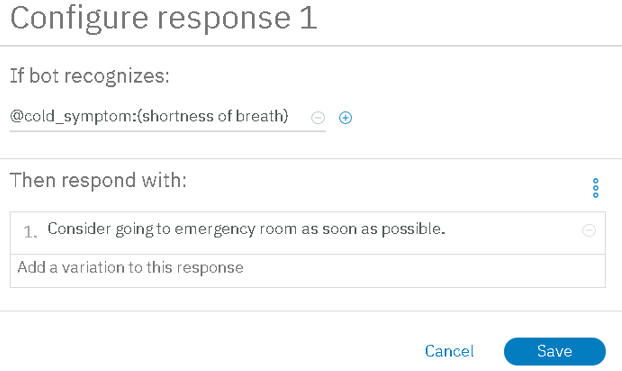
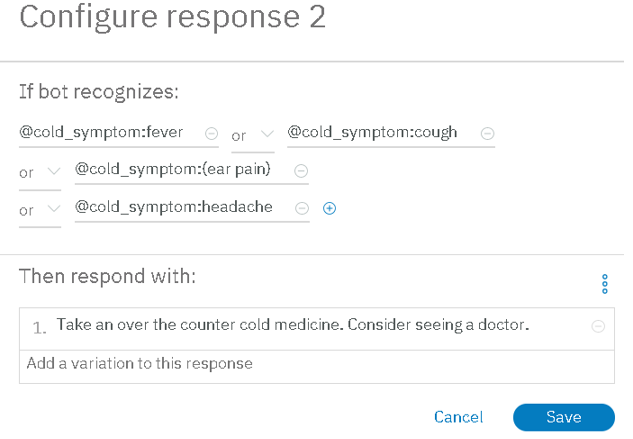


Figure 43: Response 1 configuration

Click on add response in the respond with section, and follow Figure 45 to configure the second response.



Figure 44: Click on Add response to add another response



1. This response should be triggered If cold symptom entity value is either fever, or ought, or ear pain, or headache.
2. Respond with - Take an over the counter cold medicine. Consider seeing a doctor.
3. Click on Save

The entity values can be combined using “or” and ”and ” operators.

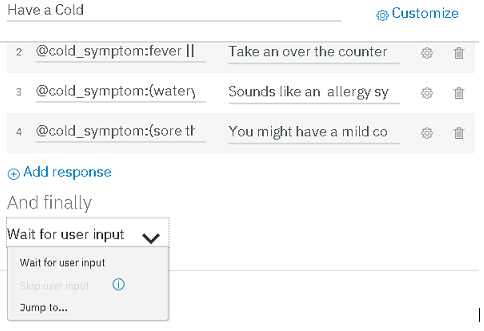
Figure 45: Configure second response

**Ungraded Exercise 4**

Configure the third response as - If the bot recognizes cold symptoms watery eyes, or sneezing, or stuffy nose, or runny nose, respond with Sounds like allergy symptoms. Take an antihistamine.

Confiture the fourth response as If the bot recognizes cold symptoms sore throat or stiff neck, response with You might have a mild cold. Try over the counter cold medicine and get some rest.

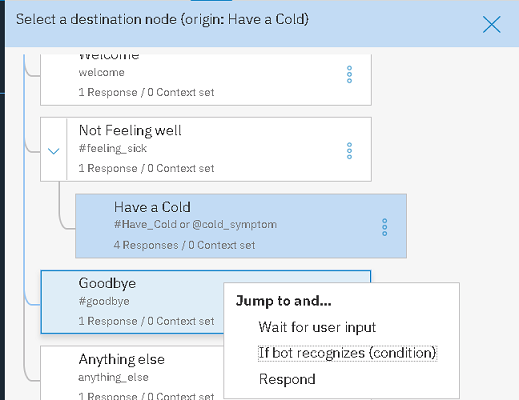
Set the finally action to jump. Follow Figure 47 to select the goodbye node as a destination and choose respond.



Click on an arrow next to “And finally”

Select jump to

Figure 46: Select Jump To for the finally action



Click on Goodbye node

Select respond

Figure 47: Choose a Goodbye node as a destination

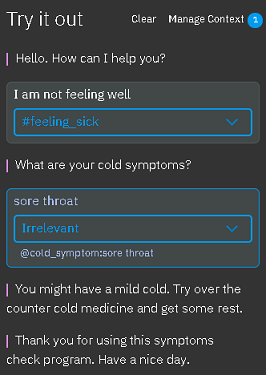
# Try it out

Click on ask Watson chat bubble Icon in the top right corner to test the dialog.



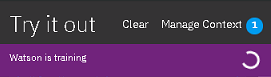
Figure 48: Click on Ask Watson

The try it out panel on   
Figure *49* will open. The “Hello. How can I help you?” message is a response in the welcome node. Follow the steps on   
Figure *49* to test the bot and to check if the correct intents in the input are recognized.

  
Figure 49: First Try it out attempt

1. The response from the welcome node is displayed
2. The I am not feeling answer is recognized as a #feeling\_sick intent.
3. The not feeling well node is triggered, and its response is displayed. The boat is waiting for a user intput.
4. An answer sore throat is recognized as a sore throat value of the @cold\_symptoms entity. However, it does not match any intents.
5. The have a cold child node is still executed because the recognized @cold\_symptoms entity meets the entry condition.
6. The @cold\_symptoms entity value sore throat triggers the fourth response.
7. The execution jumps to the Goodbye node.

To train the bot, click on an arrow next to irrelevant, and select Have\_cold intent. You will see a Watson is training message. Wait for the training to complete.



Wait for the training to complete. Then click on clear to restart the conversation.

Figure 50: Watson is training

Going forward, the answer sore throat to the cold symptoms question will be recognized as a Have\_Cold intent in addition to @cold\_symptoms entity.

# Using Context Variables

**Ungraded Exercise 5**

Open the try out panel. Respond I am not feeling well to the first prompt. Respond leg to the question about the cold symptom. What does the bot return and why?

**Ungraded Exercise 6**

Open the try it out panel. Enter what time is it in response to the How can I help you prompt. What message does the bot return and why?

We use the context variables to store user input. The stored values can be used through the rest of the dialog. To access the value, we prefix the variable name with $

For example, we may prompt the users for a name in the Welcome node and store the input in a variable $name.

Click on a Welcome node and change the response text to

“Hello. My name is Mr. Cold. My mission is to recommend the treatment for the common cold symptoms. May I have your name?”

Click on Welcome node and add a child node. Click on node you just added to configure the properties.

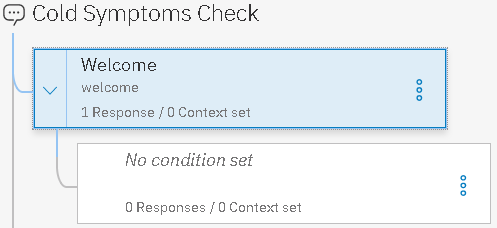
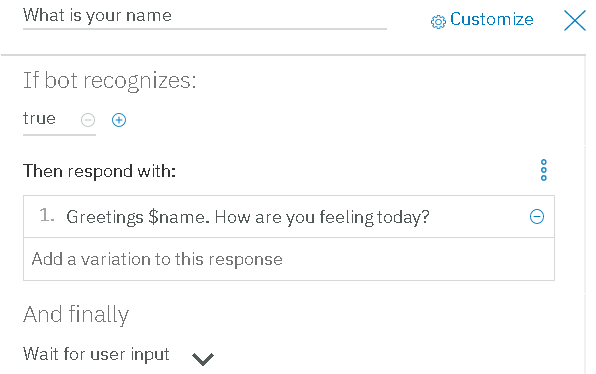


Figure 51: Welcome node and its child node



1. Name the node What is your name
2. Set entry condition to true because we want this node to execute each time the Welcome node is triggered.
3. Enter the response to Greetings $name. How are you feeling today.

$name is a variable we will use to store the name

1. Click on ellipses to open the JASON editor.

Figure 52: What is your name node customization

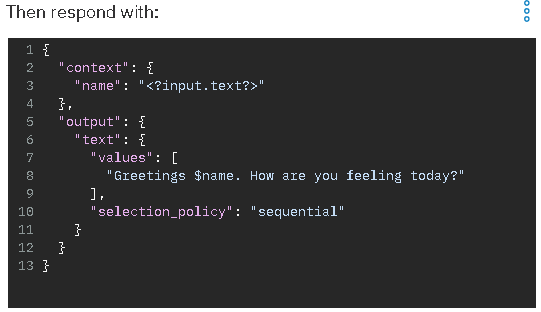
To capture the input into $name variable, enter the following code after the first {

"context": {

"name": "<?input.text?>"

},

The input.text is the user input. The leading and training question mars prevent the null pointer exception.



1. Add the code to capture the input in the name variable.
2. Click on ellipses to close the JSON editor.

Figure 53: JSON Editor code for What is your name node response

**Ungraded Exercise 7**

Try out the dialog. What message is displayed when you start the conversation? Enter your name. How does the bot respond?

**Ungraded Exercise 8**

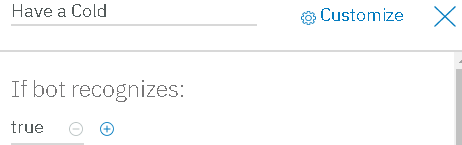
Update the response in the goodbye node to include user’s name stored in the $name variable at the beginning of the dialog.

# Using Slots

Currently, if the cold symptom entered by user is unrecognized, the anything else node is executed. To repeat prompting a user for the cold symptom until the valid response is entered, we use a slot.

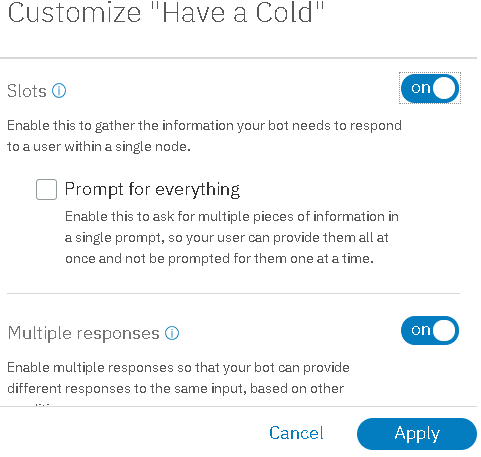
We store the response in the context variable $cold\_symptoms and check if the value matches the @cold\_symptoms entity.

Click on Have a Cold node to update the configuration.



1. Set the node entry condition as true.
2. Click on Customize

Figure 54: Change Node Entry Condition and Click Customize



Change the slots setting to on

Click Apply

Figure 55: Enable Slots for Have a Cold node

A new section “Then check for” is added to the node configuration. Click on a gear to open the slot customization window.

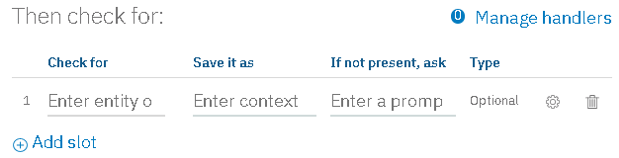
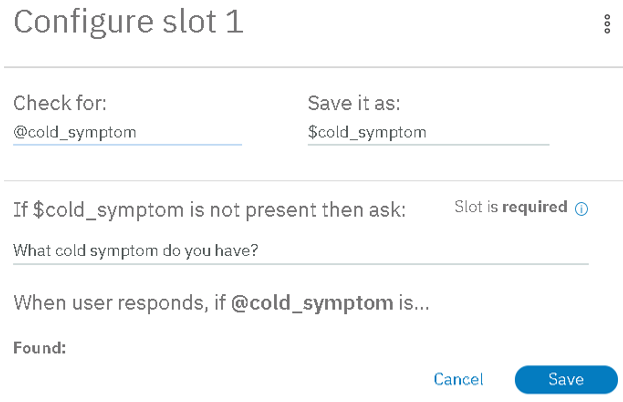


Figure 56: Click on Gear to Open Slot Configuration



1. Check for @cold\_symptom entity
2. Store the input in the $cold\_symptom variable
3. Enter “What cold symptoms do you have?” text. It’s a prompt to reenter the cold symptom
4. Click save

Figure 57: Slot Configuration

# Managing Handlers

**Ungraded Exercise 9**

Test the dialog.

1. What happens when you enter the recognized cold symptom?
2. Click clear at the top of the Try it out panel. Repeat the test, but enter leg instead of valid cold symptom. How does the bot respond?
3. Enter the invalid cold symptom again. How does the bot respond?
4. Enter the “runny nose.” How does the bot respond?
5. How does the bot handle the spelling errors?

We may use handlers within the slots to manage the input that does not meet the slot check for condition. For example, we may use a handler to stop prompting for the cold symptom if a #goodbye entity is recognized in the response.

Click on Have a Cold node, and click on manage handlers.

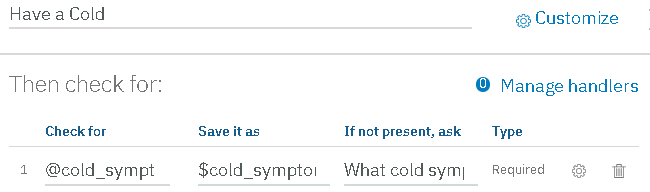


Figure 58 Click on Manage handlers

Click on a gear on Figure 59 to open the handler configuration window on Figure 60.

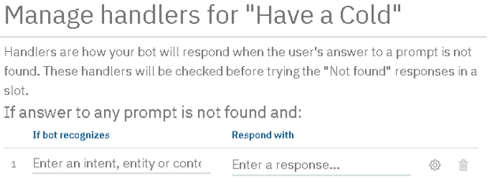
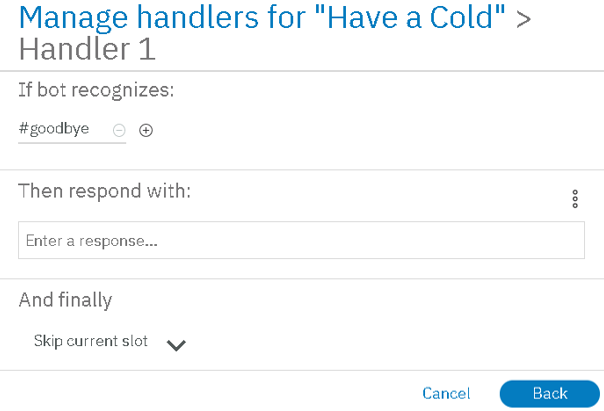
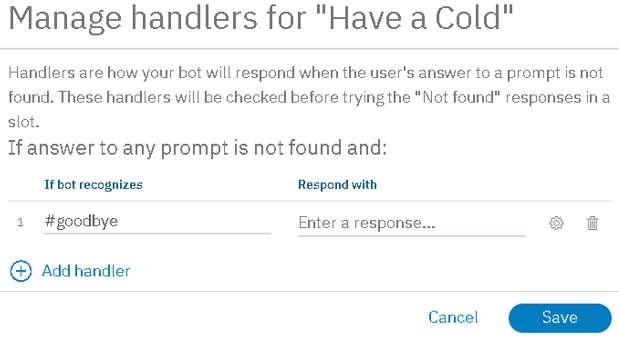


Figure 59: Manage handlers for “Have a Cold”



1. Enter #goodbye intent
2. Set the finally action to skip the slot.
3. Click Back.

Figure 60: Handler Customization



1. Click Save

Figure 61: Save Handler Customization

**Ungraded Exercise 10**

Test the dialog. Enter an invalid cold symptom. When prompted to reenter the cold symptom, enter goodbye. How does the bot handle the response?

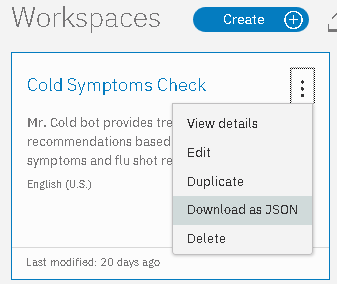
**Ungraded Exercise 11**

Edit the dialog to handle the scenarios when a user does not have cold symptoms. The bot should continue asking if a user had a flu shot this year until the valid response is recognized or until the #Goodbye intent is recognized.

If a user did not have a flu shot, the bot should respond “Make sure to get a flu shot at the earliest convenience.” If a user had a flu shot, the bot should respond “Excellent!”

# Downloading the Workspace as JSON file

To back up the work, you may download the conversation workspace as a JSON file and save the file on your PC.



1. Return to the Workspaces page, and find the tile for Cold Symptoms Check workspace.
2. Click on ellipses to open the menu.
3. Select Download as JSON from the menu.
4. Save the file on your PC. Depending on a browser you use and on browser settings, you might be prompted to choose the destination folder.

Figure 62: Select Download as JSON from workspace menu

# Troubleshooting

If you receive unable to add sibling node error, your session may have timed out.

1. Close and reopen the browser.
2. Login to Bluemix, and launch Watson Conversation service again.

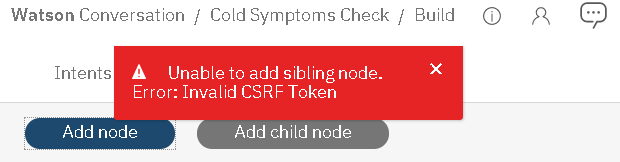


Figure 63: Unable to add sibling node error