Microsoft Power Virtual Agents in a Day

Lab 05: Integrate Pre-built Power Automate Flows

Hands-on Lab Step-by-Step

January 2022

Contents

Power Automate

This lab is subject to the Terms of Use on page 28 of this document.

Goals for this lab

What you'll learn: • Integrate Power Virtual Agents with Power Automate to initiate bot actions • Create a topic that can complete a return and exchange process • Connect Microsoft Dataverse to your chatbots, allowing for certain data recall between sessions.	The time to complete this lab is [30] minutes.

Before we start…

Ensure that you:

- **Imported** the Virtual Agent in a Day solution package to the environment where you have built your bot.
- **Fixed** connections for the imported Power Automate flows as instructed in the prerequisites.

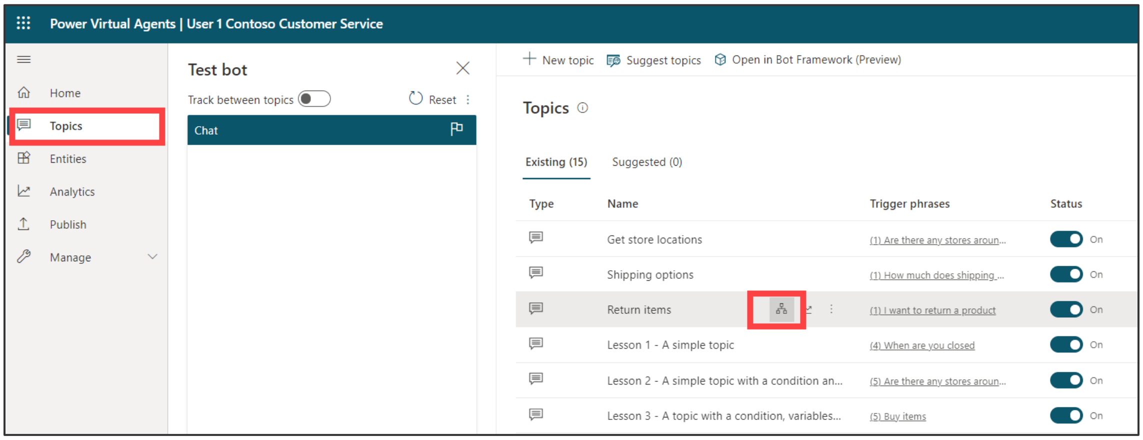
The above steps should already be completed if your trainer has provided you with an environment to use for the labs.

Navigate to https://powerva.microsoft.com/ and sign in with your credentials.

Task 1: Add on to the Return Items topic

1. Click **Topics** in the left navigation pane.

2. Find the topic you built in Lab 4, **Return items**. Hover your mouse over the topic and then click the **Go to authoring canvas** button that appears. (See screenshot below.)



Power Virtual Agents | User 1 Contoso Customer Service

Home
Topics
Entities
Analytics
Publish
Manage

Test bot
Track between topics
Reset
Chat

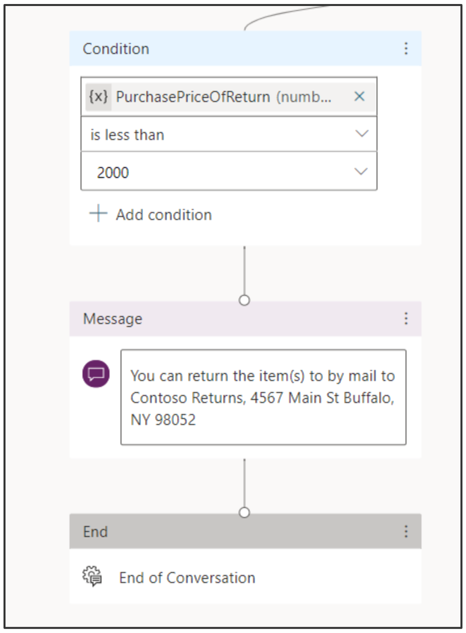
New topic
Suggest topics
Open in Bot Framework (Preview)

Topics

Existing (15)
Suggested (0)

Type	Name	Trigger phrases	Status
	Get store locations	(1) Are there any stores aroun...	On
	Shipping options	(1) How much does shipping ...	On
	Return items	(1) I want to return a product	On
	Lesson 1 - A simple topic	(4) When are you closed	On
	Lesson 2 - A simple topic with a condition an...	(5) Are there any stores aroun...	On
	Lesson 3 - A topic with a condition, variables...	(5) Buy items	On

3. Scroll down the topic design until you find the Condition node where you branched on Purchase Price less than $2000, and the Message node that says “You can return the item(s) by mail to Contoso Returns, 4567 Main St Buffalo, NY 98052.”



Condition

{x} PurchasePriceOfReturn (numb...
is less than
2000
Add condition

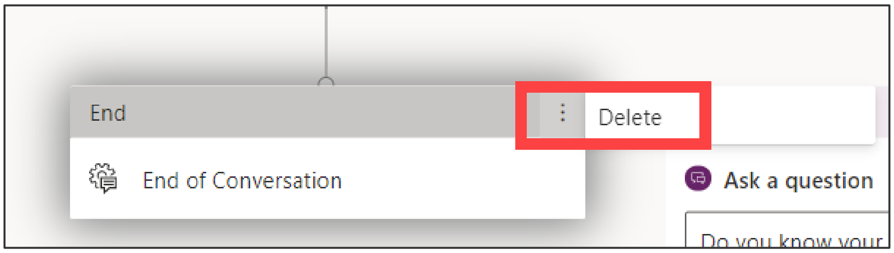
Message

You can return the item(s) to by mail to Contoso Returns, 4567 Main St Buffalo, NY 98052

End

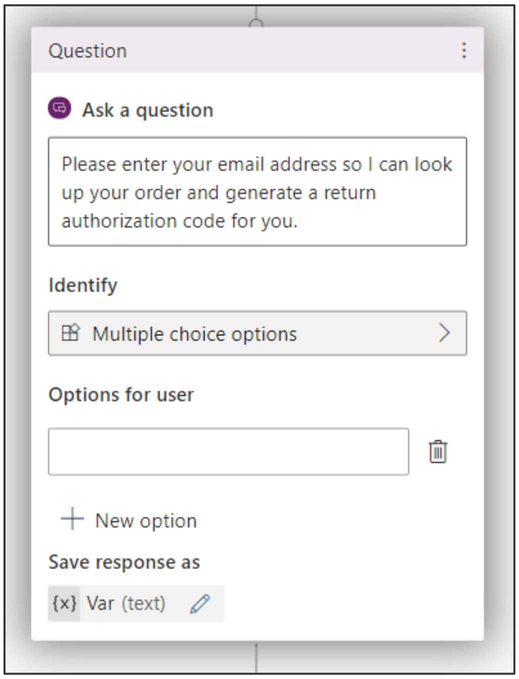
End of Conversation

4. Delete the **End of Conversation** node that follows that Message node, by clicking the Options ⋮ icon in the header and choosing **Delete**.



5. Where you just deleted the End of Conversation node, click the **Add node** + button and select **Ask a question**. Use the question:

   Please enter your email address so I can look up your order and generate a return authorization code for you.



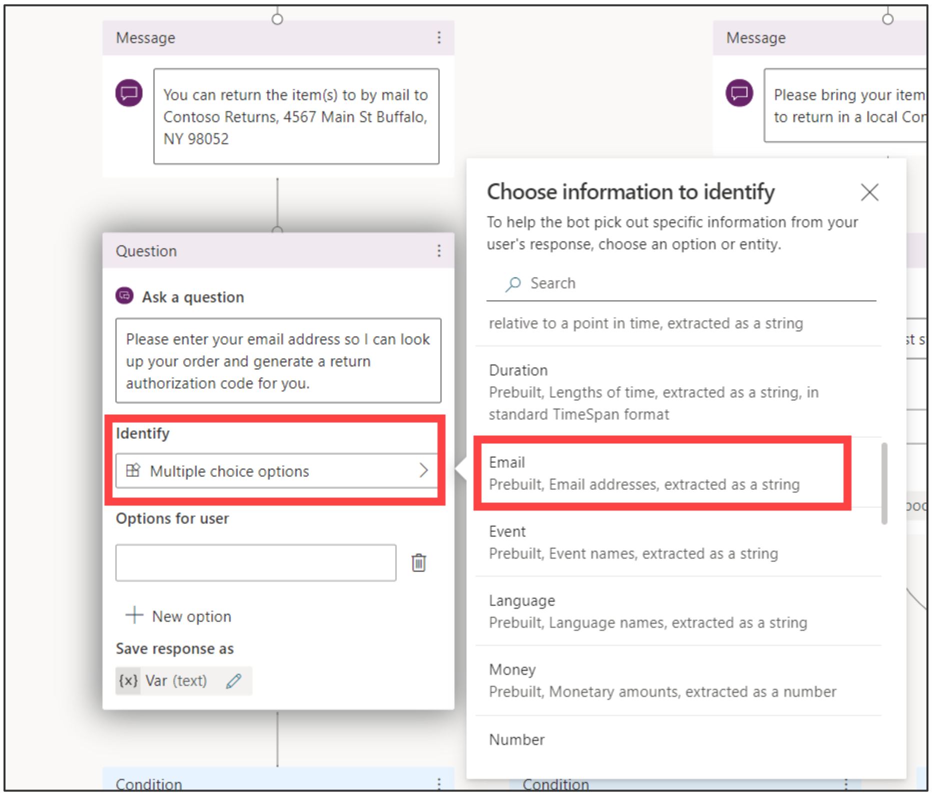
Please enter your email address so I can look up your order and generate a return authorization code for you.

Identify: Multiple choice options

Save response as: {x} Var (text)

6. Click **Identify** field and choose **Email** from the Entities list.

   When you use the Email entity, Power Virtual Agents will ensure that what the user enters is formatted as an email address. If they enter something else, it will ask again until they add a correctly formatted email address.



Message

You can return the item(s) to by mail to Contoso Returns, 4567 Main St Buffalo, NY 98052

Message

Please bring your item to return in a local Co

Question

Ask a question

Please enter your email address so I can look up your order and generate a return authorization code for you.

Identify

Multiple choice options

Options for user

New option

Save response as

{x} Var (text)

Choose information to identify

To help the bot pick out specific information from your user's response, choose an option or entity.

Search

relative to a point in time, extracted as a string

Duration
Prebuilt, Lengths of time, extracted as a string, in standard TimeSpan format

Email
Prebuilt, Email addresses, extracted as a string

Event
Prebuilt, Event names, extracted as a string

Language
Prebuilt, Language names, extracted as a string

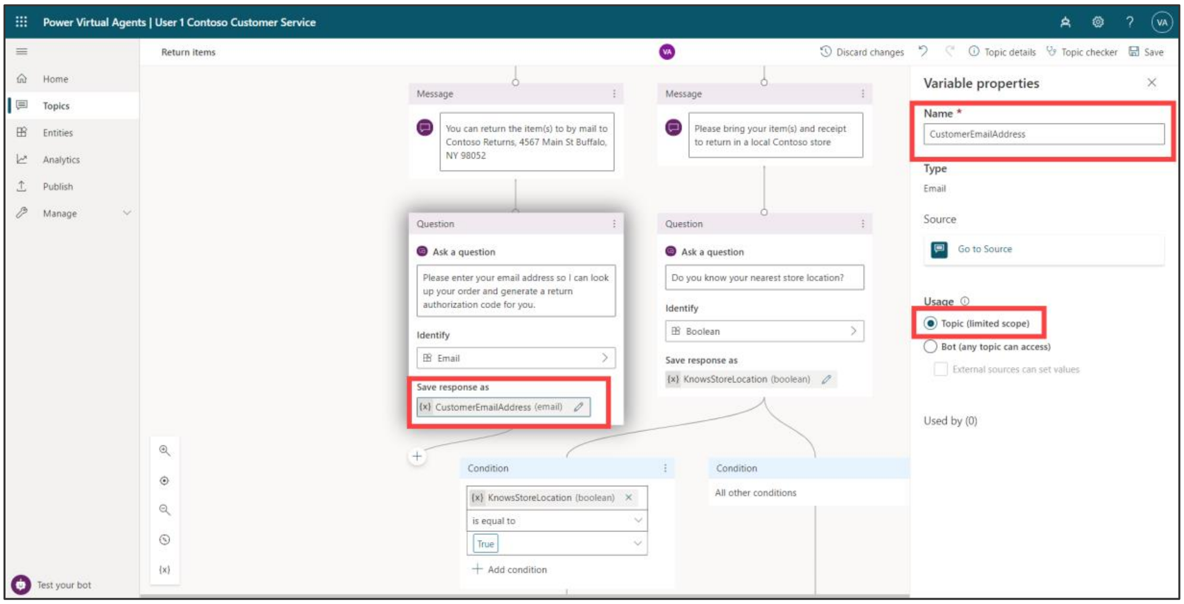
Money
Prebuilt, Monetary amounts, extracted as a number

Number

Condition

Condition

7. In the **Save response as** field, click the variable name **Var (text)**, and rename the variable name from **Var** to **CustomerEmailAddress**. Leave it as a Topic variable. Click **Save** and close the Variable Properties pane.



Message: You can return the item(s) to by mail to Contoso Returns, 4567 Main St Buffalo, NY 98052

Message: Please bring your item(s) and receipt to return in a local Contoso store

Question: Please enter your email address so I can look up your order and generate a return authorization code for you.
Identify: Email
Save response as: {x} CustomerEmailAddress (email)

Question: Do you know your nearest store location?
Identify: Boolean
Save response as: {x} KnowsStoreLocation (boolean)

Condition: {x} KnowsStoreLocation (boolean) is equal to True

Condition: All other conditions

Variable properties
Name *: CustomerEmailAddress
Type: Email
Source
Usage
● Topic (limited scope)
○ Bot (any topic can access)
□ External sources can set values
Used by (0)

In the next task, we'll use the email address in a Power Automate action that looks up their order. Then we'll use another Power Automate flow to generate a code that authorizes the return.

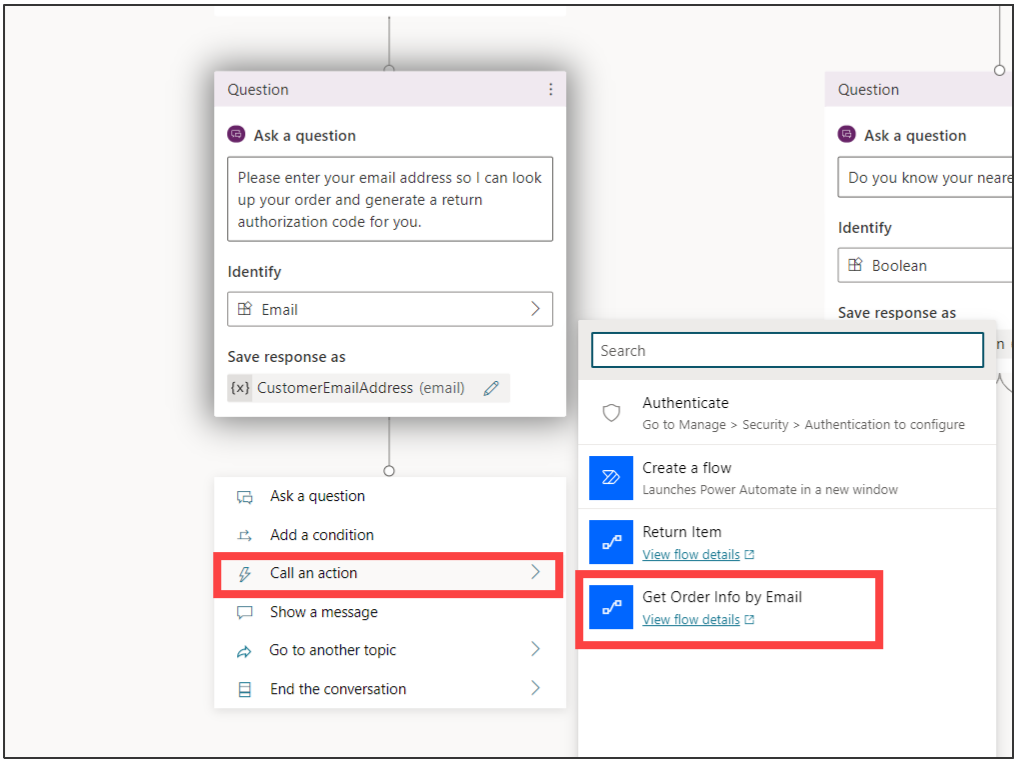
Task 2: Call Get Order Info by Email action

Now, we're going to use the email address to find the order, by adding the **Get Order Info by Email** flow (which you imported into your bot environment during the lab pre-requisites). This Power Automate flow finds the order record associated with the user's email address and returns information about the device order.

1. Click the **Add node** button and select **Call an action**.

2. Choose **Get Order Info by Email** from the action list. (If you don't see this listed, you may need to complete the lab pre-requisites.)

   TIP: Click on the name of the flow, not the View flow details link (that will open the flow in a new tab – if you accidentally click there



Question

Ask a question

Please enter your email address so I can look up your order and generate a return authorization code for you.

Identify: Email

Save response as: {x} CustomerEmailAddress (email)

Question

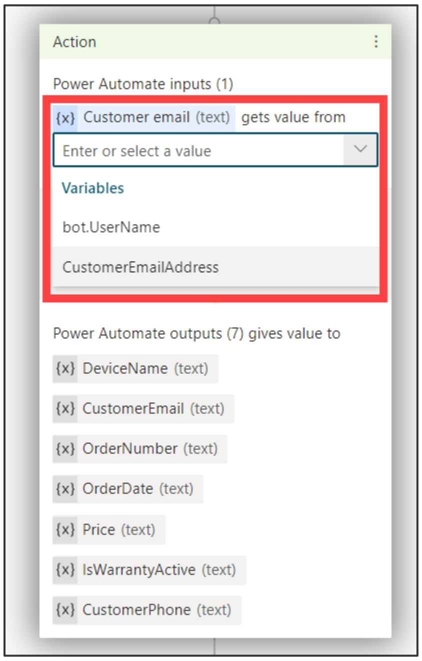
Ask a question

Do you know your neare

Identify: Boolean

Save response as

3. The Power Automate flow tells you what inputs it needs. In the **Power Automate inputs** field, set it so that the flow will receive the value for **Customer email** from the `CustomerEmailAddress` variable you used to capture the customer' email address in Task 1.



The Power Automate flow also tells you what outputs it will send you back as information stored in variables (which are named in the flow definition, not in the Power Virtual Agents topic). You can use those output variables in many ways; in this case you will show them to the user as text.

4. Click the **Add node** + button and select **Show a message.** We will show the order details to the user.

5. Copy and paste the text below into the **Show a message** node.

   Here are your order details:

   •Device name:

   •Email:

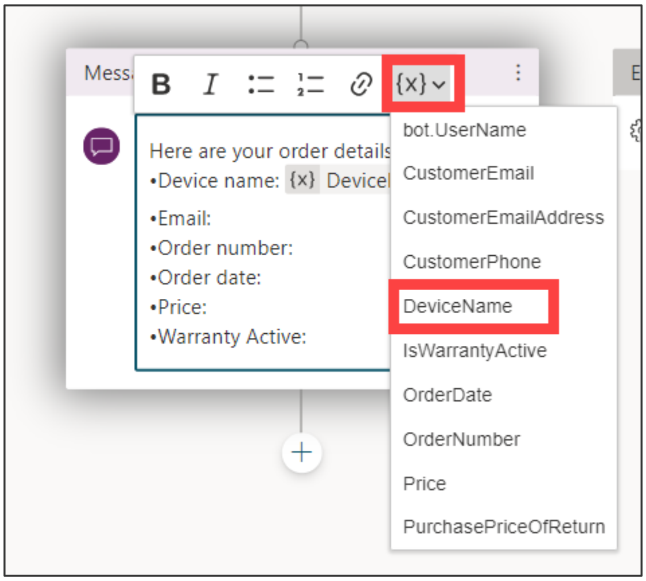
   •Order number:

   •Order date:

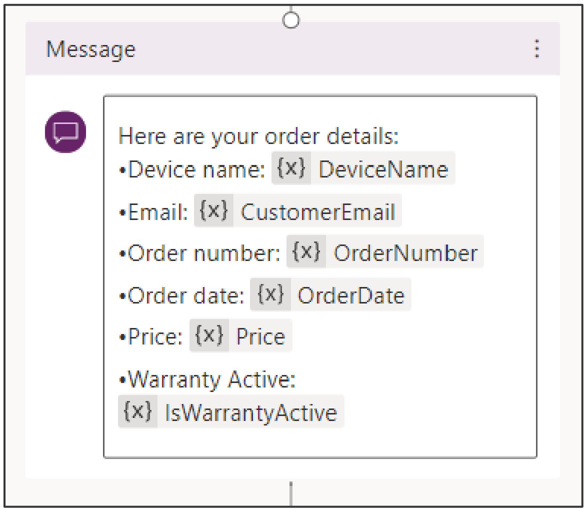
   •Price:

   •Warranty Active:

6. For each line item in the text (such as “Device name:”), click to place your cursor after the colon (“:”), add a space, and select the corresponding variable from the **{x}** drop-down list in the node’s editor bar. (Follow the screenshot below.)

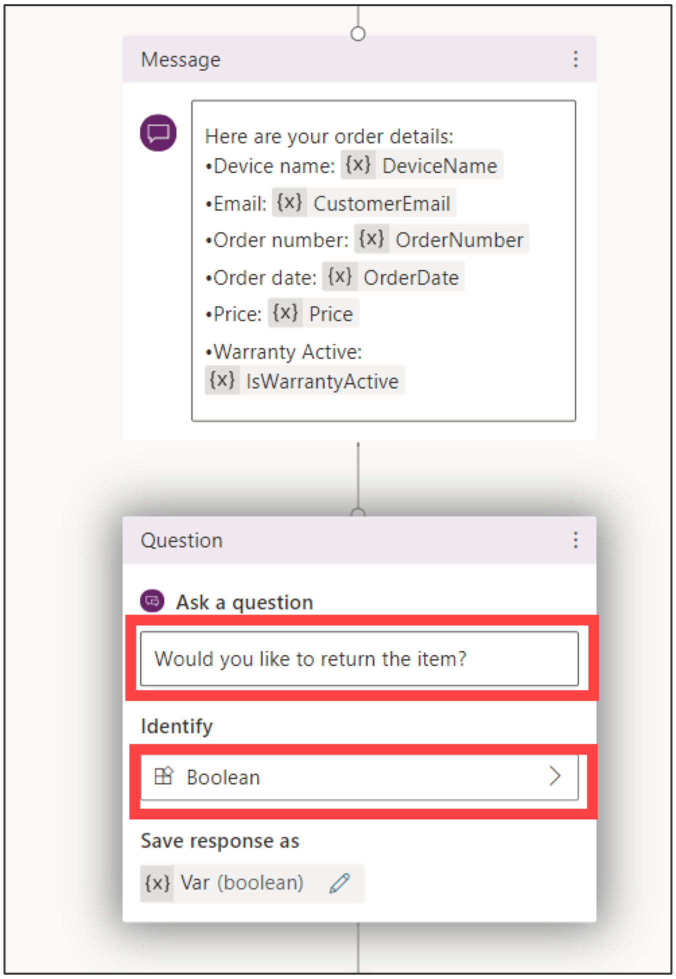


Here are your order details
•Device name: {x} Device
•Email:
•Order number:
•Order date:
•Price:
•Warranty Active:



Here are your order details:
•Device name: {x} DeviceName
•Email: {x} CustomerEmail
•Order number: {x} OrderNumber
•Order date: {x} OrderDate
•Price: {x} Price
•Warranty Active:
{x} IsWarrantyActive

7. Add an **Ask a question** node. Use the question “Would you like to return the item?” and choose **Boolean** as the entity type to identify. (This will show the user “Yes” and “No” buttons in the chat, though you won’t see them in the authoring canvas.)



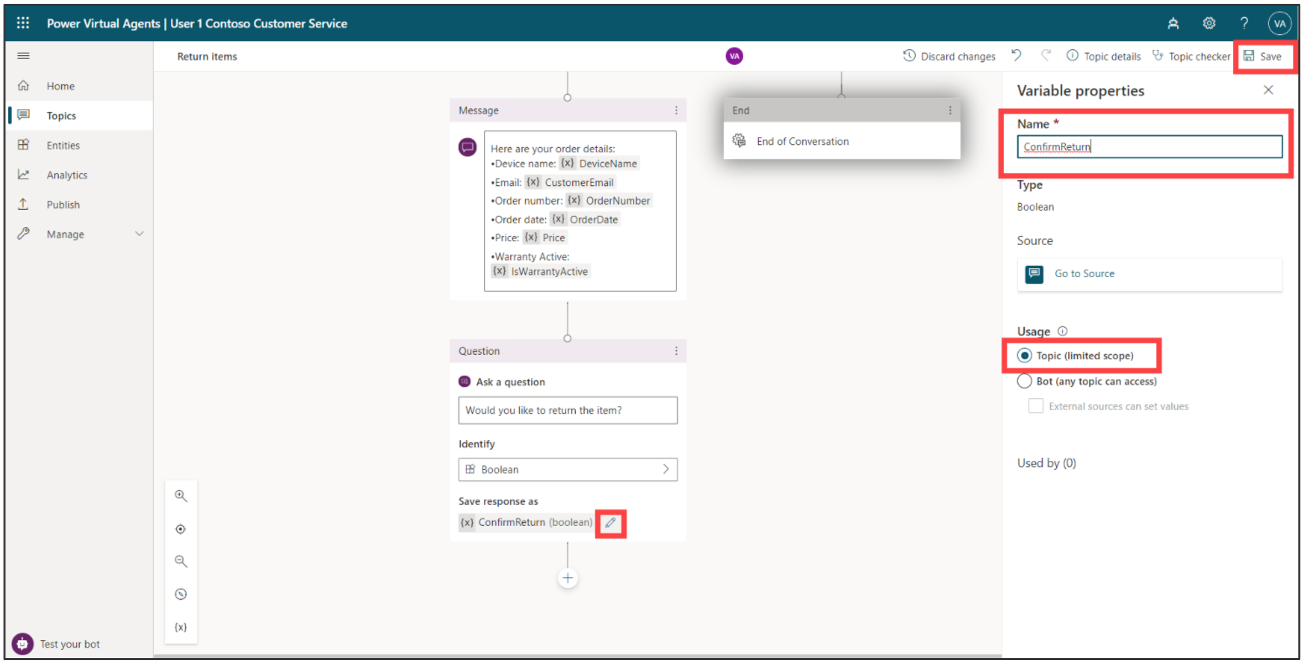
Here are your order details:
•Device name: {x} DeviceName
•Email: {x} CustomerEmail
•Order number: {x} OrderNumber
•Order date: {x} OrderDate
•Price: {x} Price
•Warranty Active:
{x} IsWarrantyActive

Would you like to return the item?

Identify: Boolean

Save response as: {x} Var (boolean)

8. Name the variable **ConfirmReturn**. Leave it as a Topic variable. Click **Save** and close the Variable Properties pane.



Here are your order details:
•Device name: {x} DeviceName
•Email: {x} CustomerEmail
•Order number: {x} OrderNumber
•Order date: {x} OrderDate
•Price: {x} Price
•Warranty Active:
{x} IsWarrantyActive

Would you like to return the item?

Boolean

{x} ConfirmReturn (boolean)

Variable properties

Name *: ConfirmReturn

Type: Boolean

Source: Go to Source

Usage

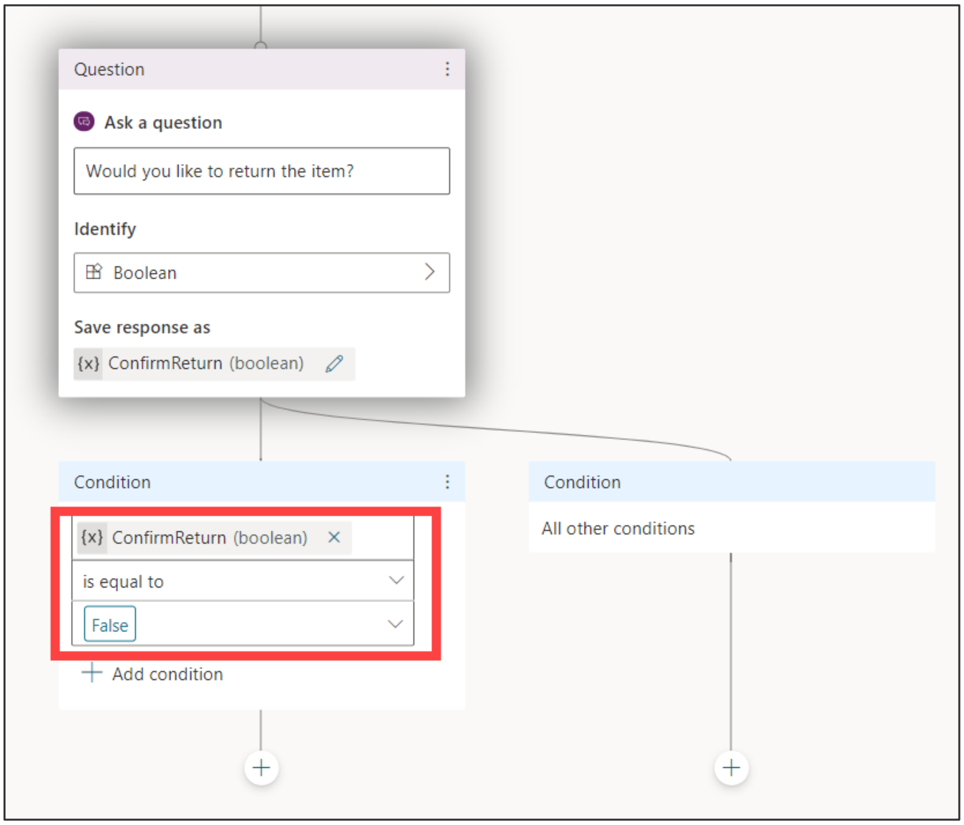
● Topic (limited scope)

○ Bot (any topic can access)

□ External sources can set values

Used by (0)

9. **Add a Condition** to the conversation. In the Conditional branch, set up the node to test for the **ConfirmReturn** variable equal to False. (This branch will be for users who don't want to return their order after all.)



Would you like to return the item?

Boolean

{x} ConfirmReturn (boolean)

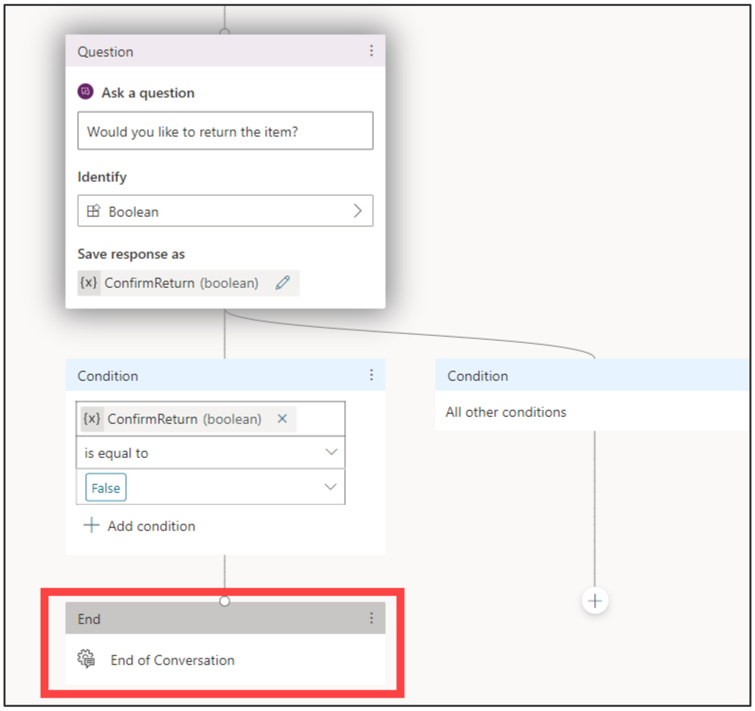
{x} ConfirmReturn (boolean)

is equal to

False

All other conditions

10. Add an **End the conversation > End with survey** node.

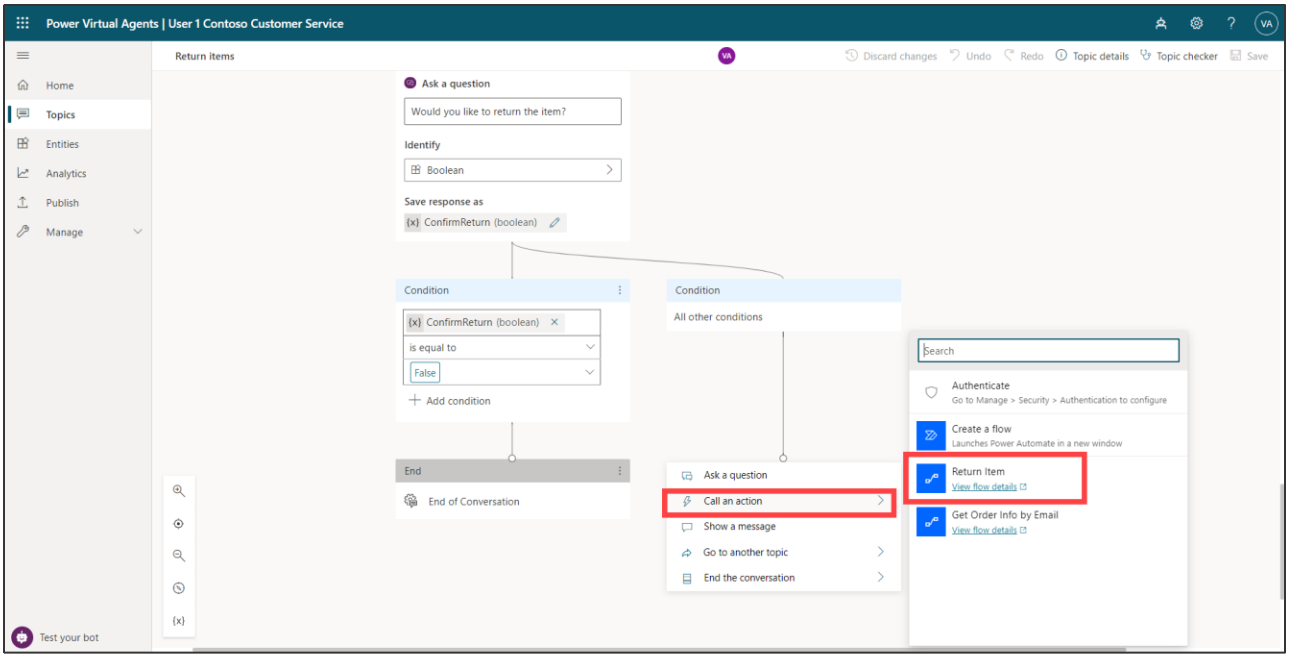


Task 3: Add Return Item action to the conversation

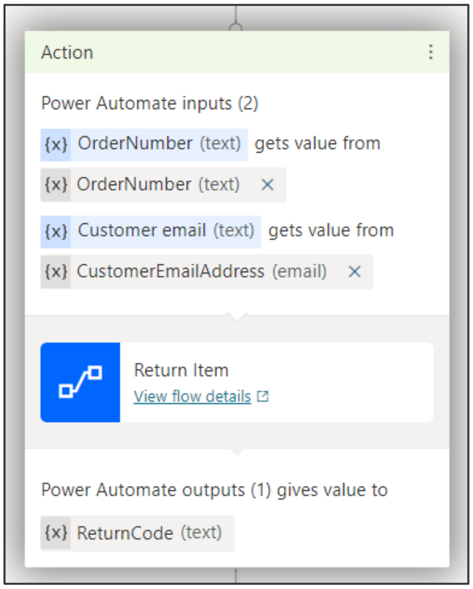
Next, we will add another Power Automate action, to return the device.

1. Under **All other conditions** (which will handle the “yes” response to the “Would you like to return the order?” question), click the **Add node** button and select **Call an action**.

2. From the actions list, select **Return Item**.



3. For **Power Automate inputs**, select the `OrderNumber` and `CustomerEmailAddress` variables. (See the following screenshot.)


Action

Power Automate inputs (2)

{x} OrderNumber (text) gets value from

{x} OrderNumber (text) ✕

{x} Customer email (text) gets value from

{x} CustomerEmailAddress (email) ✕

Return Item
View flow details

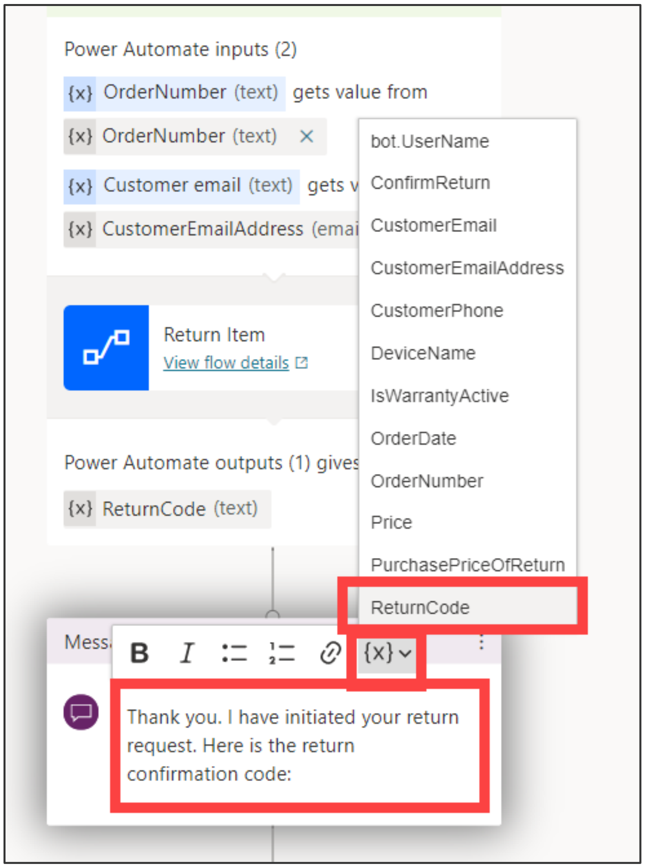
Power Automate outputs (1) gives value to

{x} ReturnCode (text)

4. Add a **Message** node after the action.

5. Enter the text below and insert the `Return Code` variable from Return Item's output:

   Thank you. I have initiated your return request. Here is the return confirmation code: {x}Return Code.



Power Automate inputs (2)

{x} OrderNumber (text) gets value from

{x} OrderNumber (text) ✕

{x} Customer email (text) gets v

{x} CustomerEmailAddress (emai

Return Item

View flow details

Power Automate outputs (1) gives

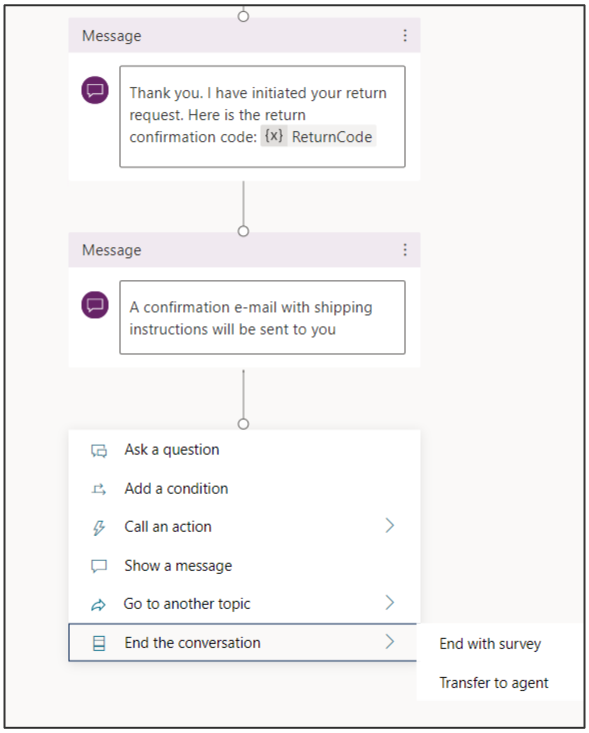
{x} ReturnCode (text)

Thank you. I have initiated your return request. Here is the return confirmation code:

6. Add a new **Message** node with message

   A confirmation e-mail with shipping instructions will be sent to you.

7. Then add an **End the conversation – End with survey**.



Message

Thank you. I have initiated your return request. Here is the return confirmation code: {x} ReturnCode

Message

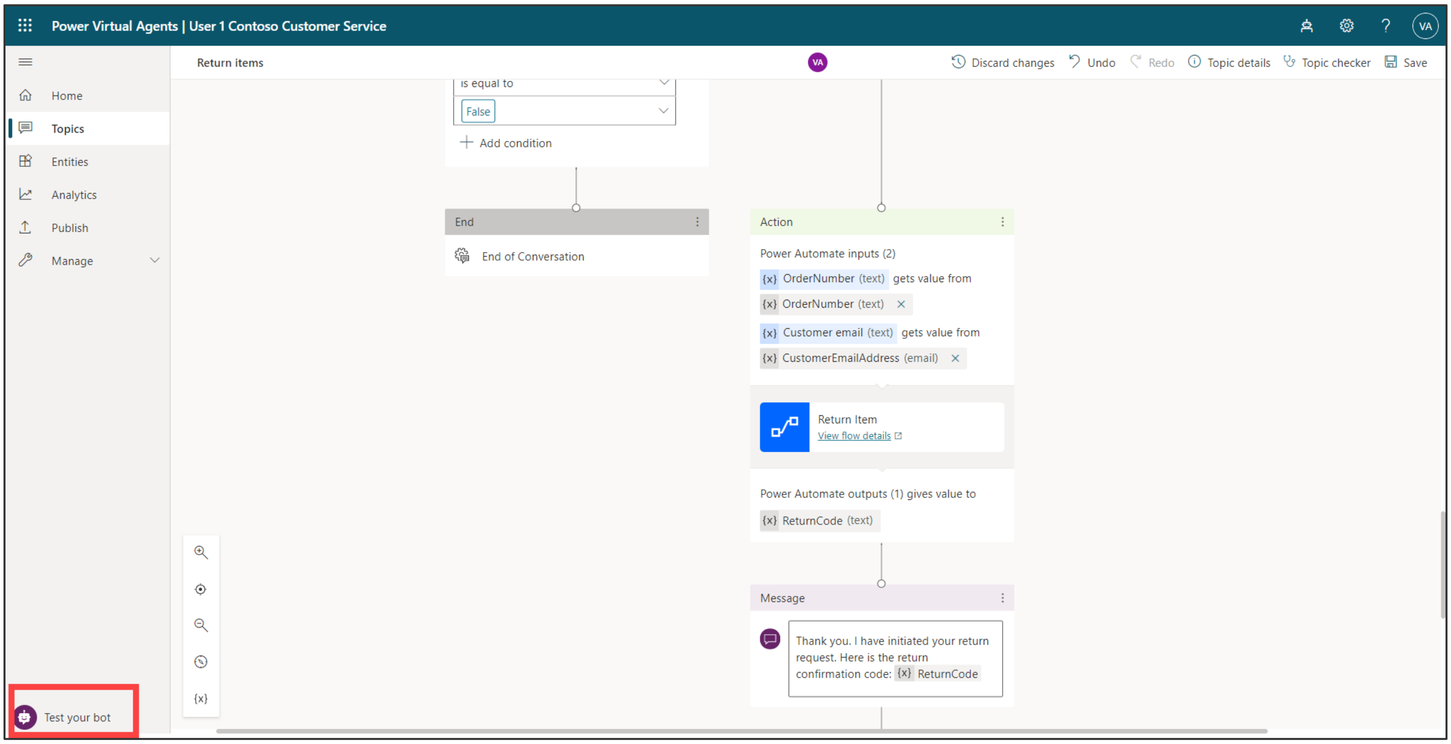
A confirmation e-mail with shipping instructions will be sent to you

8. Save the topic.

   Note: the final message is just an example. You can challenge yourself by building your own Power Automate flow to send emails through your chatbot.

Task 4: Test your conversation

1. Click **Test your bot**.



Power Automate inputs (2)

{x} OrderNumber (text) gets value from {x} OrderNumber (text)

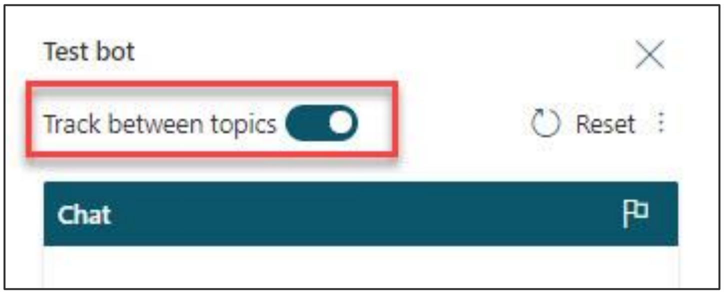
{x} Customer email (text) gets value from {x} CustomerEmailAddress (email)

Return Item
View flow details

Power Automate outputs (1) gives value to {x} ReturnCode (text)

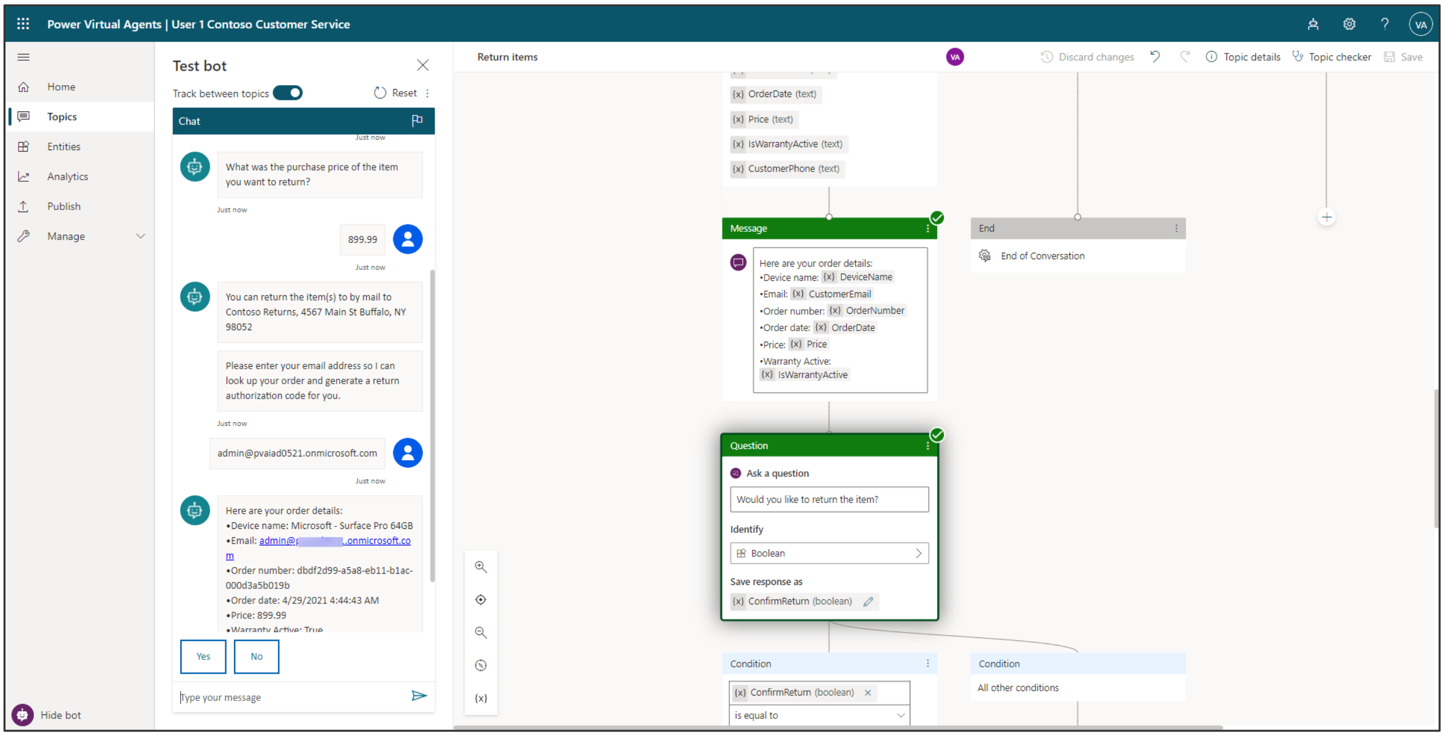
Thank you. I have initiated your return request. Here is the return confirmation code: {x} ReturnCode

2. Turn on the **Track between topics** toggle



3. Tell the bot "**I want to return a product**", then click **Send** ▷.

4. Reply to the bot’s questions about your name and purchase price ($899.99). When the chatbot asks for your email address, provide the email address you are using for the labs. The bot calls your Power Automate flow, and returns details of your order. (Your instructor will have set this data up for you before the labs)



What was the purchase price of the item you want to return?

899.99

You can return the item(s) to by mail to Contoso Returns, 4567 Main St Buffalo, NY 98052

Please enter your email address so I can look up your order and generate a return authorization code for you.

admin@pvaiad0521.onmicrosoft.com

Here are your order details:
•Device name: Microsoft - Surface Pro 64GB
•Email: admin@p[redacted].onmicrosoft.com
•Order number: dbdf2d99-a5a8-eb11-b1ac-000d3a5b019b
•Order date: 4/29/2021 4:44:43 AM
•Price: 899.99
•Warranty Active: True

{x} OrderDate (text)
{x} Price (text)
{x} IsWarrantyActive (text)
{x} CustomerPhone (text)

Message
Here are your order details:
•Device name: {x} DeviceName
•Email: {x} CustomerEmail
•Order number: {x} OrderNumber
•Order date: {x} OrderDate
•Price: {x} Price
•Warranty Active: {x} IsWarrantyActive

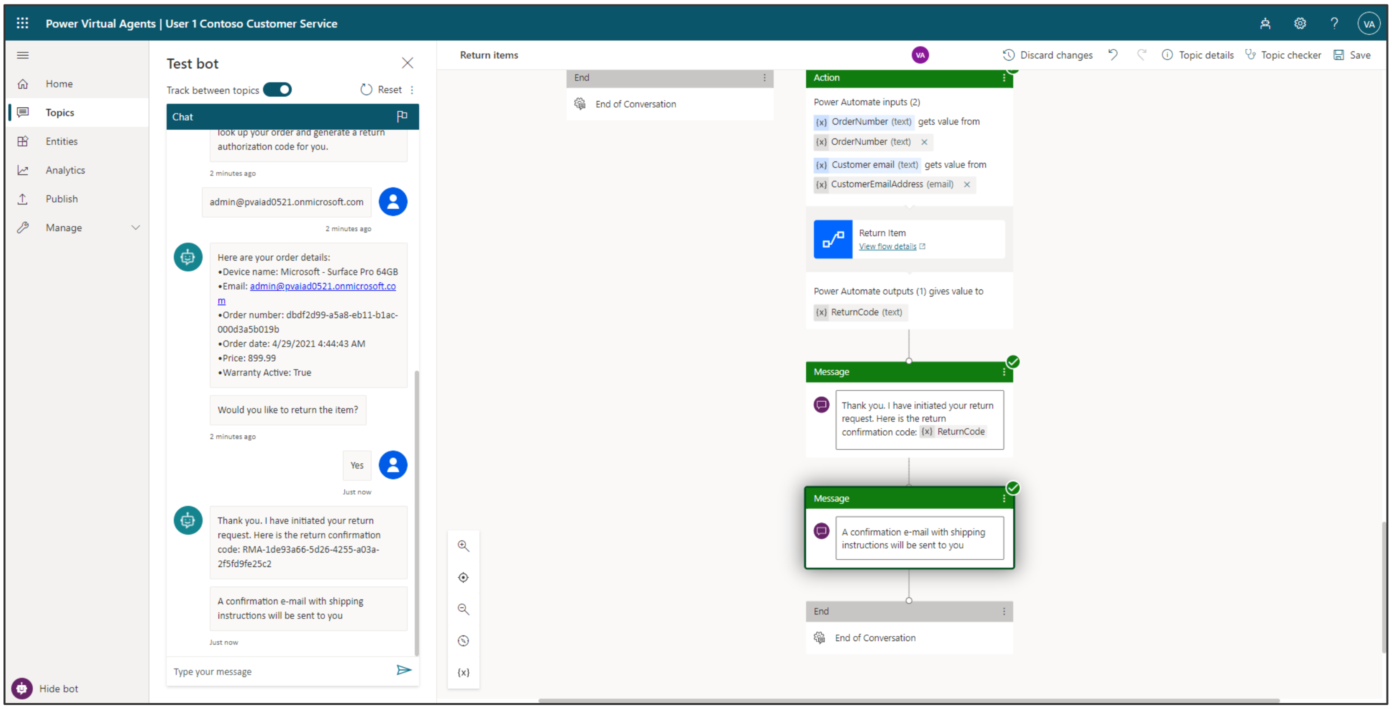
End
End of Conversation

Question
Ask a question
Would you like to return the item?
Identify
Boolean
Save response as
{x} ConfirmReturn (boolean)

Condition
{x} ConfirmReturn (boolean)
is equal to

Condition
All other conditions

5. Answer “**Yes**” to the question “Would you like to return the item?” The chatbot generates a return confirmation code and replies with the code.



look up your order and generate a return authorization code for you.

admin@pvaiad0521.onmicrosoft.com

Here are your order details:
•Device name: Microsoft - Surface Pro 64GB
•Email: admin@pvaiad0521.onmicrosoft.com
•Order number: dbdf2d99-a5a8-eb11-b1ac-000d3a5b019b
•Order date: 4/29/2021 4:44:43 AM
•Price: 899.99
•Warranty Active: True

Would you like to return the item?

Yes

Thank you. I have initiated your return request. Here is the return confirmation code: RMA-1de93a66-5d26-4255-a03a-2f5fd9fe25c2

A confirmation e-mail with shipping instructions will be sent to you

End
End of Conversation

Action
Power Automate inputs (2)
{x} OrderNumber (text) gets value from {x} OrderNumber (text)
{x} Customer email (text) gets value from {x} CustomerEmailAddress (email)
Return Item
View flow details
Power Automate outputs (1) gives value to {x} ReturnCode (text)

Message
Thank you. I have initiated your return request. Here is the return confirmation code: {x} ReturnCode

Message
A confirmation e-mail with shipping instructions will be sent to you

End
End of Conversation

6. Finish the conversation by responding to the survey.

Congratulations -you have now created a bot which can submit and retrieve information from a database on behalf of the user. In the next lab you will build your own flow.

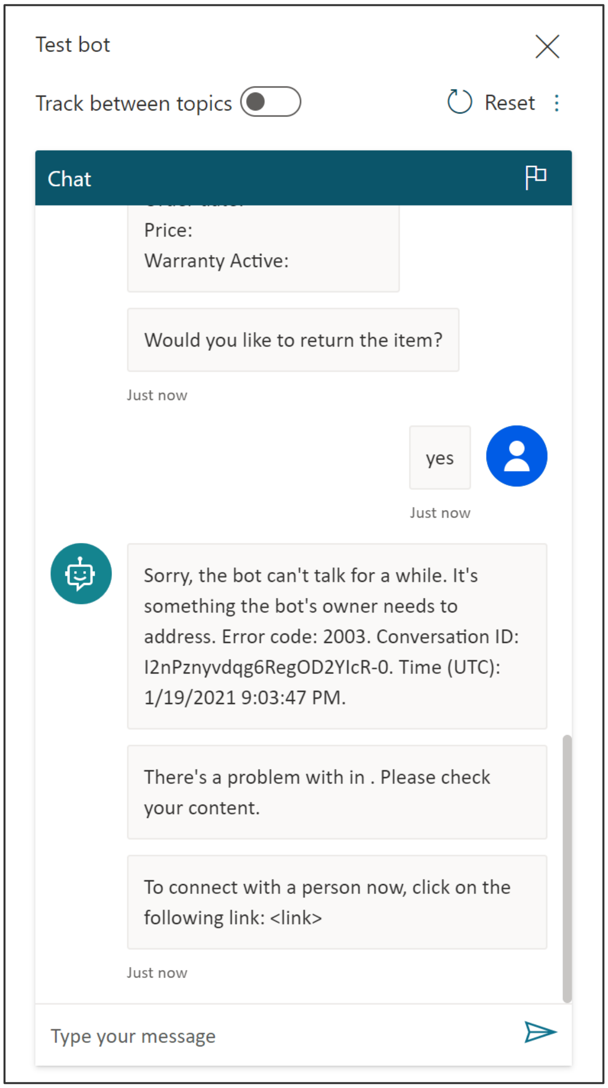
Best Practices for Flows in your Power Virtual Agent

Note: There is no lab work below. The following has been provided as information for your further learning. Please do not edit the existing flow for Return Items used below as other users in the training tenant will be using this for their lab.

Handling Exceptions

There can be instances where a flow in your Power Virtual Agents bot does not run successfully. For example, the person interacting with the bot might provide an incorrect value, which your flow is not expecting to handle. When a flow is run unsuccessfully during a conversation, the end users will be presented with an error code.

If you consider the Return Item flow used in this lab, the flow is expecting to update a record based on an existing Order Number stored in the system (Microsoft Dataverse). If the email address the customer provides cannot be found in Dataverse (i.e. there is no associated Order Number with the email address), their chat will look like this:



Price:
Warranty Active:

Would you like to return the item?

Just now

yes

Just now

Sorry, the bot can't talk for a while. It's something the bot's owner needs to address. Error code: 2003. Conversation ID: I2nPznyvdqg6RegOD2YIcR-0. Time (UTC): 1/19/2021 9:03:47 PM.

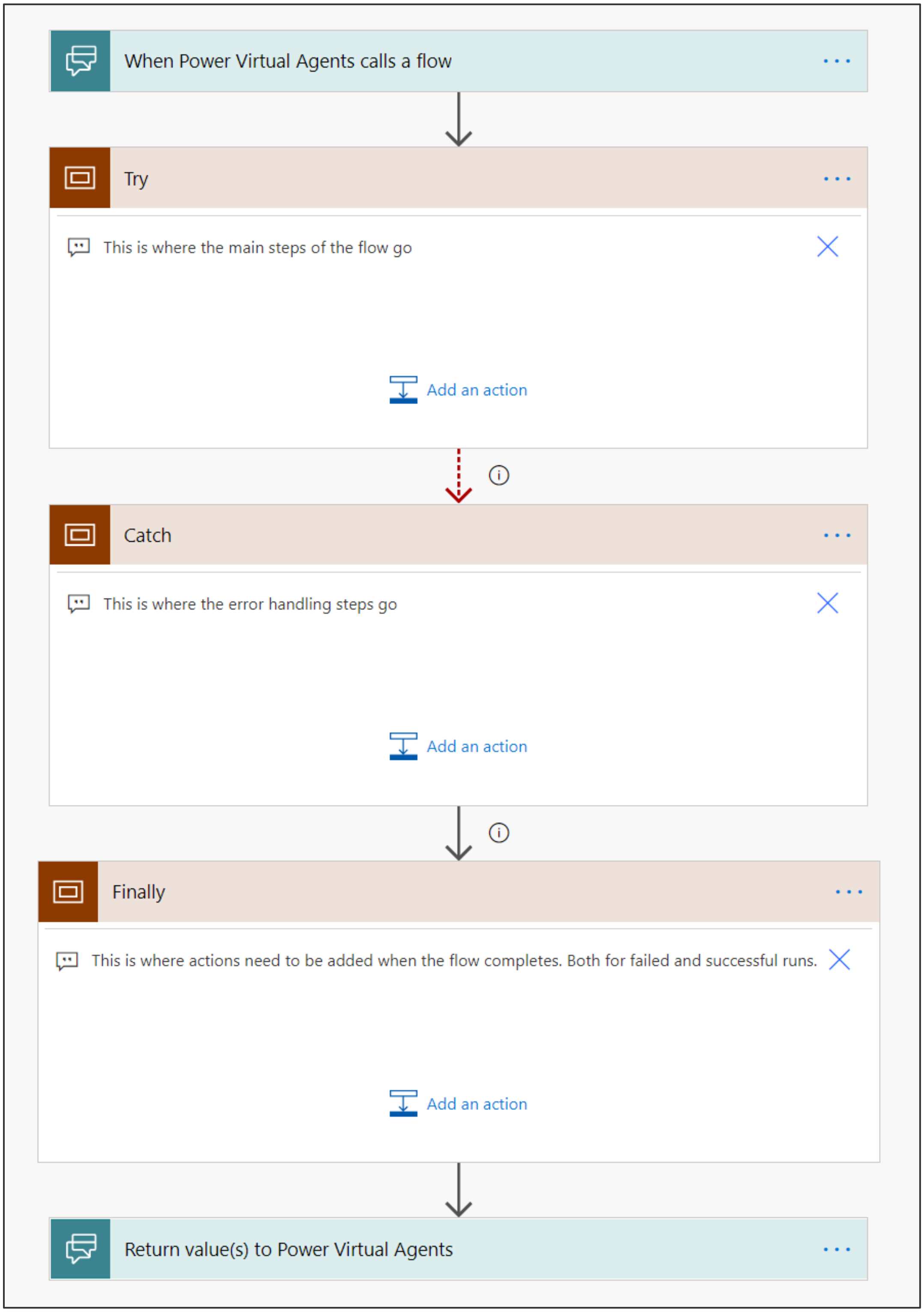
There's a problem with in . Please check your content.

To connect with a person now, click on the following link: <link>

Just now

To avoid such situations, it is best practice to build your Power Automate flow so that it can handle exceptions. If the flow itself handles exceptions, the customer/end user can be provided with a user-friendly message rather than an error code.

You can use the **Try, Catch and Finally** pattern in your flow to handle errors and exceptions.



When Power Virtual Agents calls a flow

Try

This is where the main steps of the flow go

Catch

This is where the error handling steps go

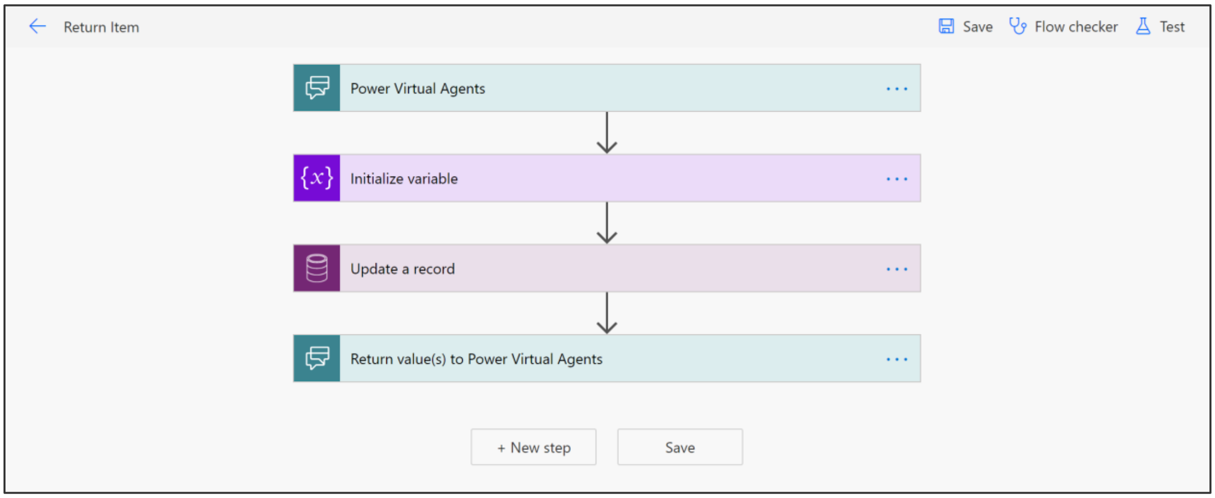
Finally

This is where actions need to be added when the flow completes. Both for failed and successful runs.

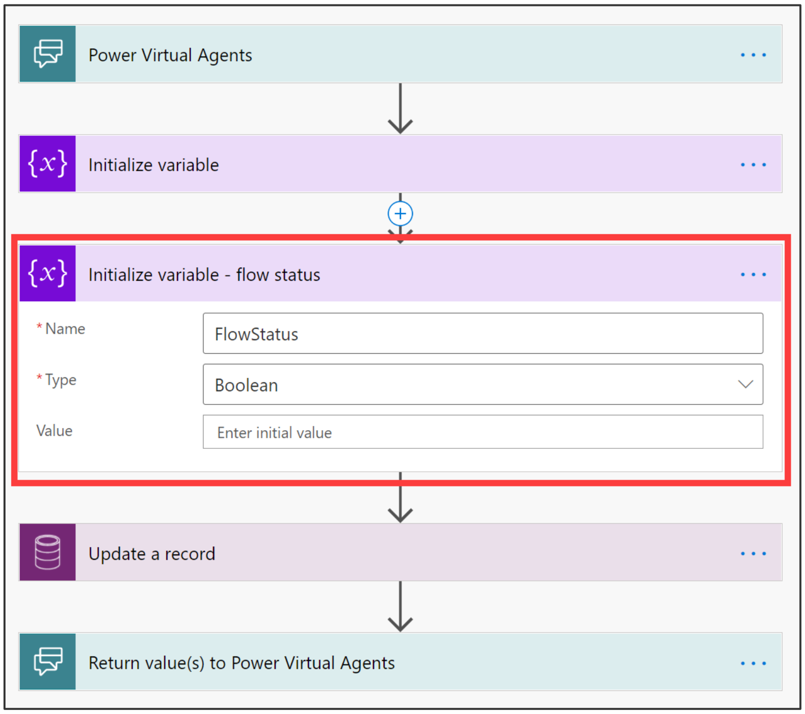
Return value(s) to Power Virtual Agents

In the case for the Return Item flow, the details below will show you how the **Try and Catch** patterns can be used to avoid errors returning to the end user of the bot.

This is what the Return Item flow looks like.



We will create another variable for our "Flow Status". This will later be returned to the bot conversation. The flow status will allow you to customize a message presented to the end user, whether the flow ran successfully or failed.


Power Virtual Agents
Initialize variable
Initialize variable - flow status
Update a record
Return value(s) to Power Virtual Agents

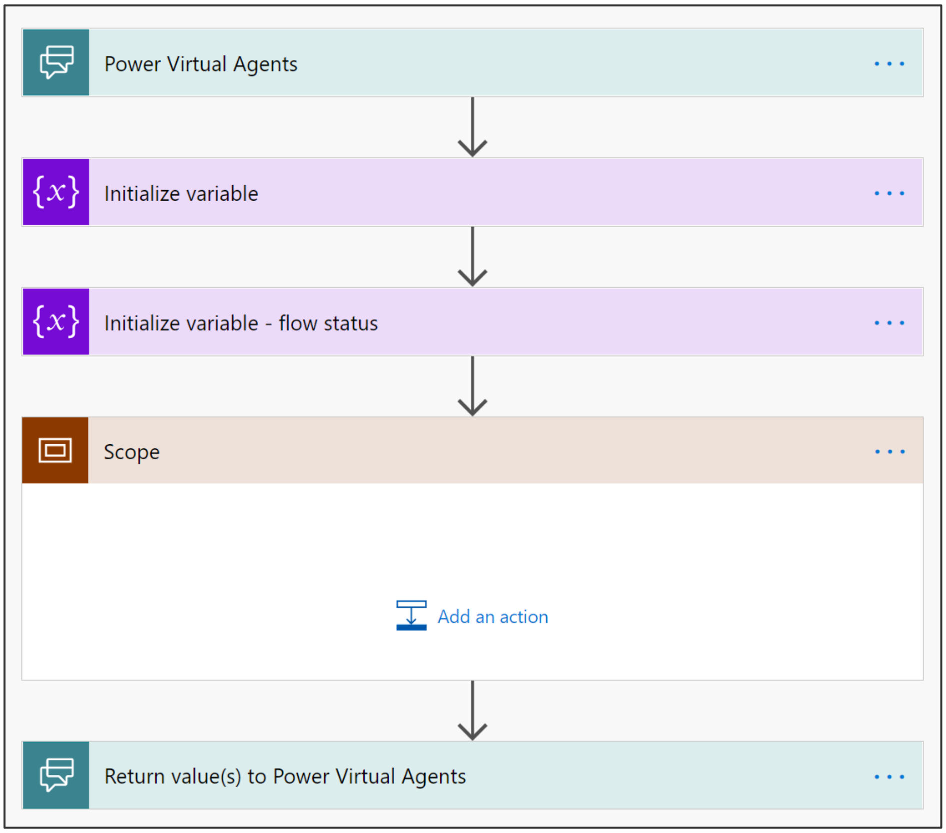
Initialize variable - flow status

* Name: FlowStatus

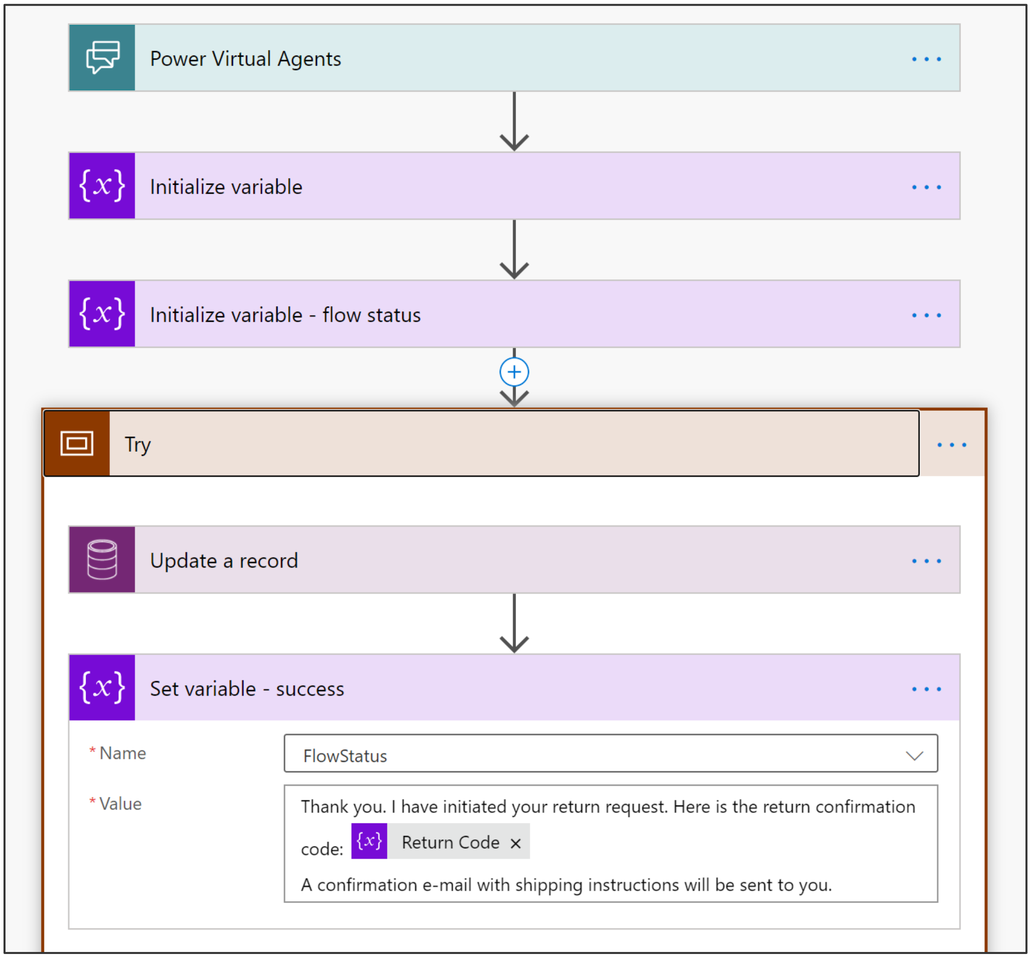
* Type: Boolean

Value: Enter initial value

Next, we will create Scopes to group together the actions that occur if the flow is successful (Try) or if it fails (Catch). A scope is a type of action you can add in your flow.


Power Virtual Agents
Initialize variable
Initialize variable - flow status
Scope
Add an action
Return value(s) to Power Virtual Agents

Within the Try scope, we will add the actions that should be performed on a successful run of the flow. Within the scope, we can set a variable for a custom message sent back to the bot if the flow is run successfully.

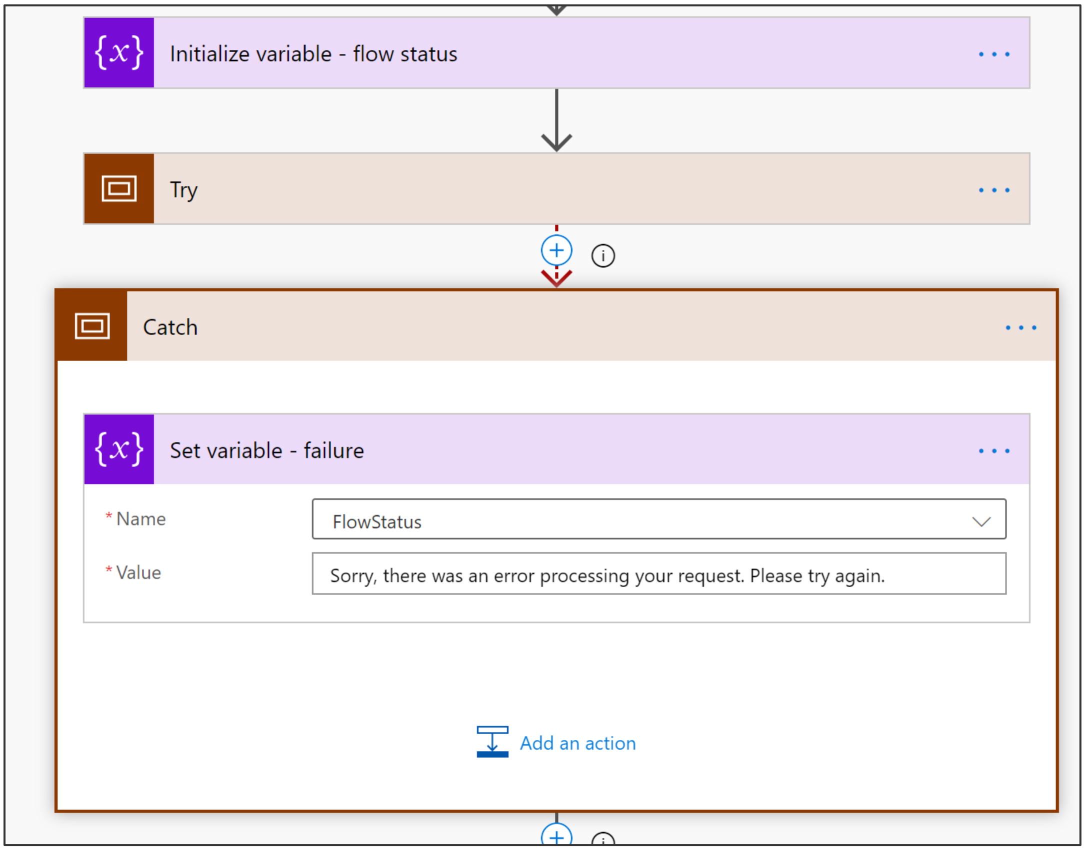


* Name: FlowStatus

* Value: Thank you. I have initiated your return request. Here is the return confirmation code: Return Code

A confirmation e-mail with shipping instructions will be sent to you.

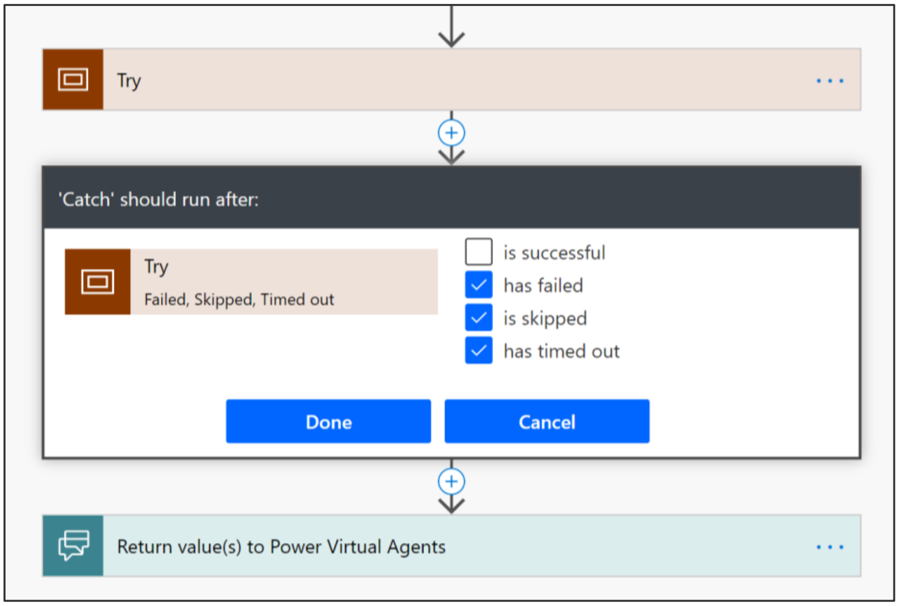
We can then create another scope for Catch. These are the actions that should occur if the actions in the Try scope are unsuccessful. In this case, a variable is set to display a custom user-friendly message back to the bot.



* Name: FlowStatus

* Value: Sorry, there was an error processing your request. Please try again.

We can configure the **run after** settings for the Catch scope. This is to specify that the Catch scope actions should only run if the Try actions fail.

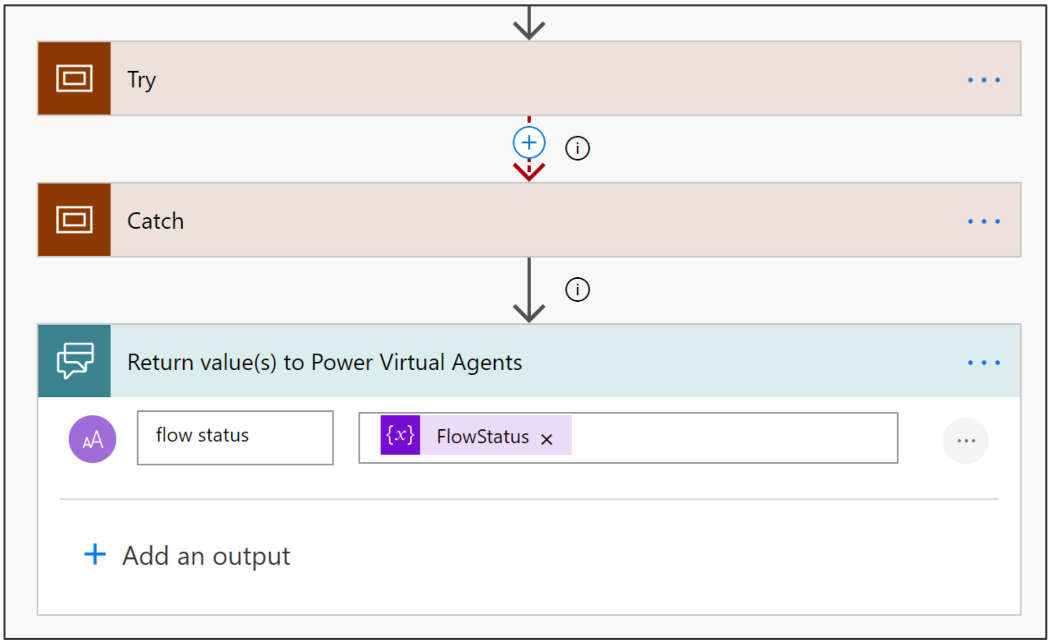


'Catch' should run after:

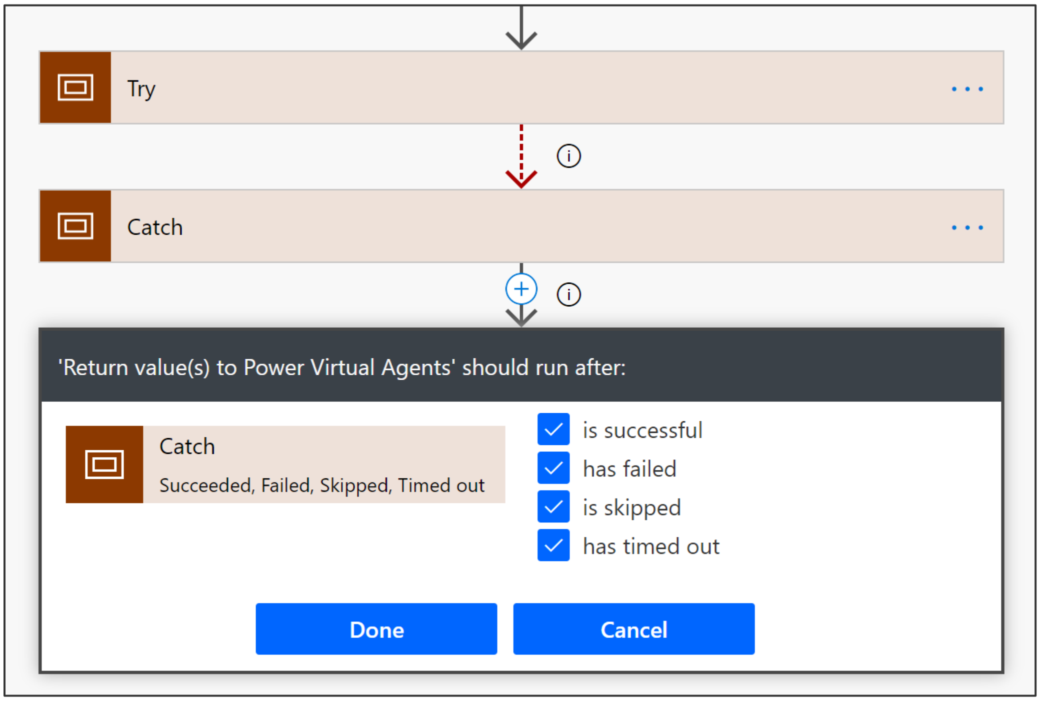
Try
Failed, Skipped, Timed out

☐ is successful
☑ has failed
☑ is skipped
☑ has timed out

Lastly, we want the variables (user friendly messages) created in the Try and Catch scopes to be returned to the bot during the conversation. We can add the variable as a value returned to Power Virtual Agents. The run after settings for the Return Value(s) to Power Virtual Agents should be set to run regardless of flow success or failure.



flow status: FlowStatus

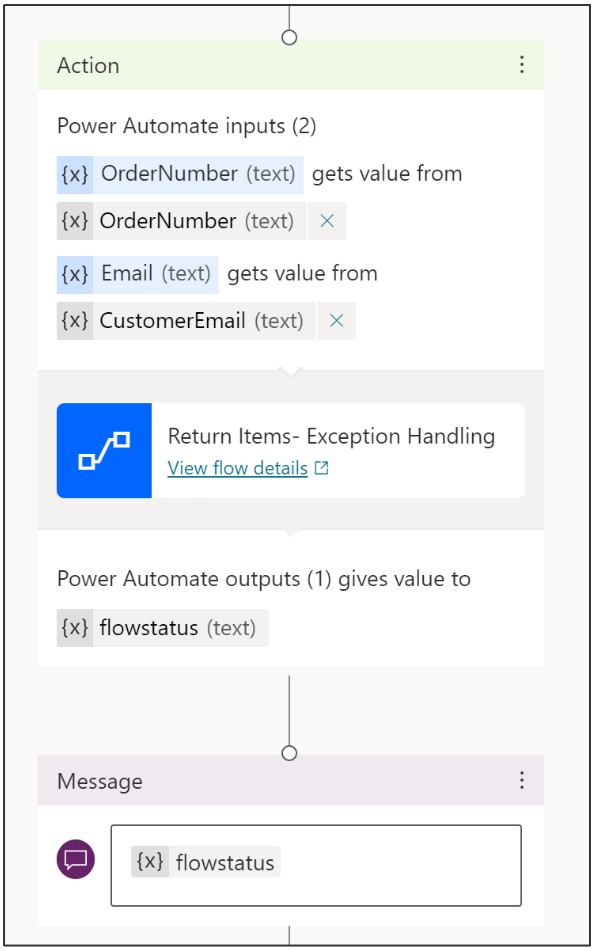


'Return value(s) to Power Virtual Agents' should run after:

Catch
Succeeded, Failed, Skipped, Timed out

☑ is successful
☑ has failed
☑ is skipped
☑ has timed out

Back in the bot conversation authoring canvas, the messages can be edited to display the variable message created in the flow.



Action

Power Automate inputs (2)

{x} OrderNumber (text) gets value from

{x} OrderNumber (text)

{x} Email (text) gets value from

{x} CustomerEmail (text)

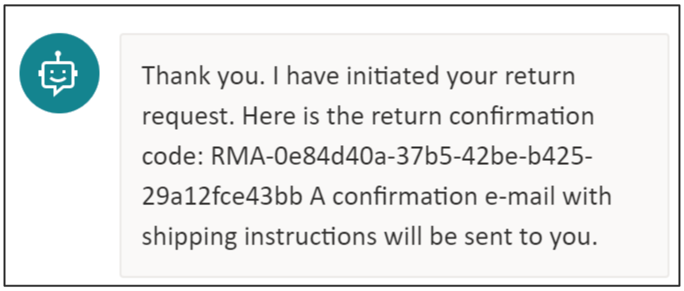
Return Items- Exception Handling
View flow details

Power Automate outputs (1) gives value to

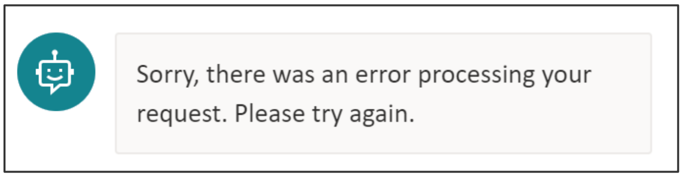
{x} flowstatus (text)

Message

{x} flowstatus



Thank you. I have initiated your return request. Here is the return confirmation code: RMA-0e84d40a-37b5-42be-b425-29a12fce43bb A confirmation e-mail with shipping instructions will be sent to you.

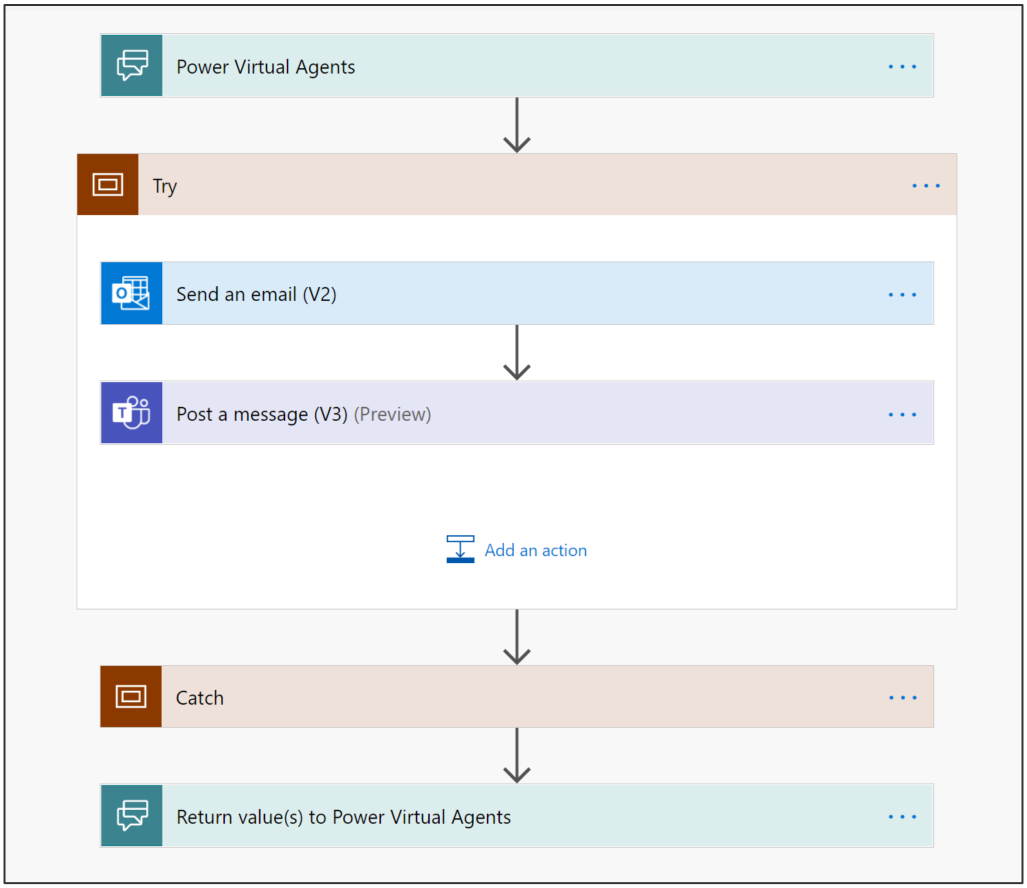


Sorry, there was an error processing your request. Please try again.

As the flow has run successfully, regardless of whether or not there was an issue processing the request, the end user will no longer be presented with an error code.

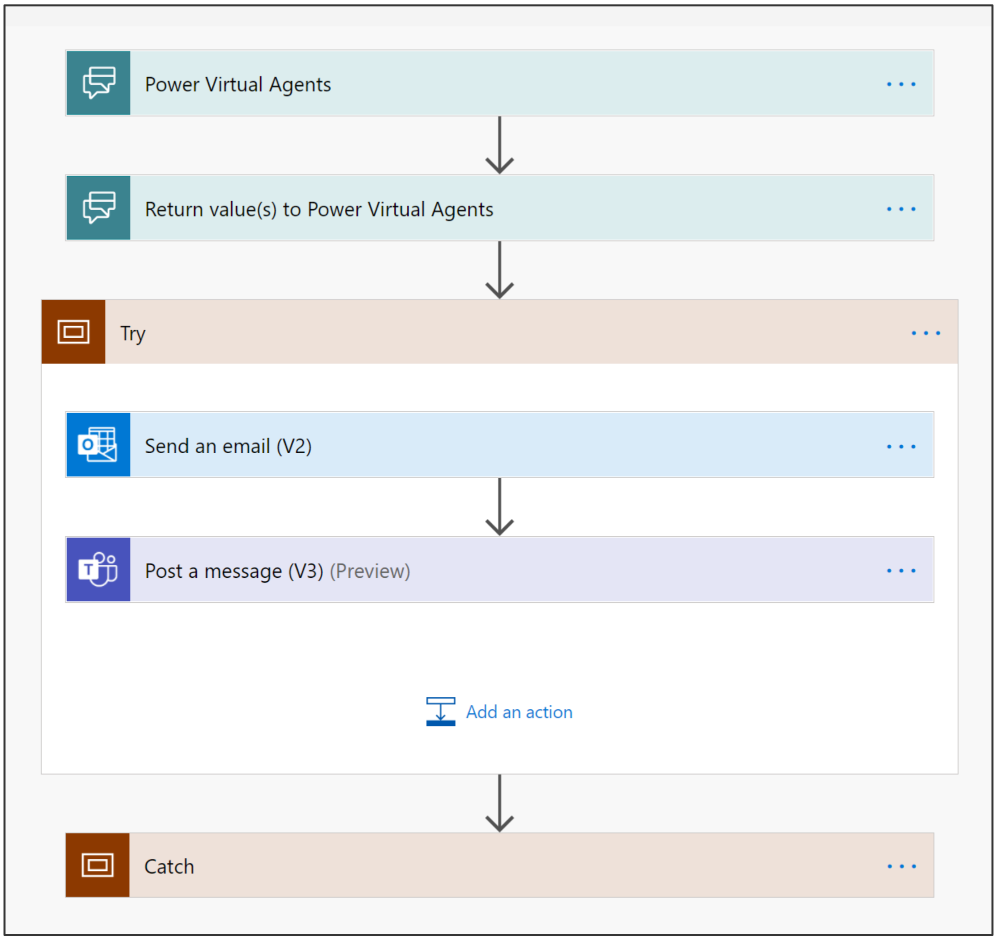
Minimize end user wait time

Another best practice for flows with you Power Virtual Agent is to minimize the time an end user has to wait for a flow to be performed. For example, you might have a flow in your bot conversation where the user can initiate an email and Teams message to be sent. Typically, your flow might look something like this:


Power Virtual Agents
Try
Send an email (V2)
Post a message (V3) (Preview)
Add an action
Catch
Return value(s) to Power Virtual Agents

According to the flow above, the end user would have to wait until the email is sent, Teams message is posted, and for any exception handling steps to occur before they get a response back from the bot. It is a better experience for the end user if they get a response from the bot as soon as possible. For example, if they flow took 1 minute to finish running, the end user would be on the other end waiting for a response.

To speed up the time taken to send a response back to the end user, it is recommended for the flow to perform the Return value(s) to Power Virtual Agents step as soon as possible. This way, the end user can continue their conversation with the bot sooner, while the remaining actions in the flow are still running in the background.


Power Virtual Agents
Return value(s) to Power Virtual Agents
Try
Send an email (V2)
Post a message (V3) (Preview)
Add an action
Catch

Lab survey

We would appreciate your feedback on Power Virtual Agents and on this hands-on-lab, such as the quality of documentation and the usefulness of the learning experience.

Please use the survey at https://aka.ms/PVAiaDSurvey to share your feedback.

You may provide feedback for each module as you complete it or at the end once you've completed all the modules. Thank you!

Terms of Use

By using this document, in whole or in part, you agree to the following terms:

Notice

Information and views expressed in this document, including (without limitation) URL and other Internet Web site references, may change without notice. Examples depicted herein, if any, are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred. This document does not provide you with any legal rights to any intellectual property in any Microsoft product.

Use Limitations

Copying or reproduction, in whole or in part, of this document to any other server or location for further reproduction or redistribution is expressly prohibited. Microsoft provides you with this document for purposes of obtaining your suggestions, comments, input, ideas, or know-how, in any form, ("Feedback") and to provide you with a learning experience. You may use this document only to evaluate its content and provide feedback to Microsoft. You may not use this document for any other purpose. You may not modify, copy, distribute, transmit, display, perform, reproduce, publish, license, create derivative works from, transfer, or sell this document or any portion thereof. You may copy and use this document for your internal, reference purposes only.

Feedback

If you give Microsoft any Feedback about this document or the subject matter herein (including, without limitation, any technology, features, functionality, and/or concepts), you give to Microsoft, without charge, the right to use, share, and freely commercialize Feedback in any way and for any purpose. You also give third parties, without charge, the right to use, or interface with, any Microsoft products or services that include the Feedback. You represent and warrant that you own or otherwise control all rights to such Feedback and that no such Feedback is subject to any third-party rights.

DISCLAIMERS

CERTAIN SOFTWARE, TECHNOLOGY, PRODUCTS, FEATURES, AND FUNCTIONALITY (COLLECTIVELY "CONCEPTS"), INCLUDING POTENTIAL NEW CONCEPTS, REFERENCED IN THIS DOCUMENT ARE IN A SIMULATED ENVIRONMENT WITHOUT COMPLEX SET-UP OR INSTALLATION AND ARE INTENDED FOR FEEDBACK AND TRAINING PURPOSES ONLY. THE CONCEPTS REPRESENTED IN THIS DOCUMENT MAY NOT REPRESENT FULL FEATURE CONCEPTS AND MAY NOT WORK THE WAY A FINAL VERSION MAY WORK. MICROSOFT ALSO MAY NOT RELEASE A FINAL VERSION OF SUCH CONCEPTS. YOUR EXPERIENCE WITH USING SUCH CONCEPTS IN A PHYSICAL ENVIRONMENT MAY ALSO BE DIFFERENT.

THIS DOCUMENT, AND THE CONCEPTS AND TRAINING PROVIDED HEREIN, IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, WHETHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING (WITHOUT LIMITATION) THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND NONINFRINGEMENT. MICROSOFT DOES NOT MAKE ANY ASSURANCES OR REPRESENTATIONS WITH REGARD TO THE ACCURACY OF THE RESULTS, THE OUTPUT THAT DERIVES FROM USE OF THIS DOCUMENT OR THE CONCEPTS, OR THE SUITABILITY OF THE CONCEPTS OR INFORMATION CONTAINED IN THIS DOCUMENT FOR ANY PURPOSE.

MICROSOFT POWER VIRTUAL AGENTS (1) IS NOT INTENDED OR MADE AVAILABLE AS A MEDICAL DEVICE FOR THE DIAGNOSIS OF DISEASE OR OTHER CONDITIONS, OR IN THE CURE, MITIGATION, TREATMENT OR PREVENTION OF DISEASE, OR OTHERWISE TO BE USED AS A COMPONENT OF ANY CLINICAL OFFERING OR PRODUCT, AND NO LICENSE OR RIGHT IS GRANTED TO USE MICROSOFT POWER VIRTUAL AGENTS FOR SUCH PURPOSES, (2) IS NOT DESIGNED OR INTENDED TO BE A SUBSTITUTE FOR PROFESSIONAL MEDICAL ADVICE, DIAGNOSIS, TREATMENT, OR JUDGMENT AND SHOULD NOT BE USED AS A SUBSTITUTE FOR, OR TO REPLACE, PROFESSIONAL MEDICAL ADVICE, DIAGNOSIS, TREATMENT, OR JUDGMENT, AND (3) SHOULD NOT BE USED FOR EMERGENCIES AND DOES NOT SUPPORT EMERGENCY CALLS. ANY CHATBOT YOU CREATE USING MICROSOFT POWER VIRTUAL AGENTS IS YOUR OWN PRODUCT OR SERVICE, SEPARATE AND APART FROM MICROSOFT POWER VIRTUAL AGENTS. YOU ARE SOLELY RESPONSIBLE FOR THE DESIGN, DEVELOPMENT, AND IMPLEMENTATION OF YOUR CHATBOT (INCLUDING INCORPORATION OF IT INTO ANY PRODUCT OR SERVICE INTENDED FOR MEDICAL OR CLINICAL USE) AND FOR EXPLICITLY PROVIDING END USERS WITH APPROPRIATE WARNINGS AND DISCLAIMERS PERTAINING TO USE OF YOUR CHATBOT. YOU ARE SOLELY RESPONSIBLE FOR ANY PERSONAL INJURY OR DEATH THAT MAY OCCUR AS A RESULT OF YOUR CHATBOT OR YOUR USE OF MICROSOFT POWER VIRTUAL AGENTS IN CONNECTION WITH YOUR CHATBOT, INCLUDING (WITHOUT LIMITATION) ANY SUCH INJURIES TO END USERS.