IBM Business Automation and Digital Labor Hands-on Labs

IBM watsonx Orchestrate
AI Assistant

For IBM watsonx Expert Partner Agentic AI Academy EMEA

Last update: Tuesday, 08 April 2025
wxO 2025.03.27

Paul Pacholski - pacholsk@ca.ibm.com
Juergen Lukas - jlukas@de.ibm.com

Table of Contents

1 Introduction 3
1.1 IBM watsonx Orchestrate 3
1.2 Lab Scenario 3
1.2.1 Procurement of Computer Accessories Use Case 3
1.2.2 Solution Architecture 3
1.3 Lab Overview 4
2 Lab Setup Instructions 5
2.1 Download Lab Files 5
2.2 Login to wxO ..... **Error! Bookmark not defined.**
3 Author AI Assistant 6
3.1 Create AI Assistant 6
3.2 Create and Configure a Custom Extension 8
3.2.1 Create Custom Extension 8
3.2.2 Add Custom Extension to the AI Assistant 10
3.2.3 Configure Custom Extension 12
3.3 Import Prebuilt Actions 12
3.3.1 Examine the Variables 14
3.3.2 Examine Actions Set by Assistant 15
3.3.3 Examine User Defined Actions 16
3.4 Complete the Get Order Status Action 17
3.4.1 Add New Phrase 17
3.4.2 Add Step 3 – Retrieve the Order ID List 18
3.4.3 Add Step 4 – Select Order ID 20
3.4.4 Add Step 5 – Query Order Status 24
3.4.5 Add Step 6 – Display Order Status 28
3.5 Preview and Debug Your Assistant 30
3.5.1 Start the Preview 31
3.5.2 Create Sample Orders 31
3.5.3 Test the Get Order Status Action 33
3.6 Integrate Web Chat with Web Page 35
3.6.1 Change Assistant Web Page Background 35
3.6.2 Customize Assistant Style 37
3.6.3 Customize Home Screen 38
3.7 Test Your Web Chat 39

1 Introduction

1.1 IBM watsonx Orchestrate

IBM watsonx Orchestrate (wxO) is a generative AI and automation solution designed to help businesses automate tasks, simplify complex processes, and save time and effort. It provides a catalog of prebuilt apps and skills, a conversational chat experience, and a low-code builder studio to create scalable AI assistants and agents.

To learn more about IBM watsonx Orchestrate, click here. To see the technical documentation, click here.

1.2 Lab Scenario

1.2.1 Procurement of Computer Accessories Use Case

Focus Corp allows employees to order computer accessories (e.g., mouse, keyboard, headsets, etc.). The ordering system implemented using wxO's advanced AI-enhanced features provides a superior user experience and operational cost reductions.

- *Employees* use wxO AI Assistant. The AI Assistant guides users through the ordering steps, advises on company policies, and helps them make the best choices.
- *Managers* do not need to get involved with standard orders. They can focus on exception cases, making decisions based on the company's business needs and financial situation.
- *The purchasing department* uses an AI-assisted process to automate the purchasing of catalog accessories, including AI-assisted processing of quote documents.

1.2.2 Solution Architecture

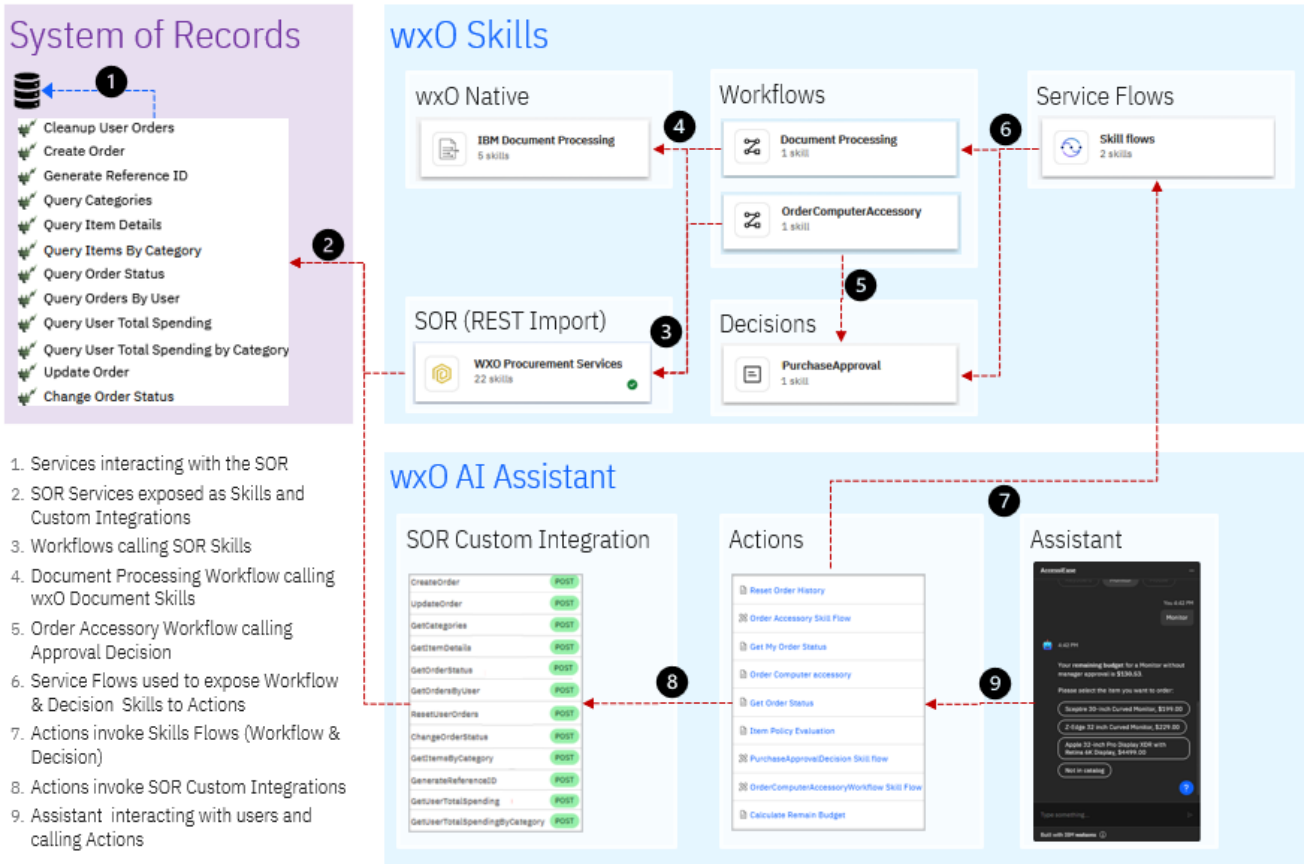


Figure 1. Procurement of Computer Accessories Solution Architecture

In case you were not familiar with some of the wxO Programming Models mentioned in the figure above, here are some definitions:

Skill

It is a self-contained capability to automate tasks. Skills implemenationcan be wxO components such as decision models, generative artificial intelligence (AI), workflows and API integrations

Sill Flow

A linear sequence of skills that uses an output of one skill as input for another, thus enabling seamless interaction between the skills.

Action

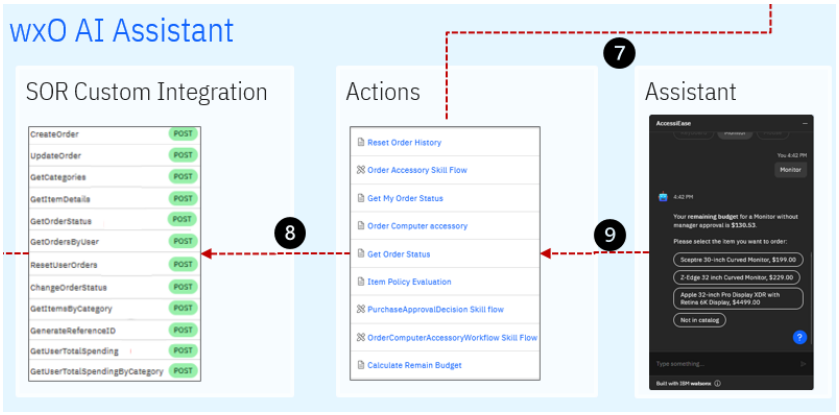
Actions are the building blocks of AI Assistants. An Action represents a discrete outcome you want your AI Assistant to accomplish in response to the user's request. It defines the interaction between a user and the Assistant about a particular question or request. This interaction begins with the user input that starts the Action. It might include more exchanges as the AI Assistant gathers more information, ending when the AI Assistant completes the request or answers the user's question.

Assistant

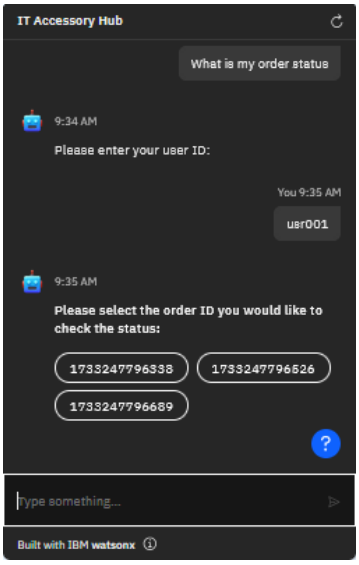
It is a tool designed to help organizations easily create and deploy intelligent virtual assistants. It leverages generative AI and large language models (LLMs) to enable responsive, natural conversations. The AI Assistant can handle tasks like answering customer queries, automating processes, and guiding users through multi-step workflows. It supports a low-code interface, making it accessible for users without extensive technical expertise.

1.3 Lab Overview

In this lab, you will implement a subset of the wxO AI Assistant part of the Solution architecture introduced in the figure above. Specifically, you will author the AI Assistant with a single action. Other labs will address different parts of the Solution.



The Action will retrieve a user's order status.



2 Lab Setup Instructions

2.1 Download Lab Files

_1. On the Github page containing this guide go to the **lab data** directory.

_2. Download all three files to your computer:

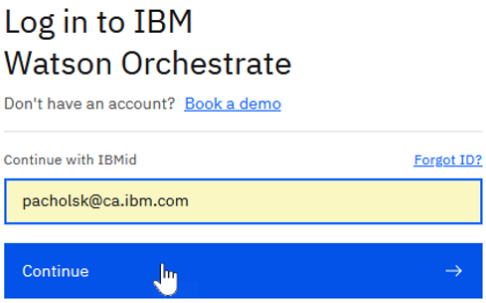
FocusCorp.jpg
ITAH_AIAssistantLab_wxoAssistantDraft.json
TAH_Assistant-lab-Import-action.json

2.2 Login to IBM watsonx Orchestrate

_1. In your web browser open IBM watsonx Orchestrate using the link provided to you.

_2. Enter your **IBM ID,** click **Continue,** enter your **password**, and click **Continue** again to log in.

_3. If you have access to multiple tenants, please select the tenant that was shared with you as part of the event.



3 Author AI Assistant

To create an AI Assistant, you will follow the below steps to complete the lab:

- Create AI Assistant
- Create Custom Extensions (external service that Actions can call)
- Create Actions
- Preview and debug your AI Assistant
- Integrate web chat with a web page

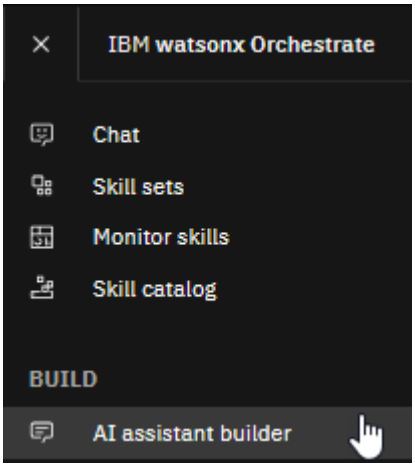
If you are unfamiliar with the AI Assistant programming model, click the link to learn more.

3.1 Create AI Assistant

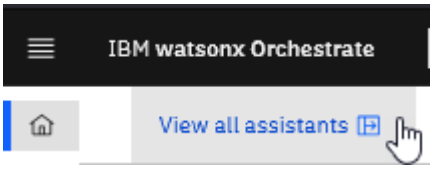
An AI Assistant in IBM Watson Orchestrate is a tool designed to help organizations create and deploy AI-powered virtual assistants quickly and easily. It leverages generative AI and digital skills to enable users to build AI assistants through a low-code experience. These AI assistants can perform tasks, complete complex processes, and provide customers and employees with a highly engaging natural language experience.

_1. Click the **Hamburger** menu.

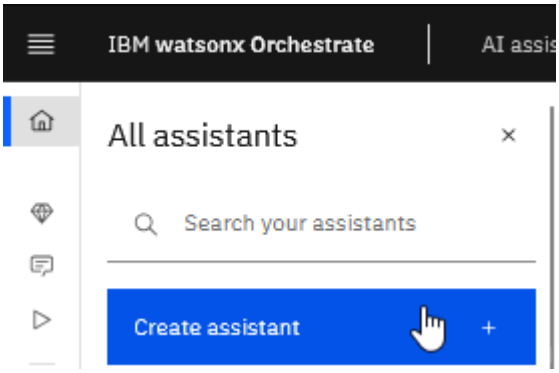
_2. Click **AI assistant builder**.



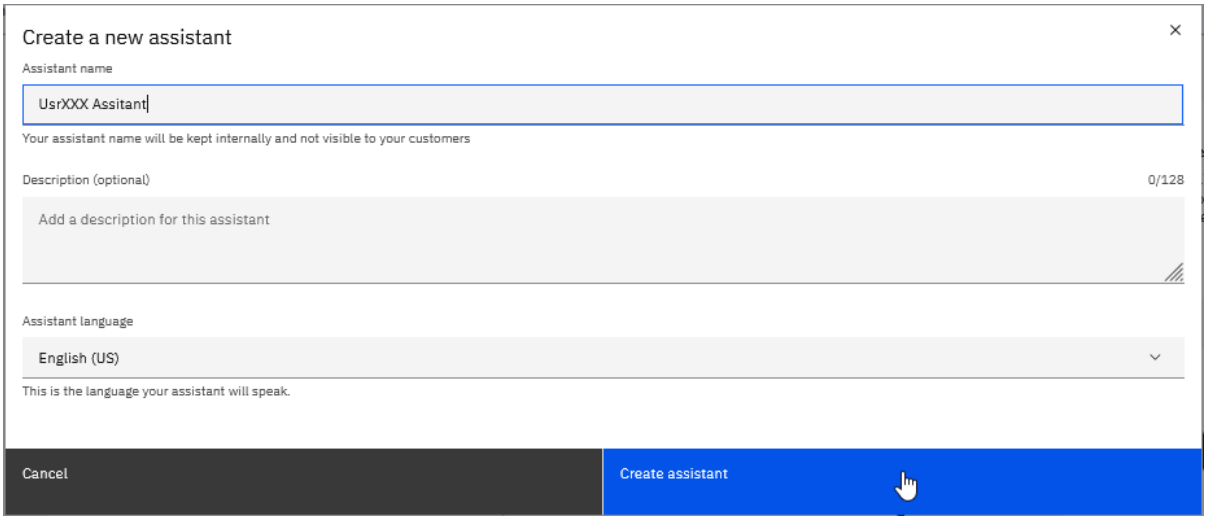
_3. Click **View all assistants**.



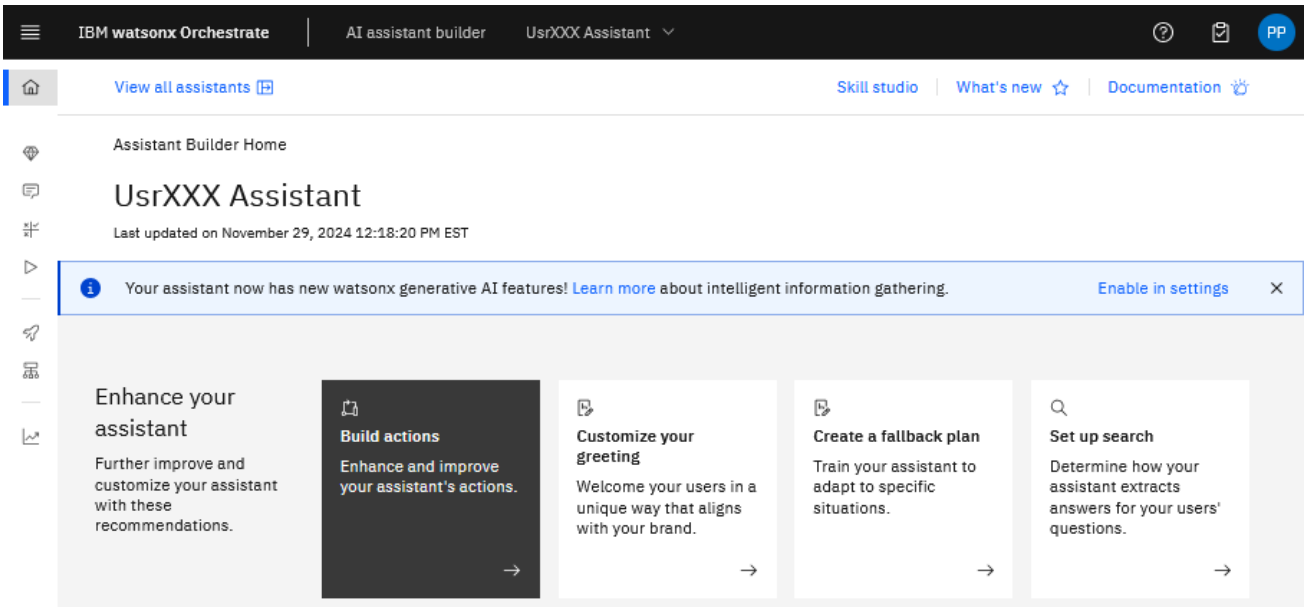
_4. Click **Create Assistant +.**



_5. For the *assistant name*, enter **UsrXXX Assistant** (replace XXX with the three-digit number in the user id you have received), provide an optional description, and then click **Create Assistant**.



Your new AI Assistant should now be available in the AI Assistant Builder.



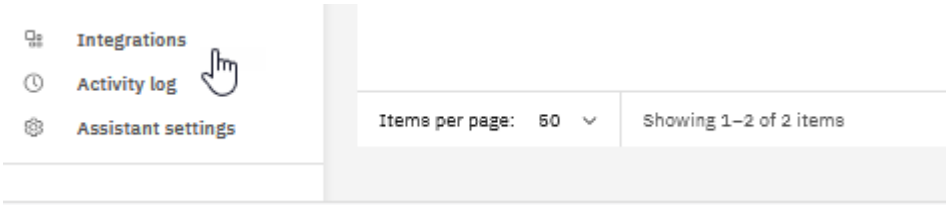
3.2 Create and Configure a Custom Extension

Since you must integrate your AI Assistant with an external service, you must build a Custom Extension by importing an OpenAPI document.

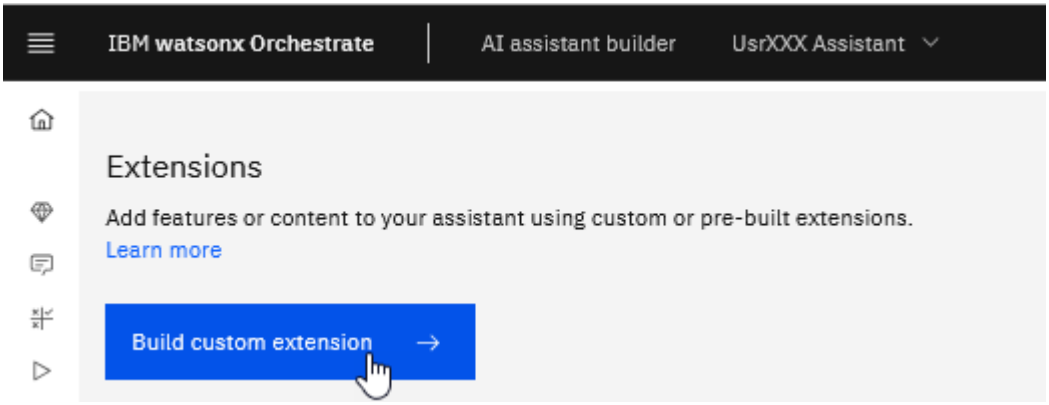
Let's create a Custom Extension by importing the **ITAH_Assitnat-lab-mport-action.json** OpenAPI file, which includes two operations you must call from your Actions.

3.2.1 Create Custom Extension

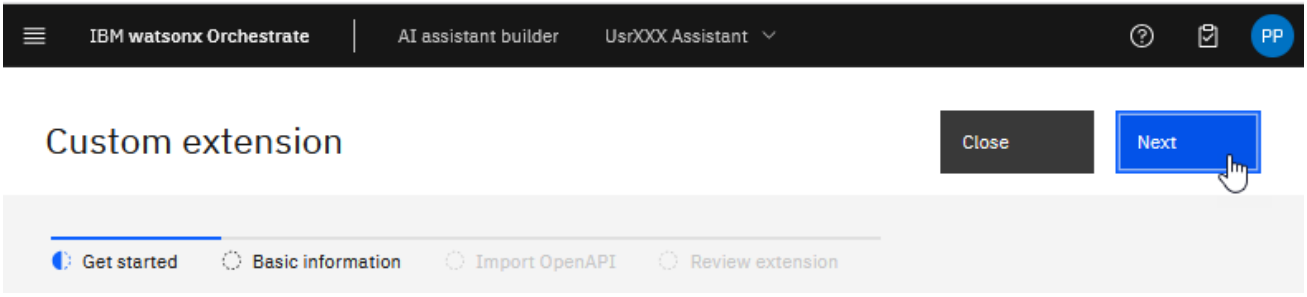
_1. In the bottom left corner, click **Integrations.**



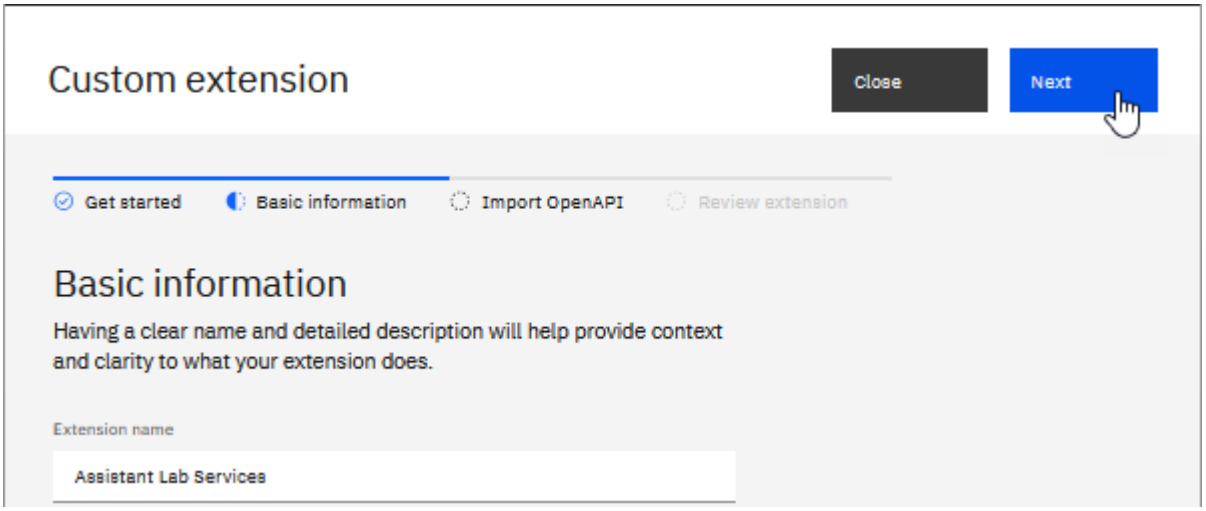
_2. Scroll down to the **Extensions** section and click the **Build customer extension** button.



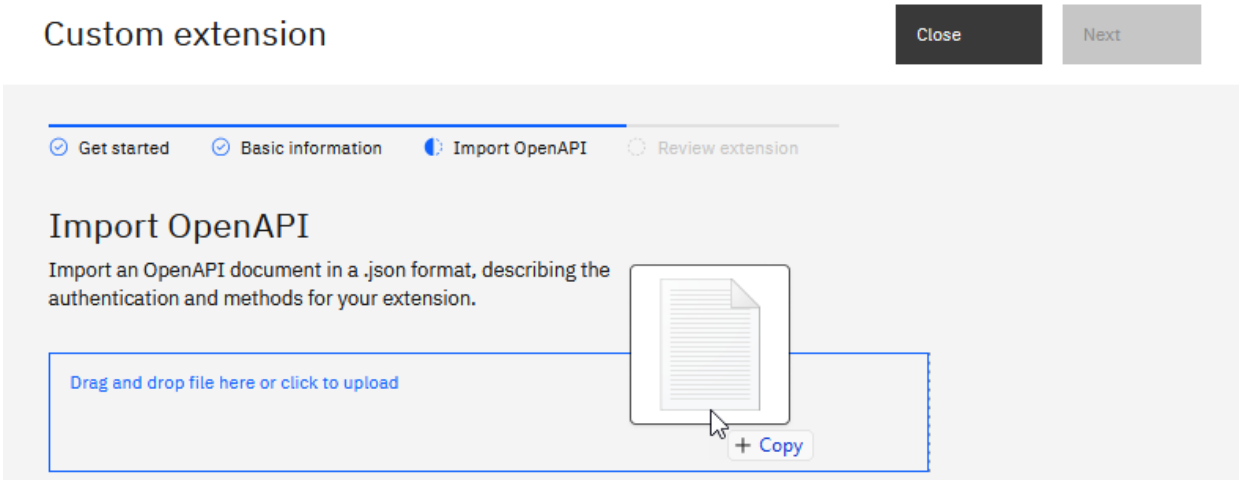
_3. Click **Next**.



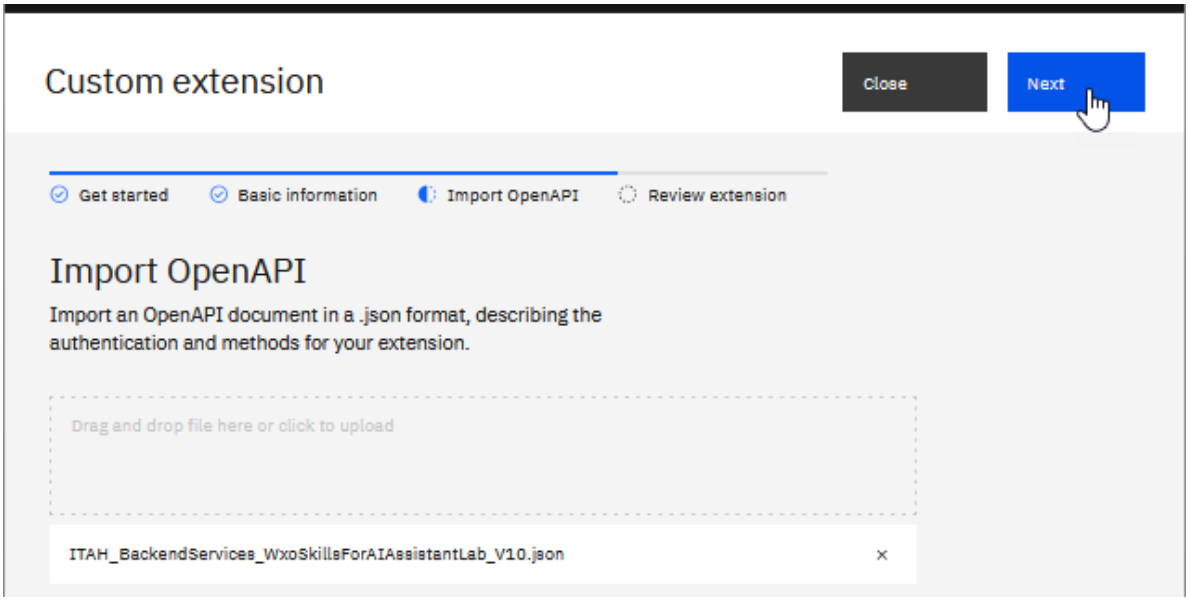
_4. For the *Extension name,* enter **Assistant Lab Services** and click **Next**.



_5. Drag and drop the **ITAH_BackendServices_WxoSkillsForAIAssistantLab_V10.json** file you downloaded.

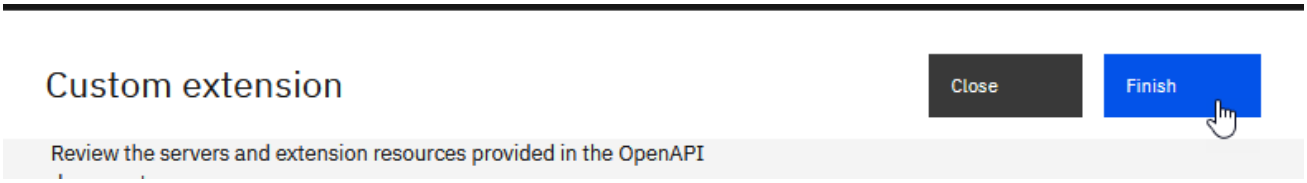


_6. Click **Next.**



Note the operations and their parameters in the JSON file. You will use the *GetOrderStatus* and *GetOrderbyUser* operations later in this lab.

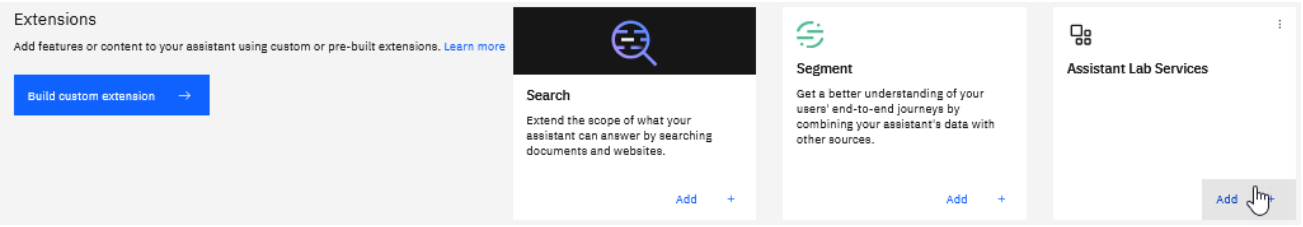
_7. Click **Finish**.



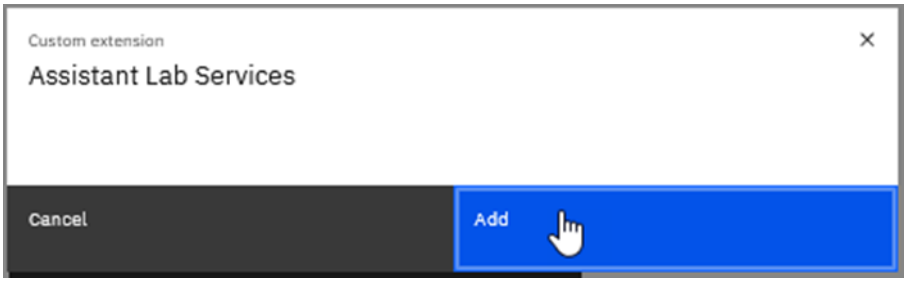
3.2.2 Add Custom Extension to the AI Assistant

You will see a new tile in the Extensions section. Now that we have created the Extension, we must add it to inform the AI Assistant. We must also configure the authentication details, such as user ID and password.

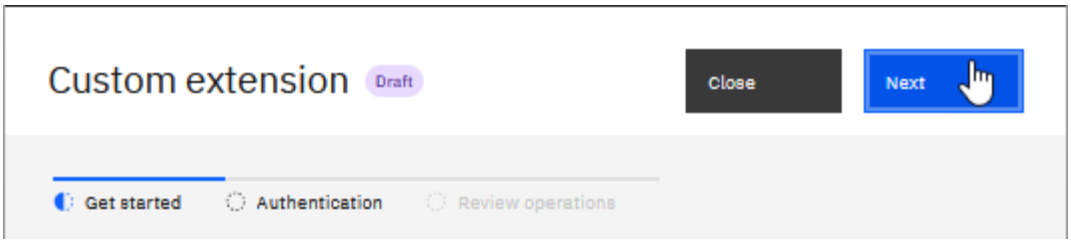
_1. On the *Assistant Lab Services tile*, click **Add +**.



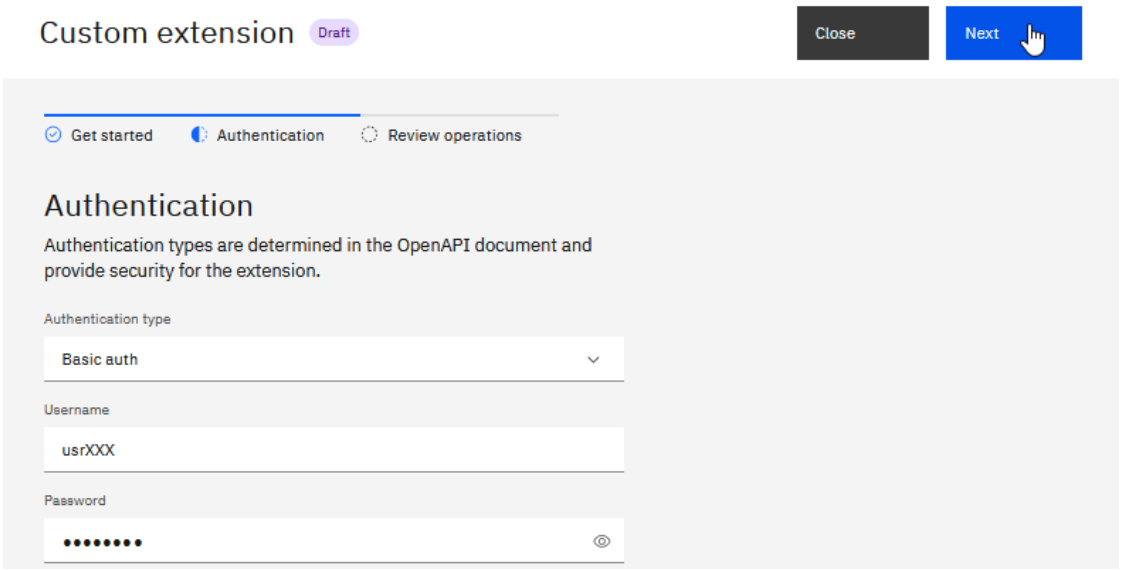
_2. Click **Add.**



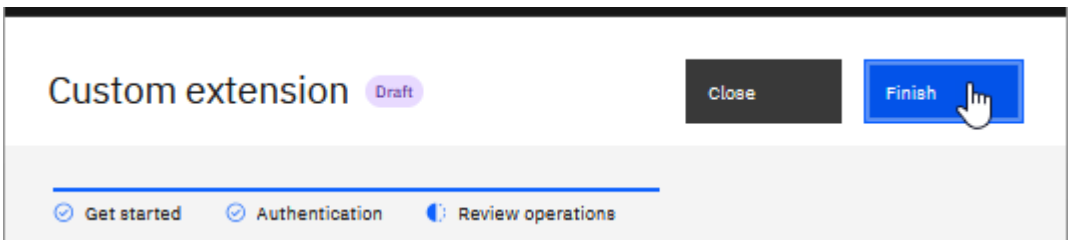
_3. Click **Next**.



_4. Select *the Authentication type as the* **Basic auth**, *use* the **credentials you have received** to enter your *username* and *password,* and then click **Next.**

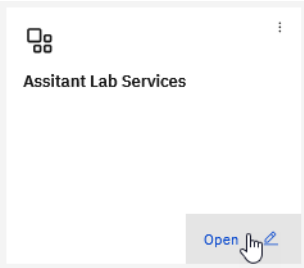


_5. Click **Finish** to add the Extension to your AI Assistant.

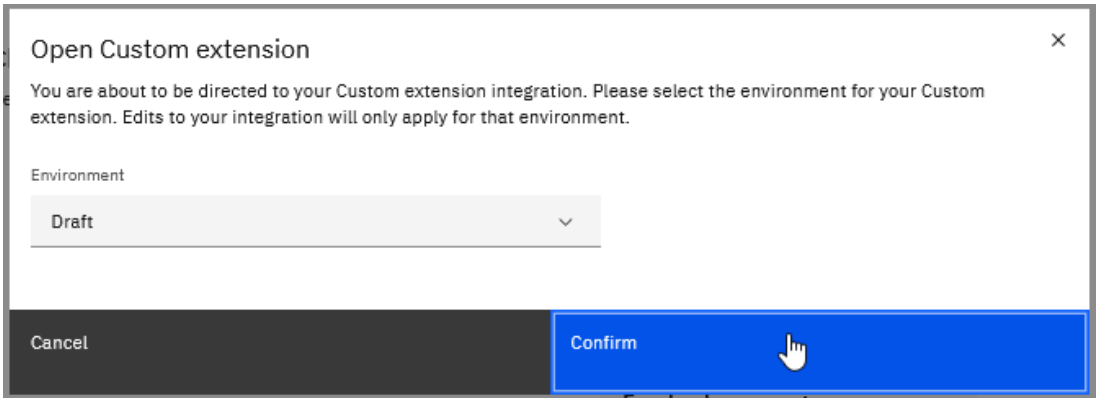


3.2.3 Configure Custom Extension

_1. On the **Assistant Lab Services** tile, click **Open.**

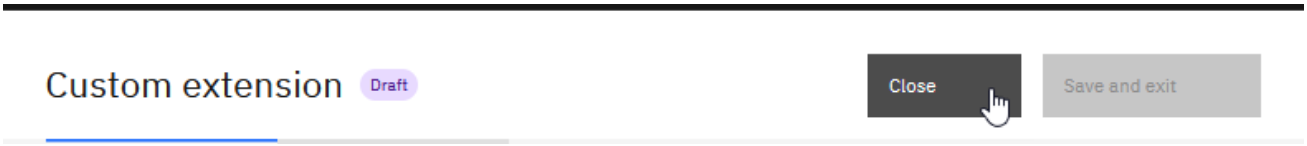


_2. Ensure that the **Draft** *Environment* is selected and click **Confirm**.



Note. AI Assistants have two environments: Draft and Live. The Draft environment is your development environment where you can build, debug, and test your AI Assistant. The Live environment is a production environment available to the users.

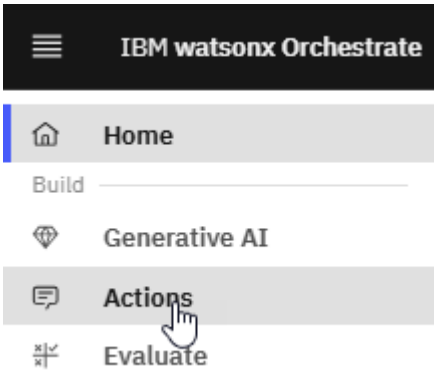
_3. Click **Close.**



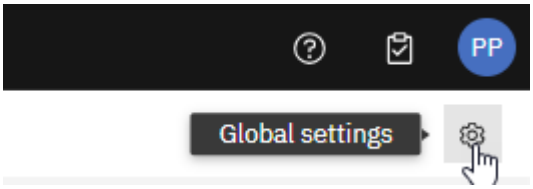
3.3 Import Prebuilt Actions

Rather than building all the AI Assistant actions from scratch, you will import prebuilt AI Assistant Actions using the *AI Assistant Lab Actions.json* file and then enhance them to call the Custom Extension you created. This will speed up the lab development time by avoiding repetitive steps required to build the entire AI Assistant.

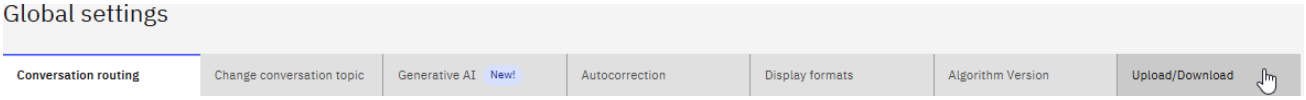
_1. From the toolbar, select **Actions**.



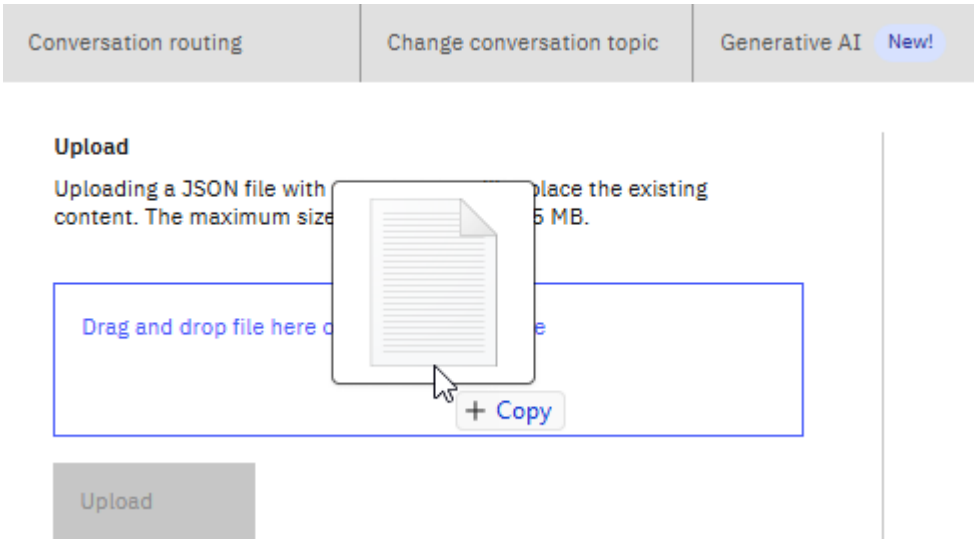
_2. Click the **Global settings icon** in the top right corner.



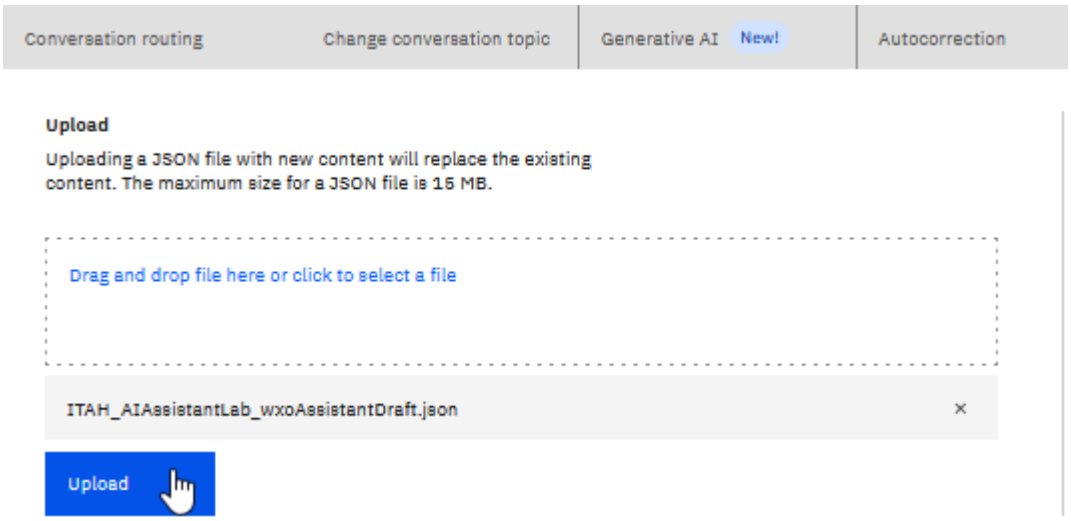
_3. Click the **Upload/Download** tab (you might need to click on the > multiple times to see it).



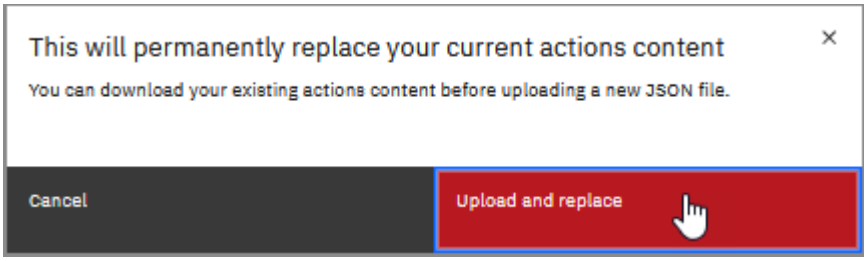
_4. Drag and drop the **ITAH_AIAssistantLab_wxoAssistantDraf.json** you downloaded from the lab folder.



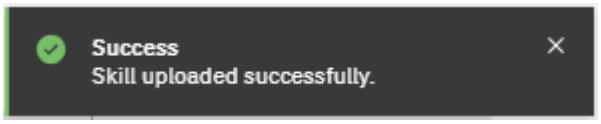
_5. Click **Upload.**



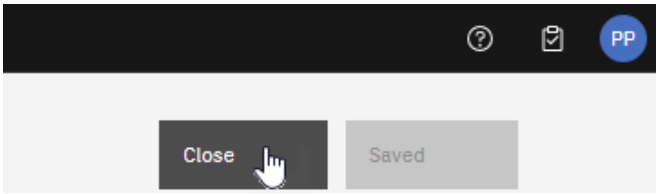
_6. Click **Upload and replace**.



You should see a Success popup message.

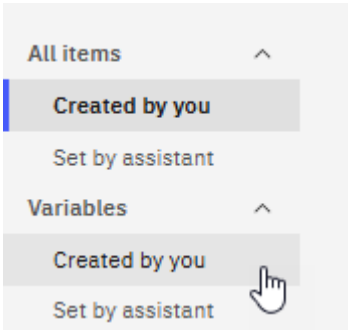


_7. Click **Close.**



3.3.1 Examine the Variables

_1. Click the **Variables > Created by you** tab.



_2. Note the variable *userID*.

Name	Actions count	Initial value	Description	Variable ID
Tт userID	3			userID

The userID variable is a Session Variable. It is initialized when you log into the AI Assistant and remains available for all subsequent actions.

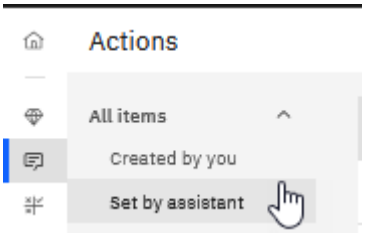
Session Variables persist throughout the user's interaction with the AI assistant, and your AI assistant can reference them from any action. If you want to retain the value for future use, you can create a session variable to store the value from an *Action Variable*.

Action Variables persist only during an Action's execution. When an Action step collects user information, the response is automatically stored in an Action Variable.

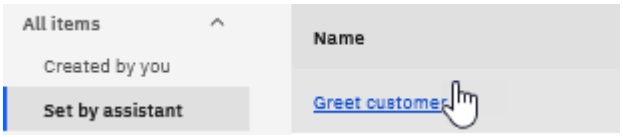
3.3.2 Examine Actions Set by Assistant

The Actions in the *Actions set by assistant* category are predefined in all AI Assistants. They can be customized.

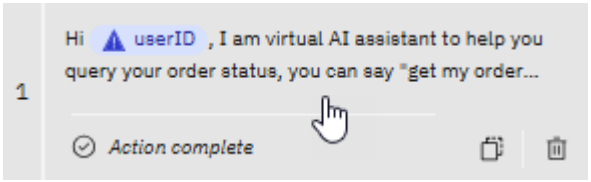
_1. Click **All items > Set by assistant.**



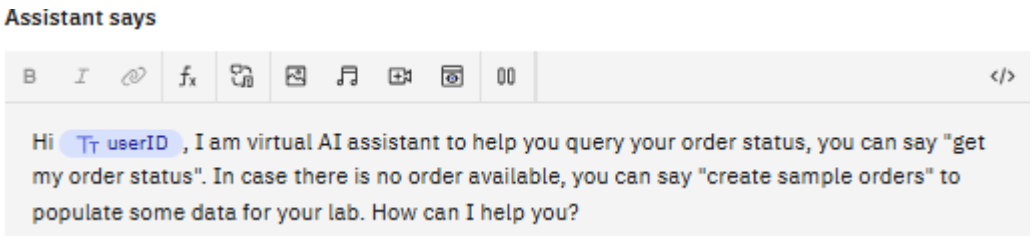
_2. Click the **Greet customer** action.



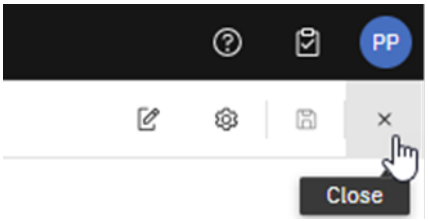
_3. Click the Step **1** tile.



_4. Notice that the greeting includes the userID variable, which is initialized from the login dialog when the AI Assistant is started.

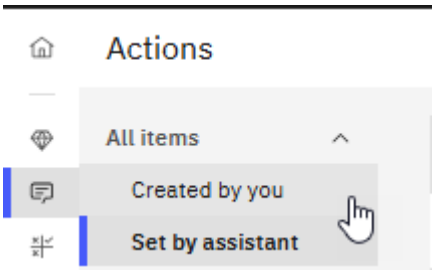


_5. Click **X** at the top right corner to close the *Greet customer* action.

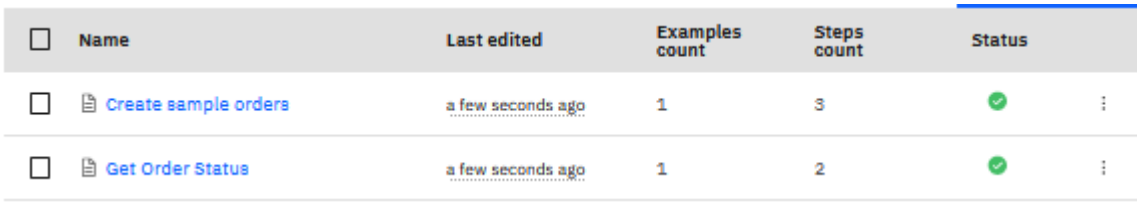


3.3.3 Examine User Defined Actions

_1. Click **All items > Created by you.**



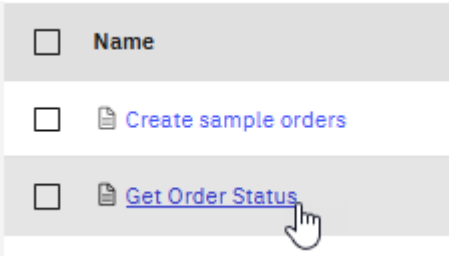
_2. Note that there are two user-defined Actions.



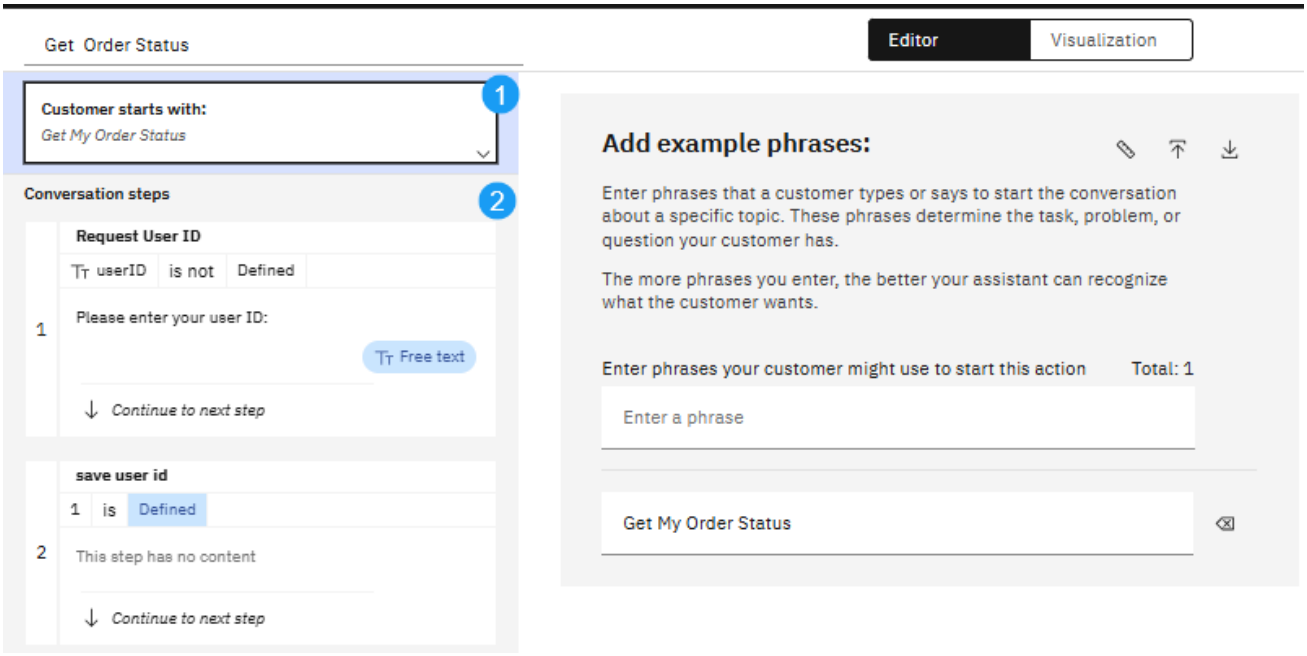
The **Create Sample Orders action** creates a sample computer accessory order for you. You will use this Action to generate a simple order that will associate an order with your user ID.

The **Get Order Status action** shows the status of your accessory order request (based on your user ID). It is only partially completed. In the following lab steps, you will add steps to complete the implementation.

_3. Click the **Get Order Status** action.



_4. Note that the two steps in this Action were already created for you:



1) The **Customer starts with** a section with a phrase to start this Action.
2) There are two **Conversation steps:** (1) Ask the user for a user ID and (2) save the user id.

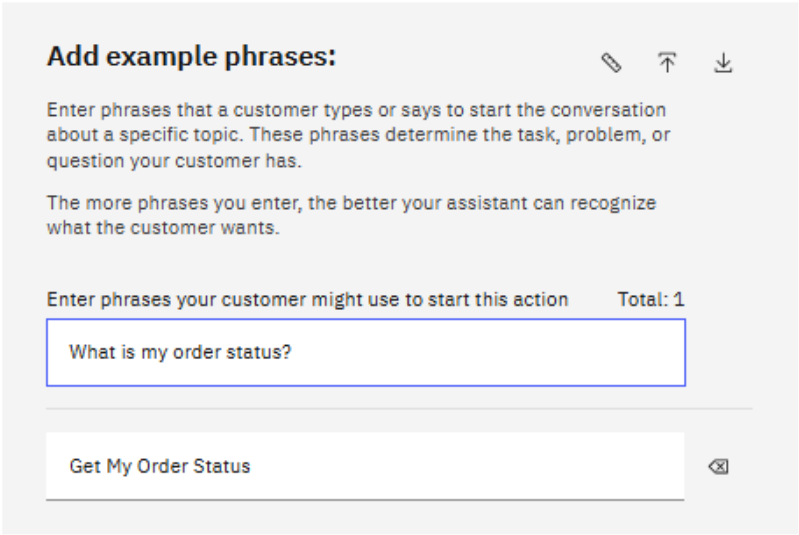
3.4 Complete the Get Order Status Action

In this part of the lab, you will complete authoring the *Get Order status* Action. Specifically, you will:

- Add a new phrase to start the Action.
- Add additional steps to query order status:
    - Call the Custom Extension to retrieve the user's orders.
    - Allow the user to select and order.
    - Call the custom extension to retrieve its status.
    - Display the order status in the Chat.

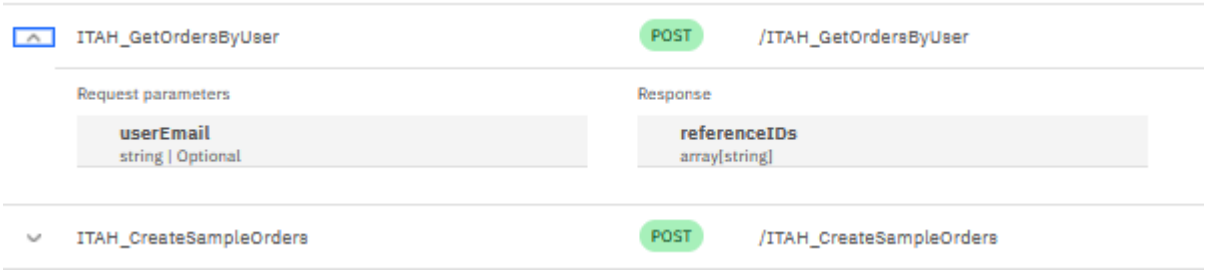
3.4.1 Add New Phrase

_1. For *Enter phrases your customer might use to start this Action,* enter a second phrase: **What is my order status?**



3.4.2 Add Step 3 – Retrieve the Order ID List

In this step, you will invoke the Custom Integration and map its input variable (userEmail).



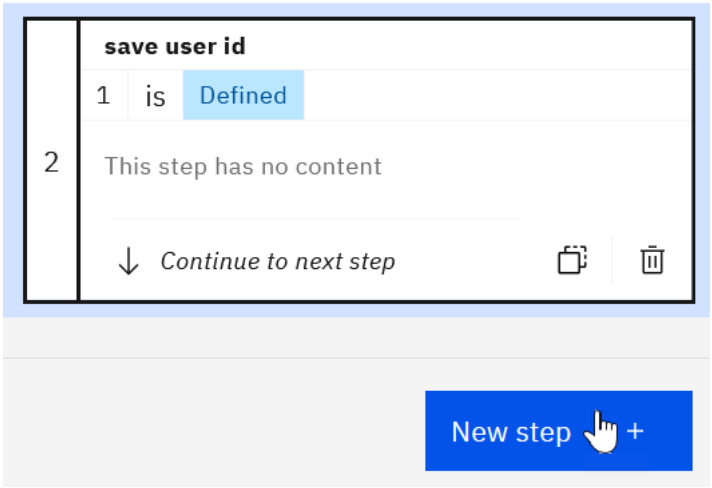
The output variable does not need to be mapped, as it will automatically be set to the body variable:

In IBM AI Assistant, the **body variable** plays a crucial role when retrieving the output of custom integrations. After the AI Assistant sends a request to an external service using the body variable as the payload, the response from the service is processed and mapped to action variables. These action variables can then be used in subsequent steps of the Action or conversation.

3.4.2.1 Create Step 3

In this step, you will retrieve orders associated with the logged-in user. A user may have multiple orders, which will be stored as a session variable associated with this step.

_1. Select **Step 2** and click **Next step +** to add a new step after Step 2.



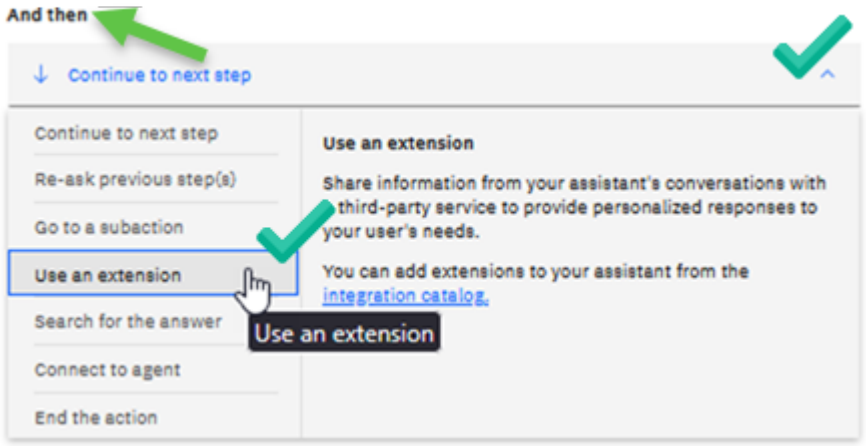
Note: If the Action label is not 3 (because you did not select step 2 first), you can reorder the newly created step by dragging and dropping.

_2. Click **Edit step title**.

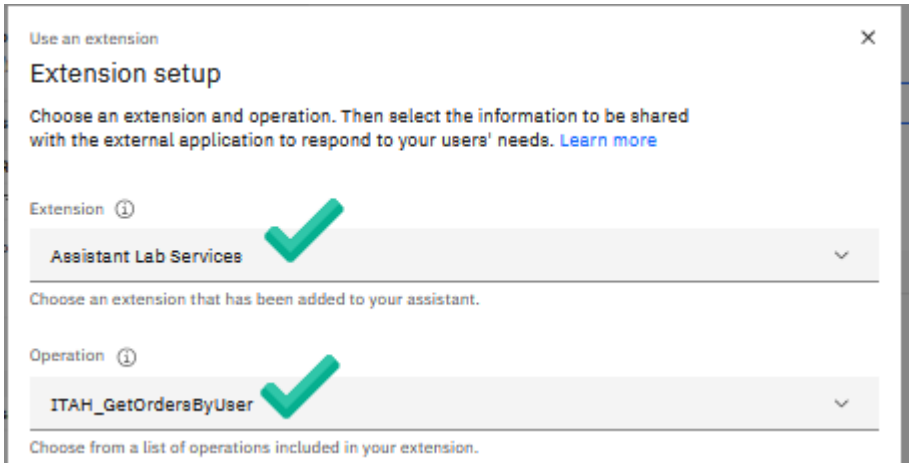
_3. Enter **Retrieve order ID List**.

3.4.2.2 Call Custom Extension to Get Order IDs

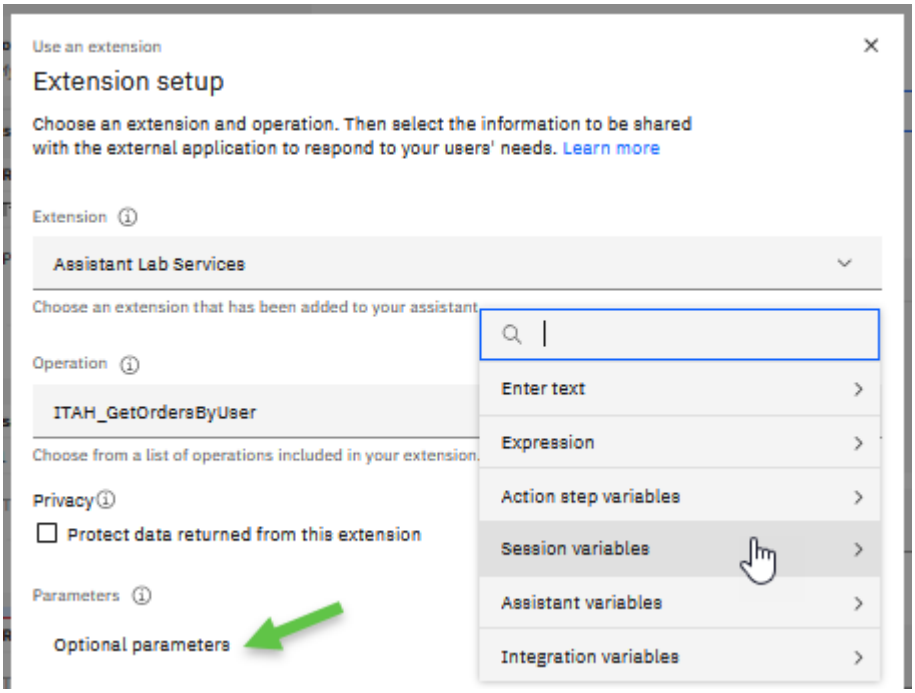
_1. In the *And then* section from the **dropdown**, select **Use an extension**.



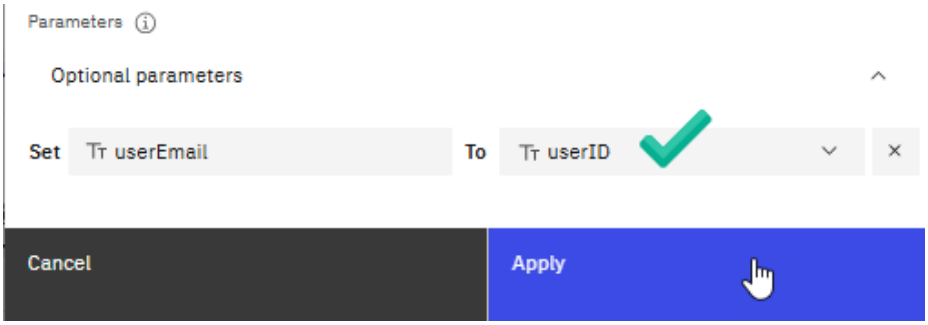
_2. For *Extension,* select **Assistant Lab Services**; for *Operation,* select **ITAH_GetOrdersByUser**.



_3. In the *Optional parameters* section from the *To* dropdown, select **Session variables**.



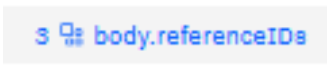
_4. From the dropdown, select **userID** and click **Apply.**



3.4.3 Add Step 4 – Select Order ID

Recall that the Custom Integration call output you added in Step 3 was a list of order IDs for the logged-in user.

The list of order IDs was retrieved (using the Custom Integration ITAH_GetOrderByUser) in the previous step and saved in step 3's session variable:

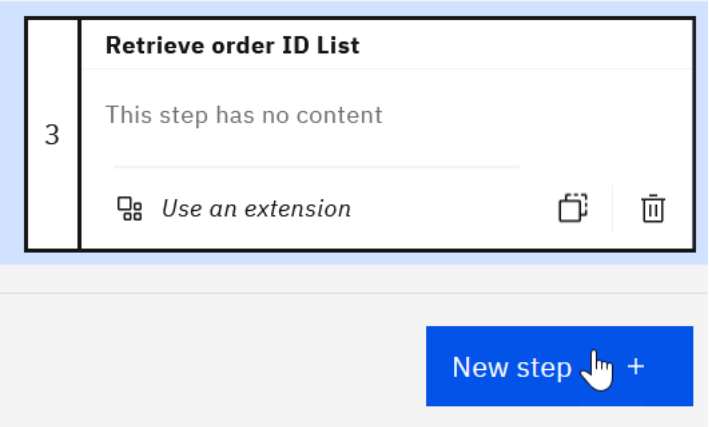


This step will ask the user to select one order ID from **body.referenceIDs**.

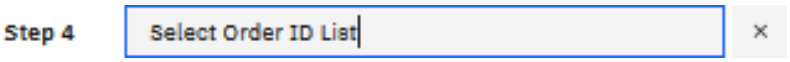
We must, however, ensure that this step is executed if the retrieved order ID **body.referenceIDs** variable contains at least one Order ID.

3.4.3.1 Create Step 4

_1. Click **New step +**.



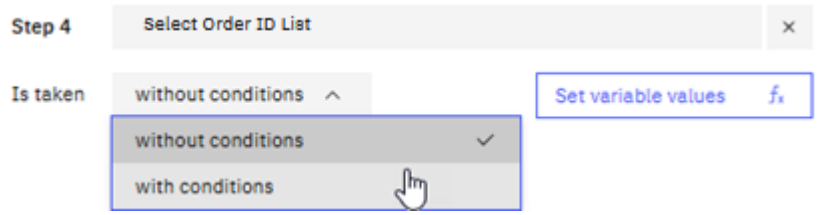
_2. Set the step name to **Select Order ID List**.



3.4.3.2 Ensure there is at least one Order ID

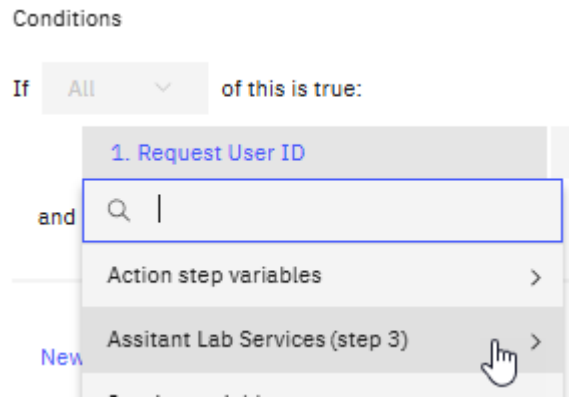
Let's make sure that we have at least one Order ID.

_1. For *Is taken*, select **with conditions**.

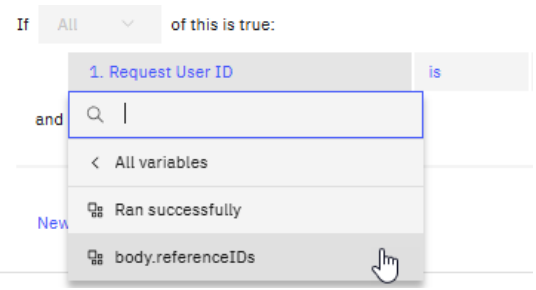


_2. In the *Conditions* section, click **1. Request User ID**.

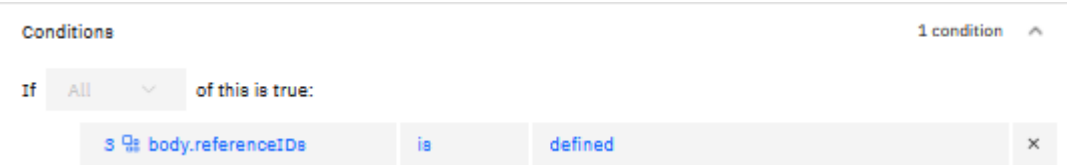
_3. Select **Assistant Lab Services (step 3)**.



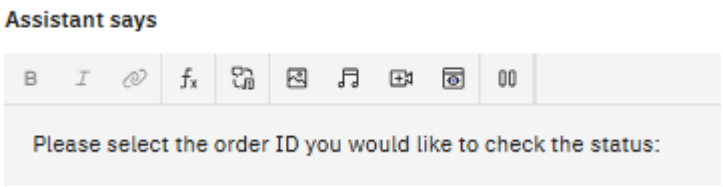
_4. Select **body.referenceIDs** and keep *is defined*.



The Condition should look as shown below:



_5. For *Assistant Says*, enter **Please select the order ID you would like to check the status:**



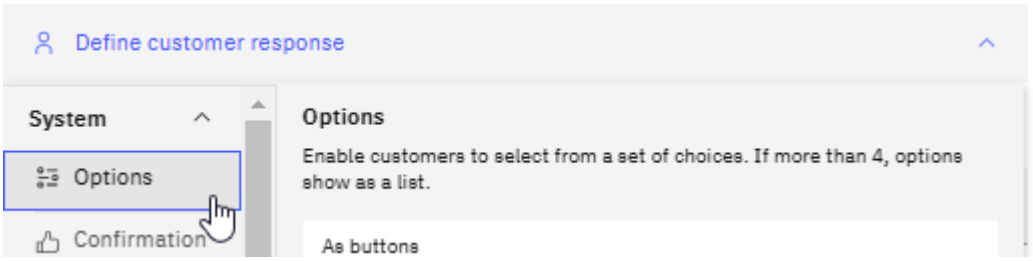
3.4.3.3 Create Response

The response will create a list of Order IDs so the user can select one.

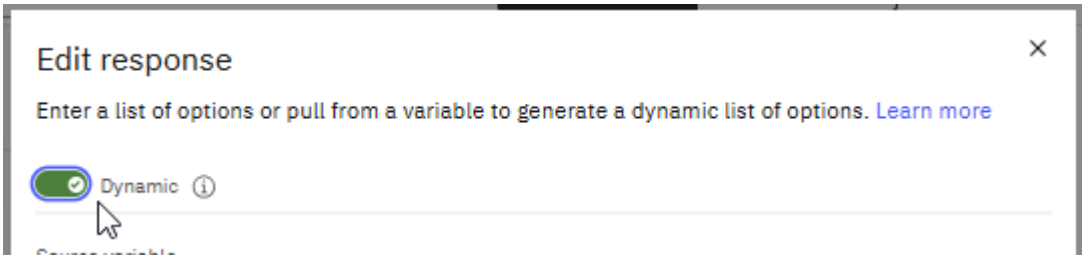
_1. In *Assistant says* click **Define customer response**.

You will now create a response with a list of order IDs and ask the user to select one.

_2. Click **Options**.

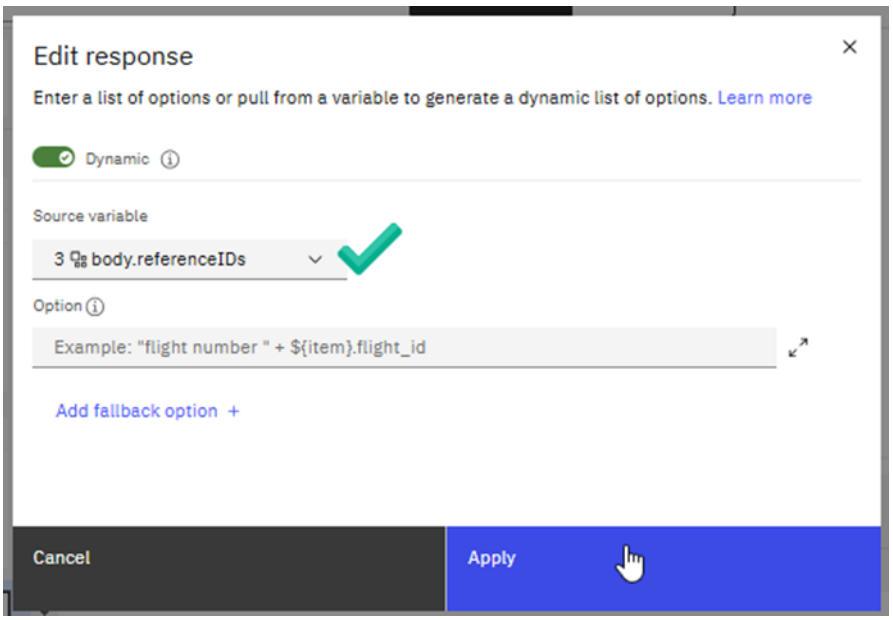


_3. Check the **Dynamic checkbox**.

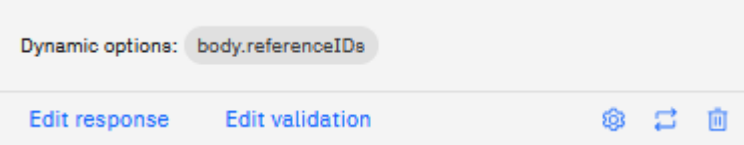


_4. For the *Source variable*, select **Assistant Lab Services (step3) > body.referenceIDs** and click **Apply**.

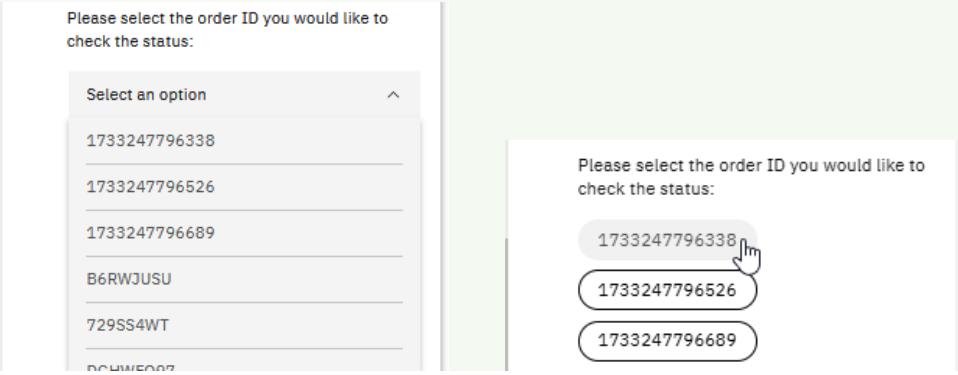
Since the referceIDs is a list of strings, we need to set the display option for this variable as dynamic since we do not know a priori how many reference IDs there may be.



_5. Verify that Dynamic options for displaying a list of reference IDs look as shown below:



Note that depending on the number of values in the body.referenceIDs list, the output will be either Bubbles or a List format:

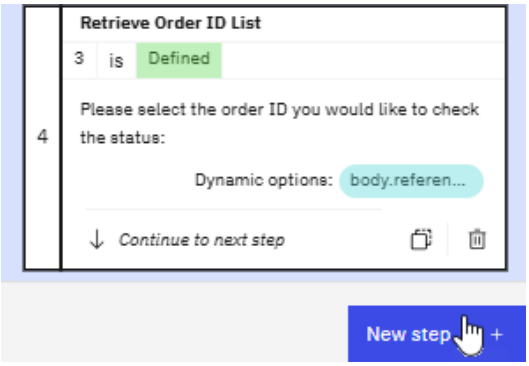


3.4.4 Add Step 5 – Query Order Status

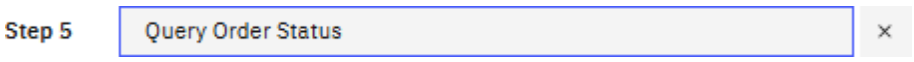
This step will return the status of their computer accessory request.

3.4.4.1 Create Step 5

_1. Click **New step +**.

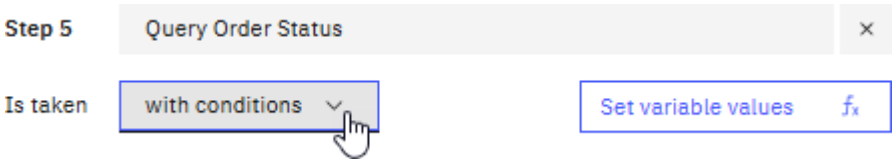


_2. Set the step name to **Query Order Status**.



3.4.4.2 Ensure an Order ID is selected.

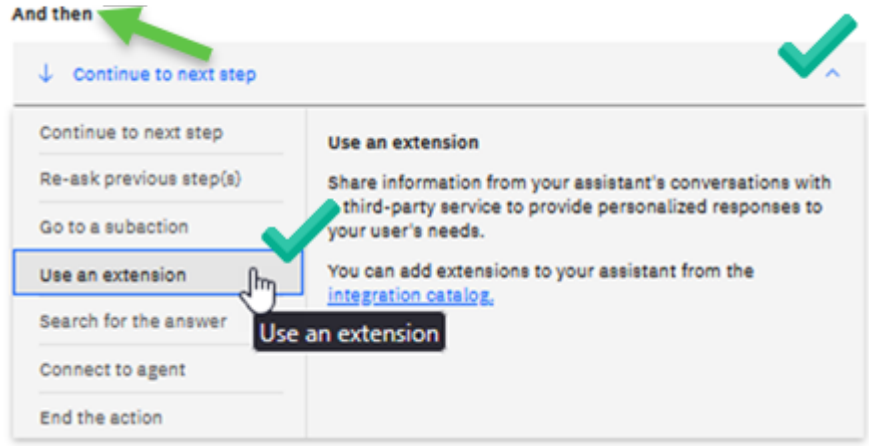
_1. For *Is taken*, keep the already defined condition: **with conditions**.



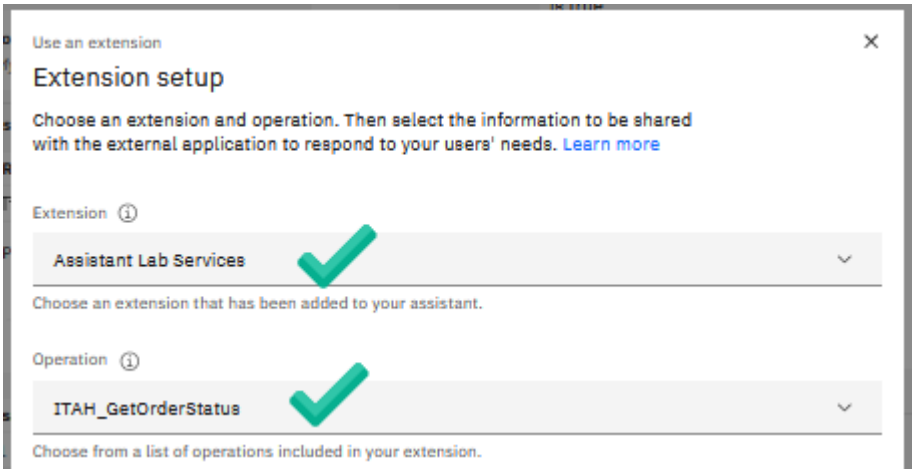
_2. Ensure the *Action step variable* **4. Select Order ID List** is selected in the *Conditions* section.

3.4.4.3 Invoke Custom Integration to get the status of the Order ID

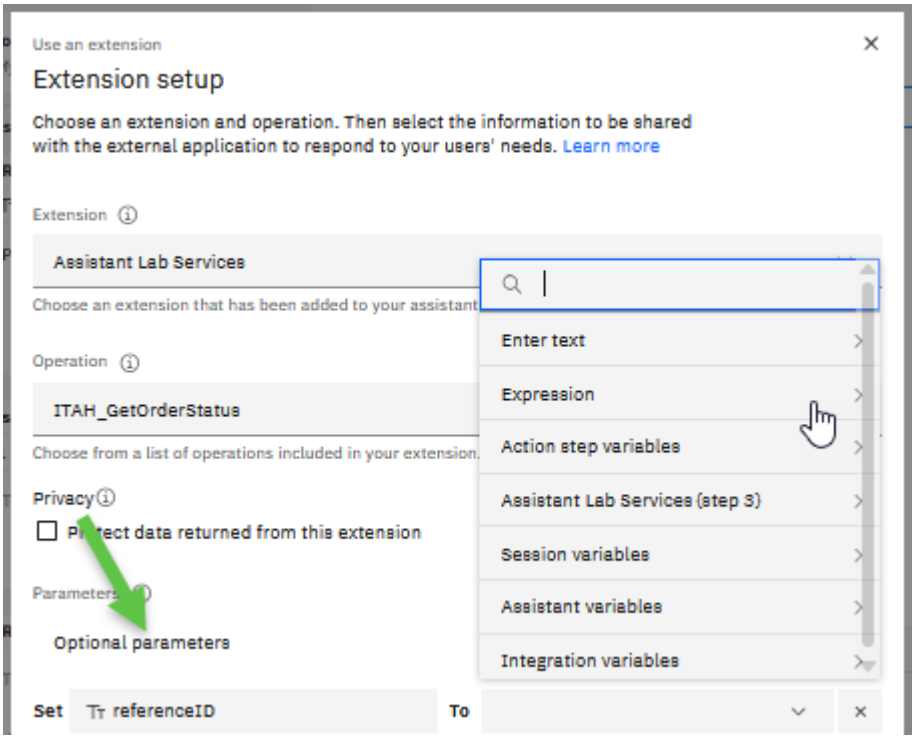
_1. In the *And then* section from the dropdown, select **Use an extension**.



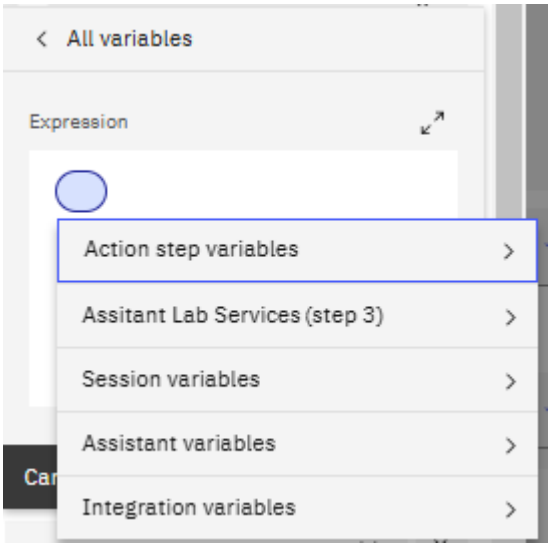
2. For *Extension,* select **Assistant Lab Services**; for *Operation,* select ITAH**GetOrderStatus**.



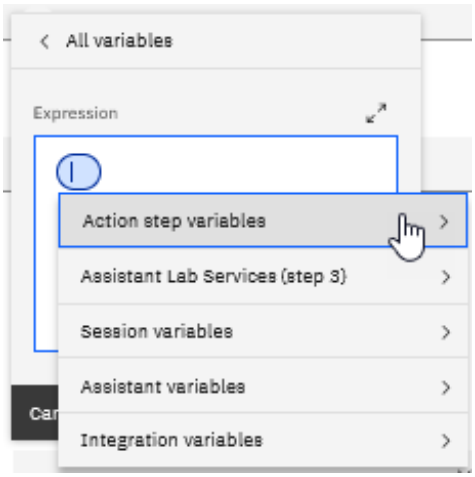
_3. In the *Optional parameters* section from the *To* dropdown, select **Expression**.



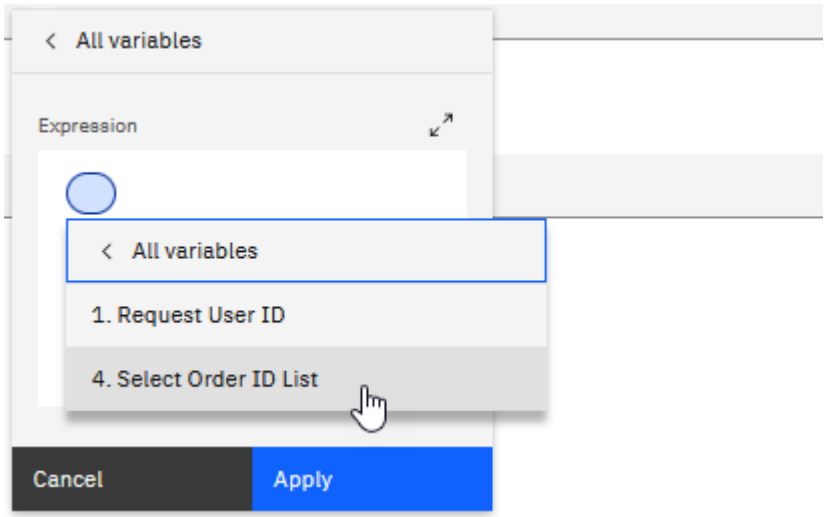
_4. For *Expression*, enter **$**. A bubble with a dropdown list will appear.



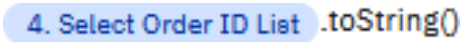
_5. Select **Action step variables.**



_6. Select **4. Select Order ID List.**

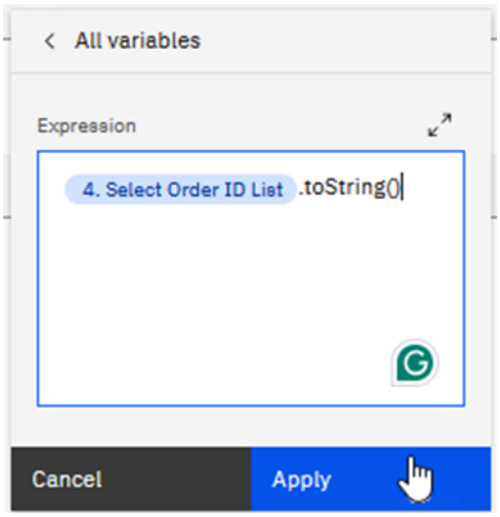


_7. Append "**.toString()**" to the selected variable.

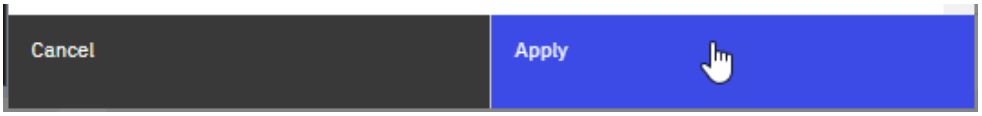


Note: We need to convert the order ID to a String because the AI Assistant will convert the order ID selected from Step 4 to a long number. But since the Extension Service input parameter is String, we need to make the long to String conversion.

_8. Click **Apply.**



_9. Click **Apply** on Extension setup.

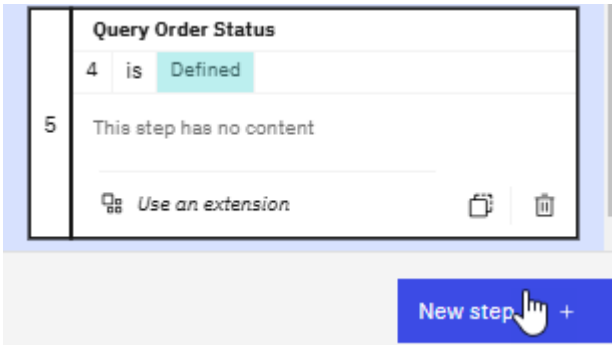


3.4.5 Add Step 6 – Display Order Status

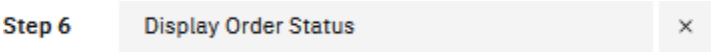
This step will display the status of their computer accessory request.

3.4.5.1 Create Step 6

_1. Click **New step +**.

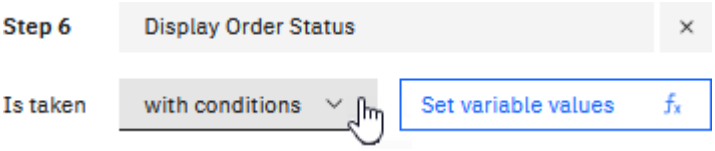


_2. Set the step name to **Display Order Status.**

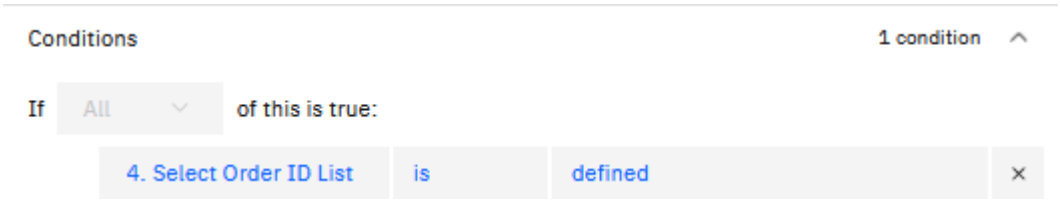


3.4.5.2 Ensure an Order ID is selected.

_1. For *Is taken*, select **with conditions**.

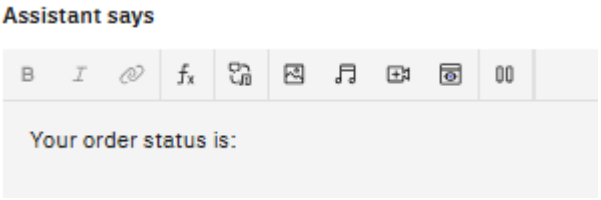


_2. Ensure the condition is set to: **4. Selected Order ID List.**

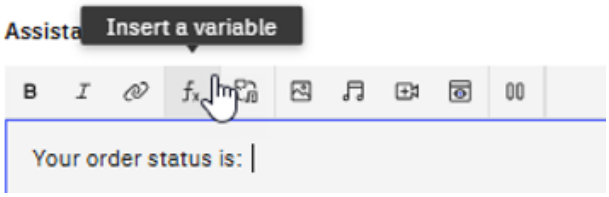


3.4.5.3 Display the Order Status

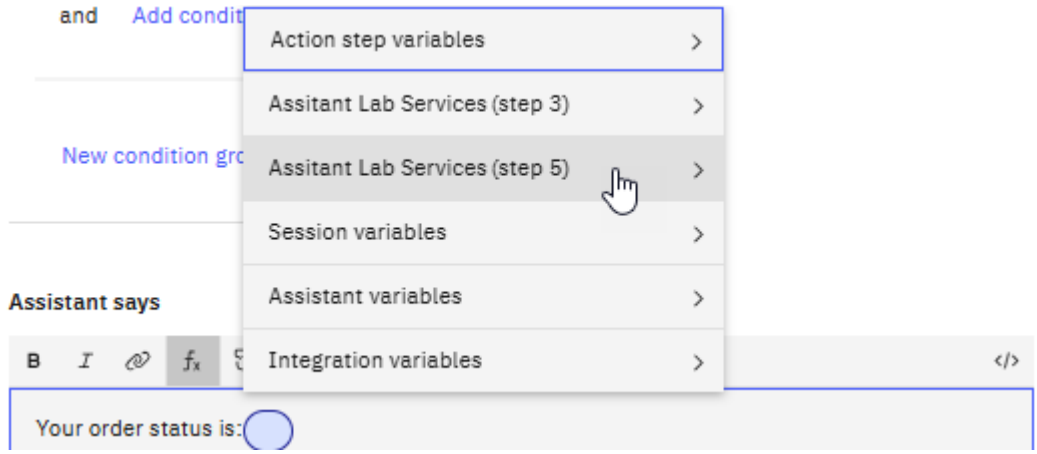
_1. For *Assistant says,* enter "**Your order status is: ".** Ensure there is a space after the colon.



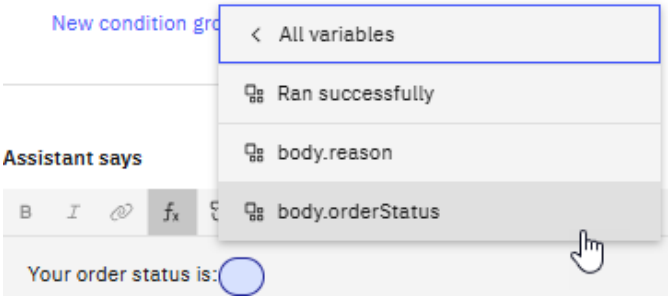
_2. Place the cursor after **"is: "** and click **Insert a variable icon (fx).**



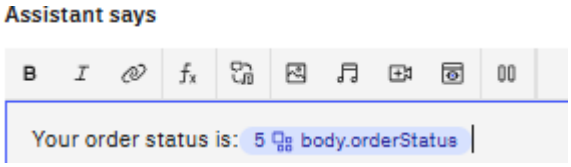
_3. Select **Assistant Lab Services (step 5)**.



_4. Select **body.orderStatus**.

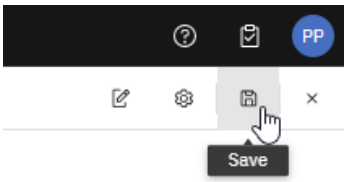


The output of this Action should look exactly like this:



_5. Since this is the last step of the Action, for *And then* select **End the Action**.

_6. In the top right corner, click the **Save** icon.

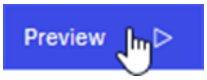


3.5 Preview and Debug Your Assistant

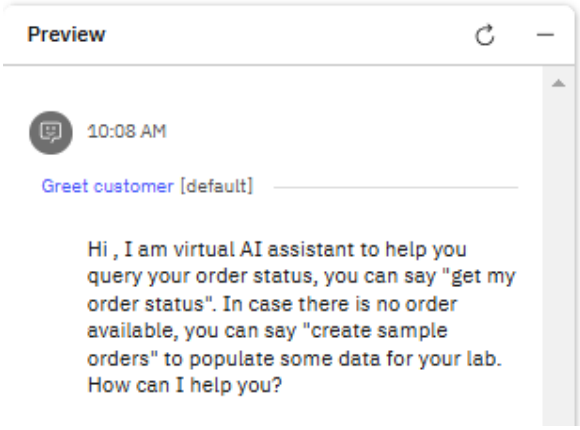
After you build an assistant, you can test it to ensure it works as you intended before making it available to the users. The AI assistant builder makes it straightforward to preview and debug your Assistant in the development environment before you're ready to publish the final version.

3.5.1 Start the Preview

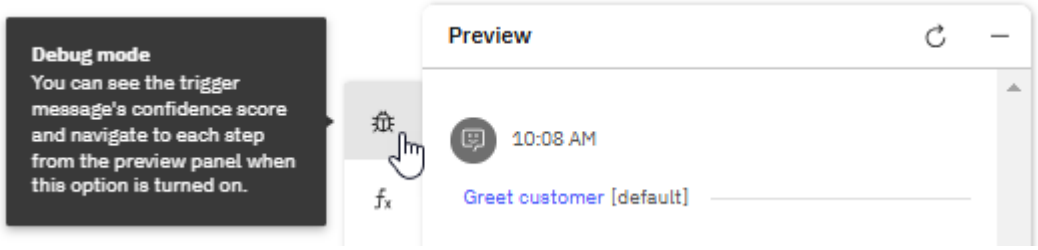
_1. Click the **Preview** button located in the bottom right.



You should now see the default Greeting defined in the **Greet customer** action we have configured for you.



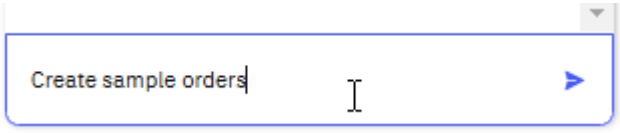
_2. Click the **Debug mode icon**.



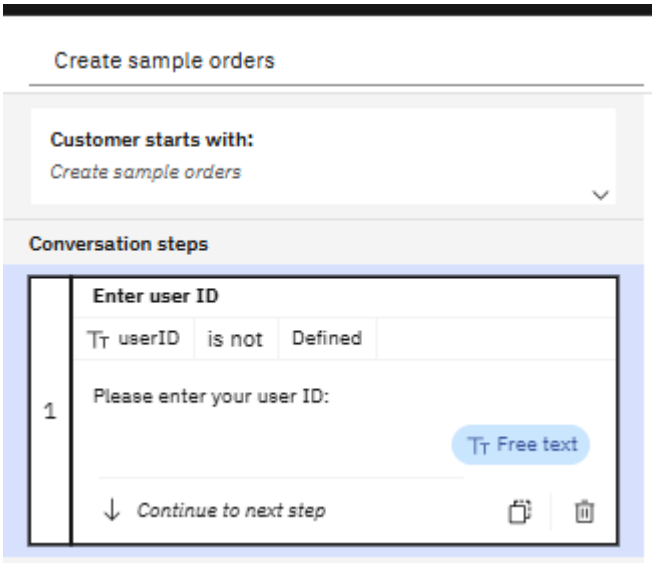
3.5.2 Create Sample Orders

Let's create some sample orders to test the Get Order Status Action!

_1. In the Chat, enter **Create sample orders** phrase and press **enter.**



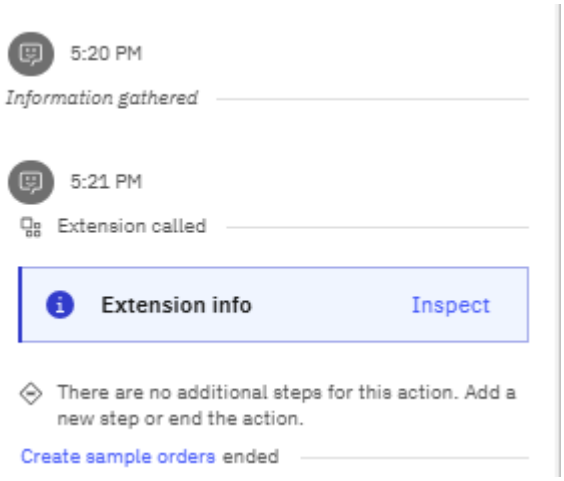
_2. Note that since you are in Debug Mode, the editor opens at Step 1 of the Create sample order Action.



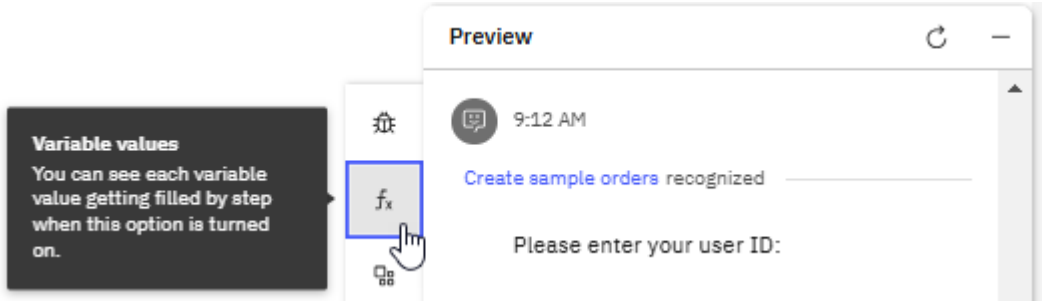
Note that the Assistant will recognize that the user ID variable userID has yet to be initialized because we did not log in to the Assistant like we would in a production environment. The Assistant will prompt you to enter your user ID.

_3. In the Chat, enter **usrXXX** (remember to replace XXX with the three-digit number in the user id you have received) and press **enter.**

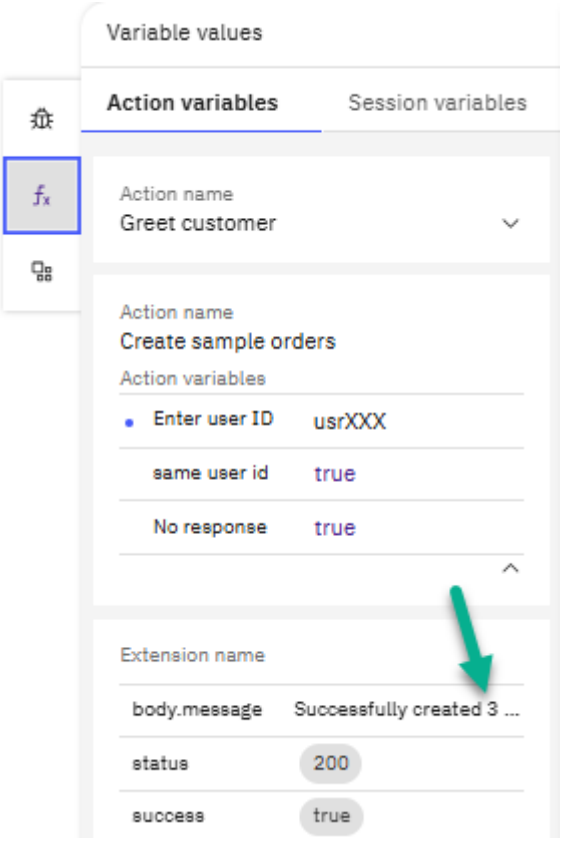
_4. Note the Assistant's response. The AI Assistant invoked the CreateSampleOrders Extension, which created three test orders.



_5. Click the **Variable values** (fx) icon.



_6. Note that the Inspector shows all the Action variables, including the message that the Extension invocation generated three sample orders for the usrXXX.

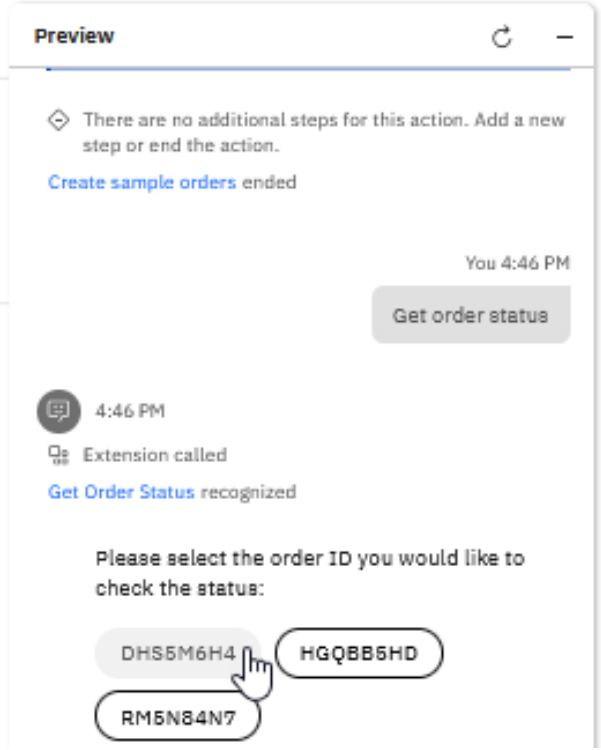


_7. Click the **Variable values** (fx) icon again to close the Inspector.

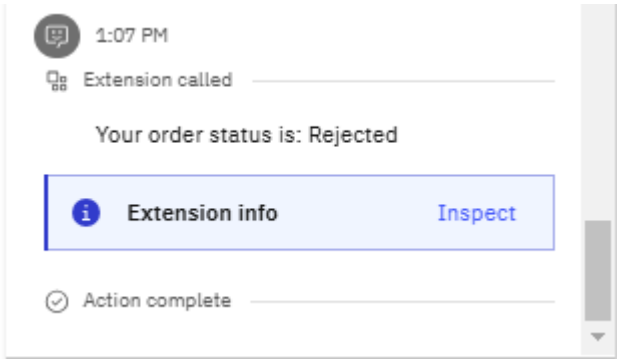
3.5.3 Test the Get Order Status Action

_1. In the Chat, enter **Get order status** and press **enter.**

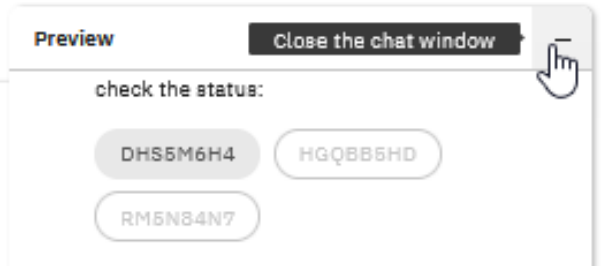
_2. You should now see the three sample orders you created in the previous step. **Click a bubble** to select one of the orders. Note that the format of your order IDs may differ from the following screenshots.



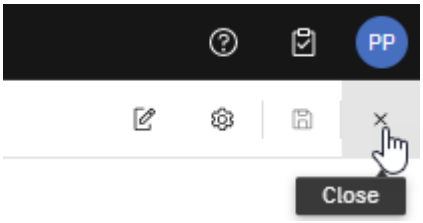
_3. You should now see the status of the selected order. In the screenshot below, you will see the status of Rejected. You may see a different status in your lab as the sample order generation is random.



_4. Click **– icon** to close the Chat window.



_5. Click the **X icon** to close the Get Order Status Action.

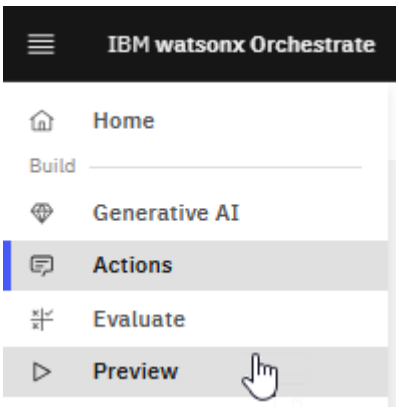


3.6 Integrate Web Chat with Web Page

The web chat integration provides an assistant interface that can integrate with your website without requiring the time and effort to build your custom user interface. It can help users start the conversation with common questions or tasks.

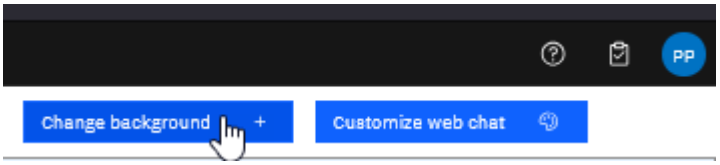
Specifically, we will add web chat to Focus Corp's internal employee portal to provide a more straightforward way to order computer accessories instead of using the more powerful but also more complex web application.

_1. Click **Preview** from the left menu bar.

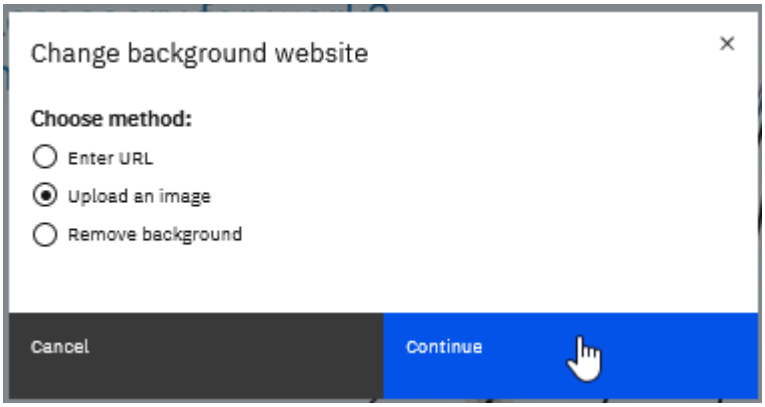


3.6.1 Change Assistant Web Page Background

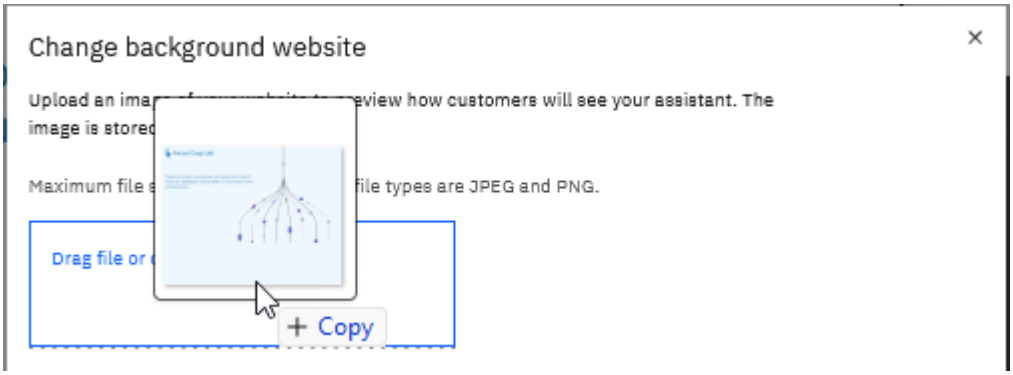
_1. Click **Change background +.**



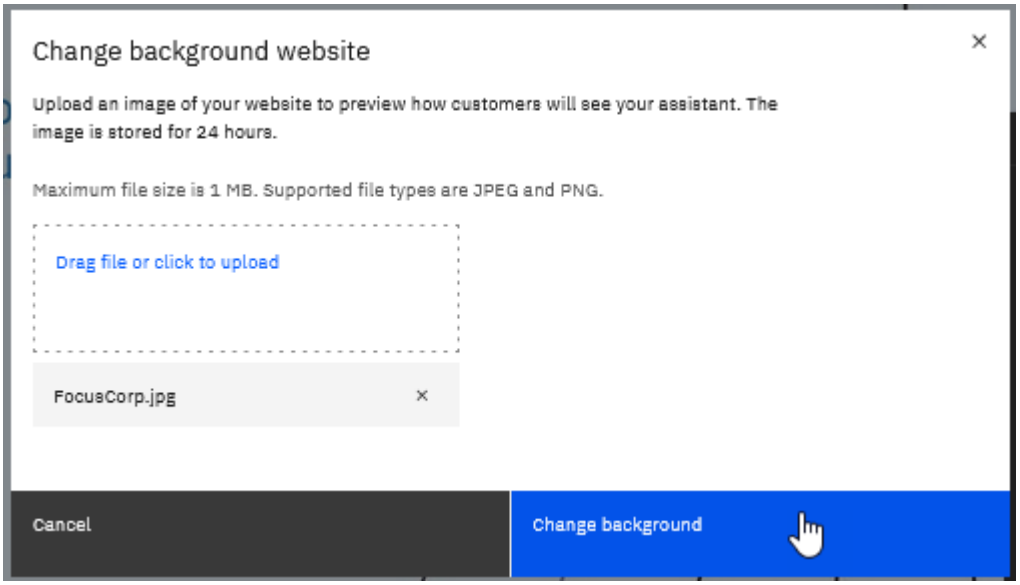
_2. Select **Upload an image** and click **Continue**.



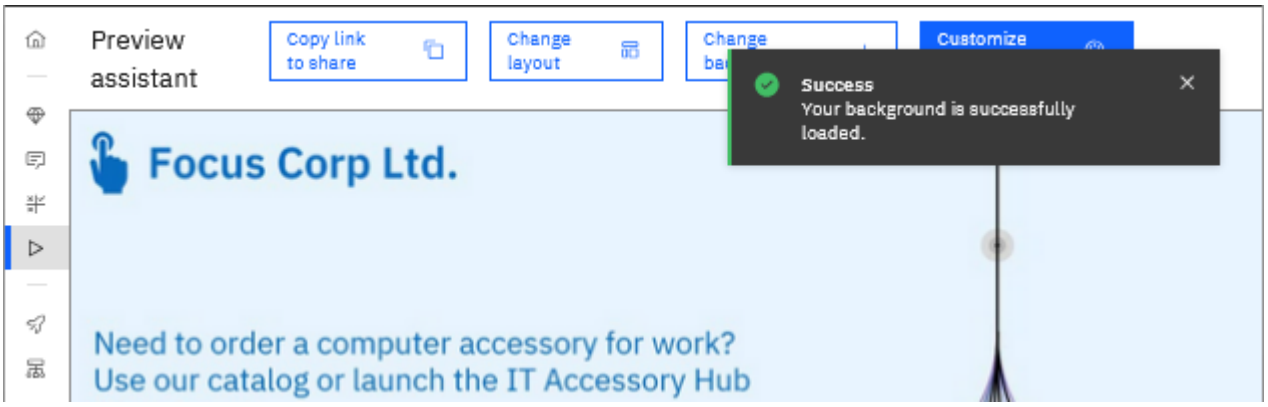
_3. Drag and drop **FocusCorp.jpg** file to *Change background website*.



_4. Click **Change background**.

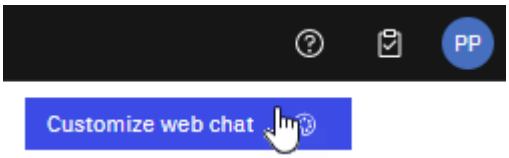


You should now see the Focus Corp Ltd. background



3.6.2 Customize Assistant Style

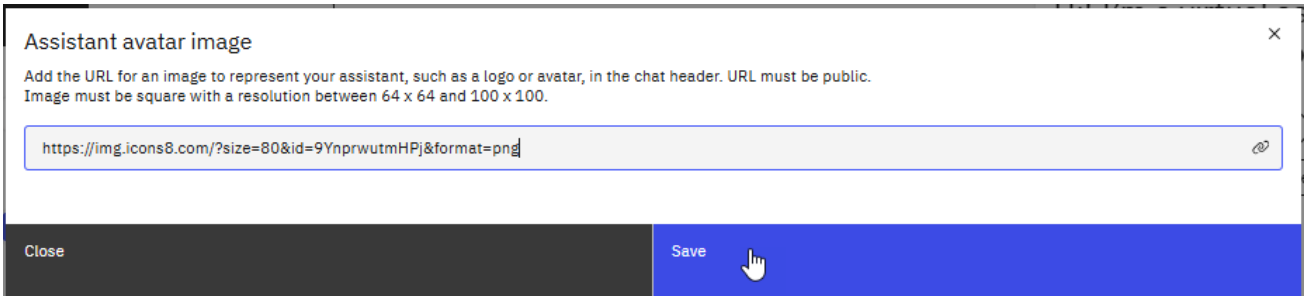
_1. Click **Customize web chat.**



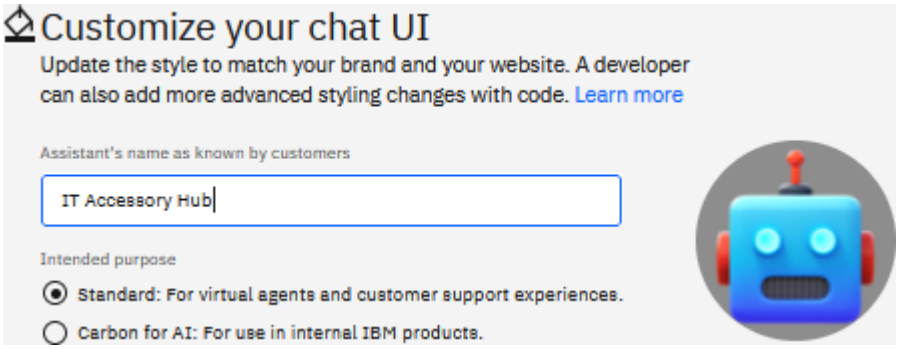
_2. Click the **Add an avatar image** link.

_3. For an *Assistant avatar image,* copy the URL below and click **Save**.

https://img.icons8.com/?size=80&id=9YnprwutmHPj&format=png

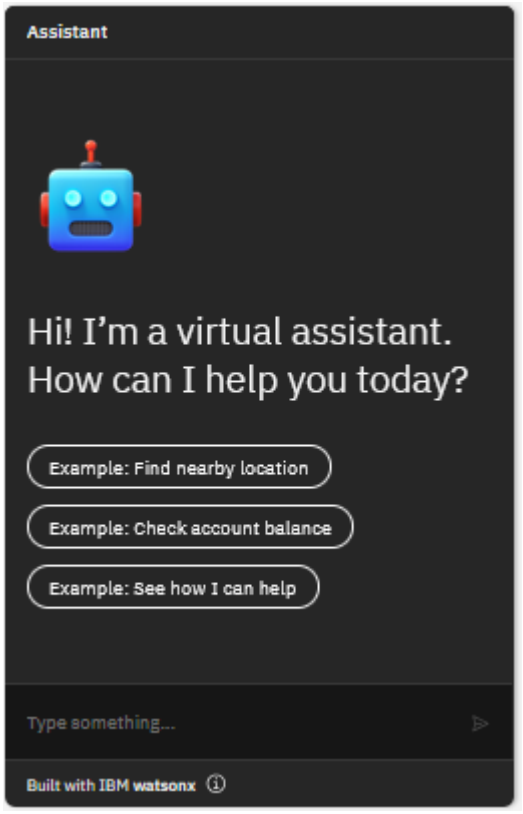


_4. For the *Assistant's name as known by customers,* enter **IT Accessory Hub.**



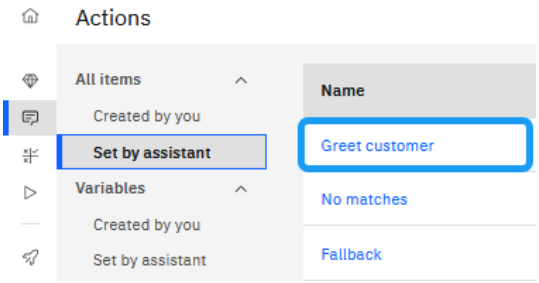
_5. For *Choose a theme* select **Dark**.

Your customized web chat should look similar to this:



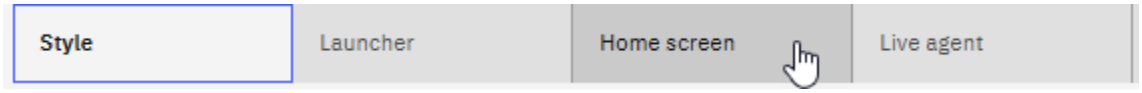
3.6.3 Customize Home Screen

On the Home screen tab, you can configure the contents of the home screen, which welcomes customers and helps them start the conversation with the AI Assistant. Recall that you customized the *Greet customer* system action:

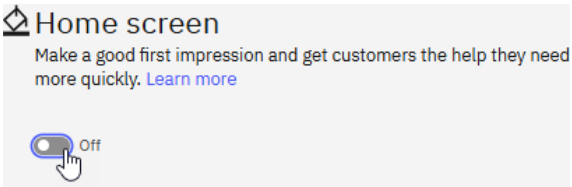


Let's enable it on the AI Assistant's home screen.

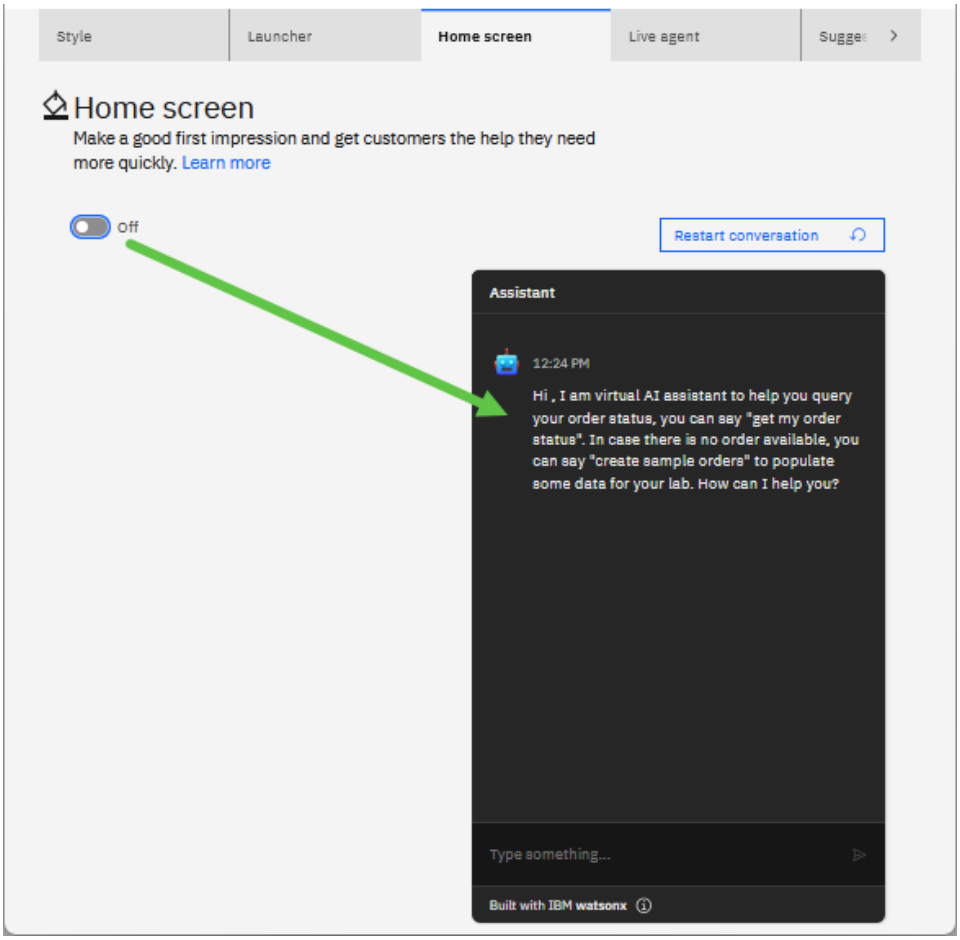
_1. Click the **Home screen** tab.



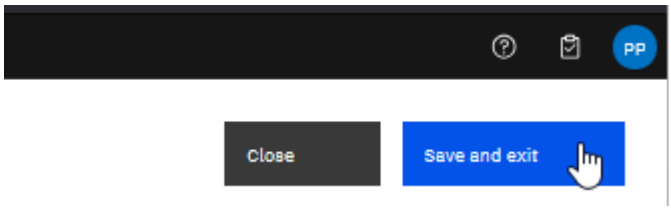
_2. Turn the Home screen feature **Off**.



_3. Notice that now we see the greeting you confirmed in the Greet customer action.



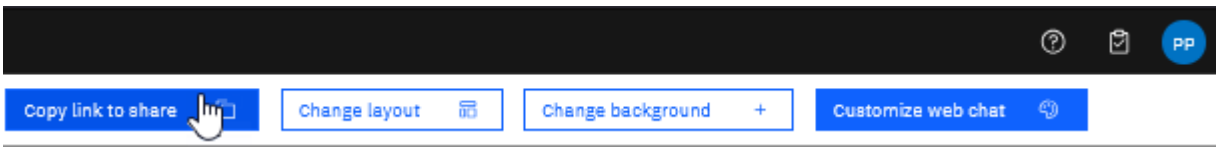
_4. Click **Save and exit**.



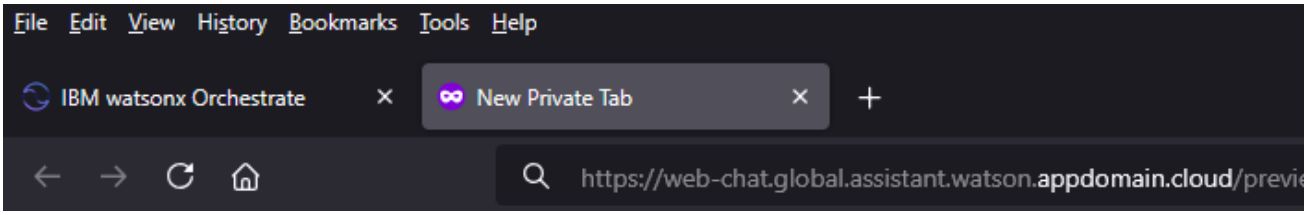
3.7 Test Your Web Chat

You can share an unauthenticated version of your AI Assistant with your users by sending them a link. The link opens a sample web page with an interactive web chat widget where you can test out your AI Assistant as if you were a customer. Web chat users can test your in-progress AI Assistant without access to wxO AI Assistant.

_1. Click **Copy link to share** to copy the AI Assistant web page to the clipboard.



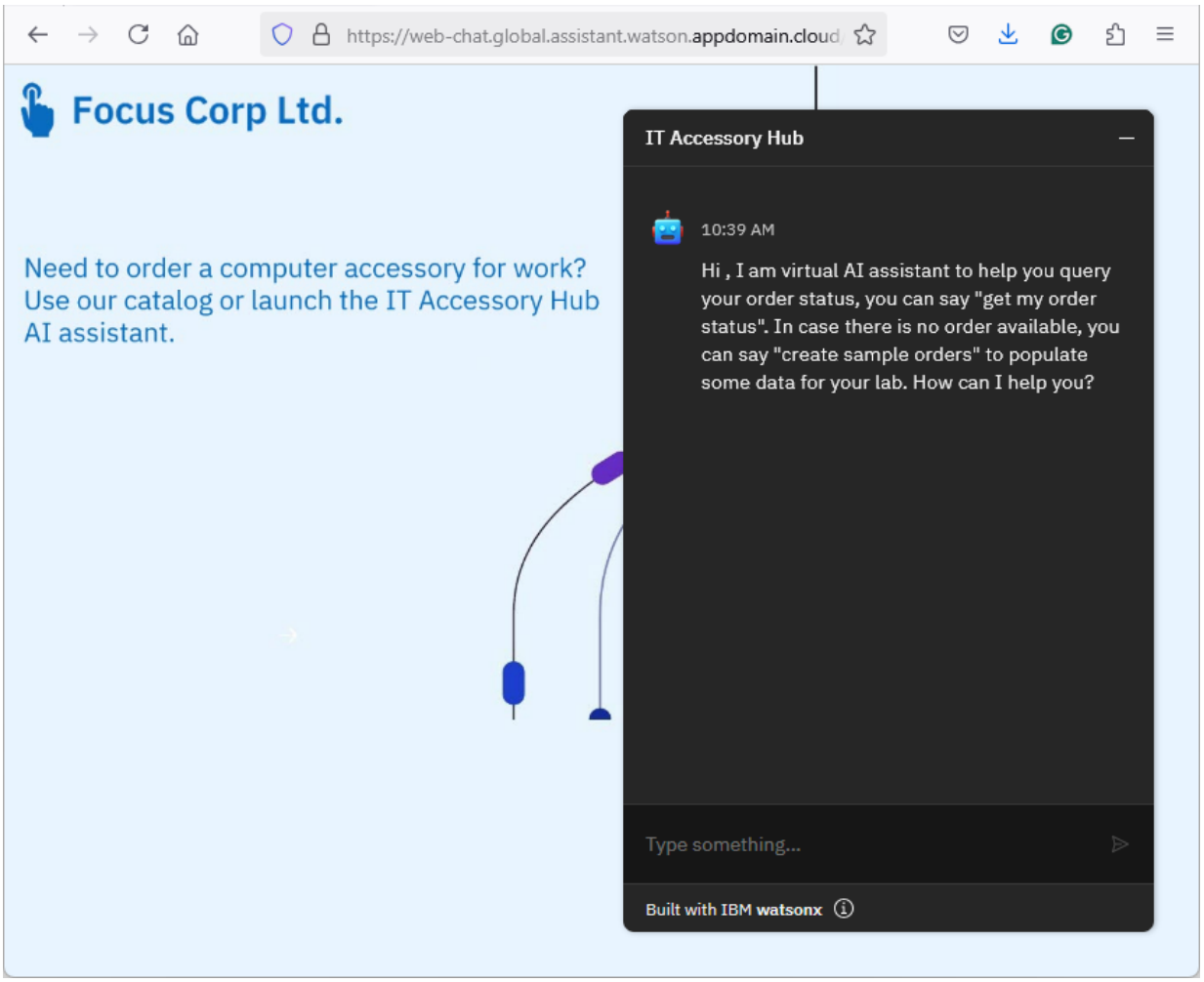
_2. Open a **new tab** in your Web browser, **paste the link from the clipboard**, and press **enter**.



_3. Click the web chat blue icon.

_4. Verify that you see the web chat with your modifications:

- Chatbot name: IT Accessory Hub
- Black Theme
- Chatbot icon
- Background image



Congratulations, you have completed this lab!

K27,$&6 ! 1%%,6$/! ,0 &56

RzPzD29&510 &17R6&56 O&675,$7&%O,(+76 w 86&}%83/,$! 7,21 25%,6$/2685& 5&675,$7&%#< DP> >AM P$+&%8/& @2175! $7: ,7+ F?J z

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information.

Q+,6 %2$80 &17,6 %,675,#87&%Ç! 6 ,6Ç: ,7+287! 1<: ! 55! 17<} &,7+&5 &; 35&66 25,0 3/,&%zF1 12 &9&17} 6+! // F?J #&/,! #/&'25! 1<%! 0 ! (&! 5,6,1('520 7+& 86& 2' 7+,6 ,1'250 ! 7,21} ,1$/8%,1(#87127/,0 ,7&%72} /266 2' %! 7! } #86,1&66 ,17&55837,21} /266 2' 352„ 725/266 2' 233257 81,7<z IBM products and services are warranted per the terms and conditions of the agreements under which they are provided. The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

IBM products are manufactured from new parts or new and used parts.
In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."

>1<67! 7&0 &176 5&(! 5%,1(F?J Å6 '8785& %,5&$7,21} ,17&1725352%8$73/! 16! 5& 68#-&$772 $+! 1(& 25 : ,7+%5! : ! / : ,7+287127,$&z

Performance data contained herein was generally obtained in a controlled,
isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the Customer's responsibility to ensure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the Customer's business and any actions the Customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the Customer follows any law.

K27,$&6 ! 1%%,6$/! ,0 &56 ë@217,18&%6

Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. F?J &; 35&66/<%,6$/! ,0 6! // : ! 55! 17,&6} &; 35&66&%25,0 3/,&%} ,1$/8%,1(#87127/,0 ,7&%72} 7+& ,0 3/,&%: ! 55! 17,&6 2' 0 &5$+! 17! #,/,7<! 1%„ 71&66 '25! 385326&z

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.