



# Microsoft Cloud for Healthcare in a Day

## Lab 04: Azure Health Bot

Step-by-Step Lab

90 minutes

April 2022

# Contents

<b>Overview .....</b>	<b>3</b>
Learning Objectives.....	3
Prerequisites .....	3
Azure Health Bot.....	3
Industry Prioritized Scenarios .....	4
Healthcare Story .....	4
<b>Exercise 1: Set Up Azure Health Bot .....</b>	<b>5</b>
Task 1: Install Azure Health Bot in Azure Subscription .....	5
Task 2: Update Azure Health Bot Settings to Enable Dynamics 365 Integration.....	9
Task 3: Obtain Azure Application ID.....	13
<b>Exercise 2: Configure Omnichannel Live Chat.....</b>	<b>15</b>
Task 1: Assign Omnichannel Agent Security Role .....	15
Task 2: Create Health Bot User in Dynamics 365 Customer Service.....	19
Task 3: Create and Configure Human Agent Queue .....	25
Task 4: Create Workstream with Context Variables and Routing Rules.....	28
<b>Exercise 3: Embed Health Bot in Power Apps Portal .....</b>	<b>40</b>
<b>Exercise 4: Extend Azure Health Bot with Custom Scenarios .....</b>	<b>43</b>
Task 1: Create MCH_PatientService Scenario.....	44
Task 2: Create MCH_PatientServiceWelcome Scenario .....	60
Task 3: Configure Welcome Scenario as Automatic .....	64
Task 4: Test Health Bot Escalation from Power Apps Portal to Dynamics 365 Omnichannel .....	65
<b>Summary .....</b>	<b>69</b>

# Overview

## Learning Objectives

In this lab, you will learn to do the following:

- Set up Azure Health Bot
- Configure Dynamics 365 Customer Service Omnichannel Live Chat
- Embed Azure Health Bot in a Power Apps Portal
- Extend Azure Health Bot with custom scenarios

## Prerequisites

- None

## Azure Health Bot

The Azure Health Bot Service is a cloud platform that empowers developers in healthcare organizations to build and deploy their compliant, AI-powered virtual health assistants and health bots, that help them improve processes and reduce costs. It allows you to offer your users *intelligent* and *personalized access* to health-related information and interactions through a natural conversation experience.

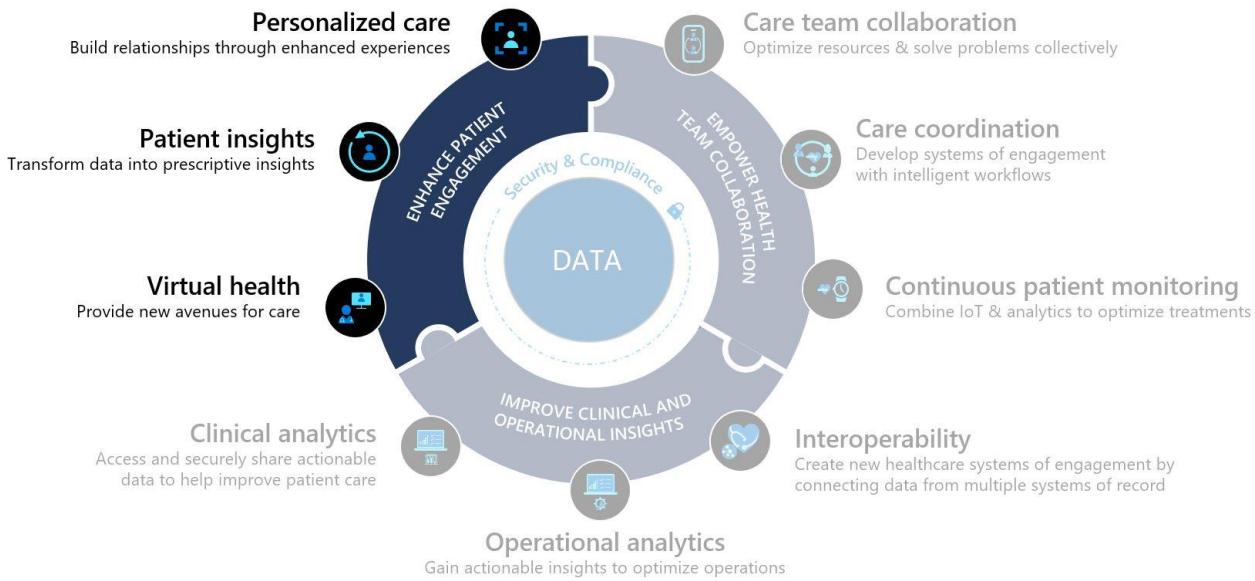
Using the service, healthcare organizations can build a "health bot instance" and integrate it with their systems that patients, nurses, doctors, and other representatives interact with. Building an instance allows you to:

- Improve processes
- Improve services
- Improve outcomes
- Reduces cost

The Health Bot Service contains a **built-in medical database**, including **triage protocols**. You can also extend a health bot instance to include your own scenarios and integrate with other IT systems and data sources. To learn more about Azure Health Bot, you can reference this Microsoft Docs article: [Azure Health Bot Overview](#).

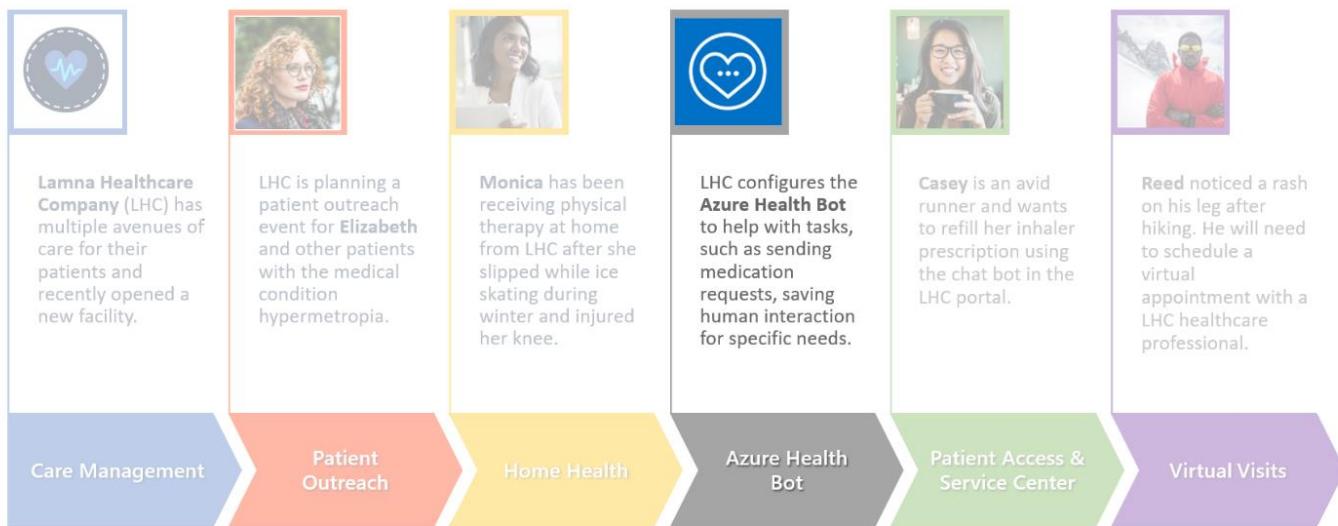
## Industry Prioritized Scenarios

The Azure Health Bot focuses on the **Enhance patient engagement** priority scenario by creating a virtual bot health option to allow for new avenues of care with embedded insights.



## Healthcare Story

This lab will focus on Lamna Healthcare Company.



As part of their digital transformation efforts, Lamna Healthcare Company is seeking to streamline their patient engagement capabilities by implementing Azure Health Bot to help improve processes and services, including receiving medication requests. By allowing patients to interact with this service, Lamna Healthcare Company will move one step closer to their goal of improving patient outcomes while reducing overall costs.

In this lab, you will play the role of a Lamna Healthcare IT developer and configure Azure Health Bot for a medication refill scenario.

# Exercise 1: Set Up Azure Health Bot

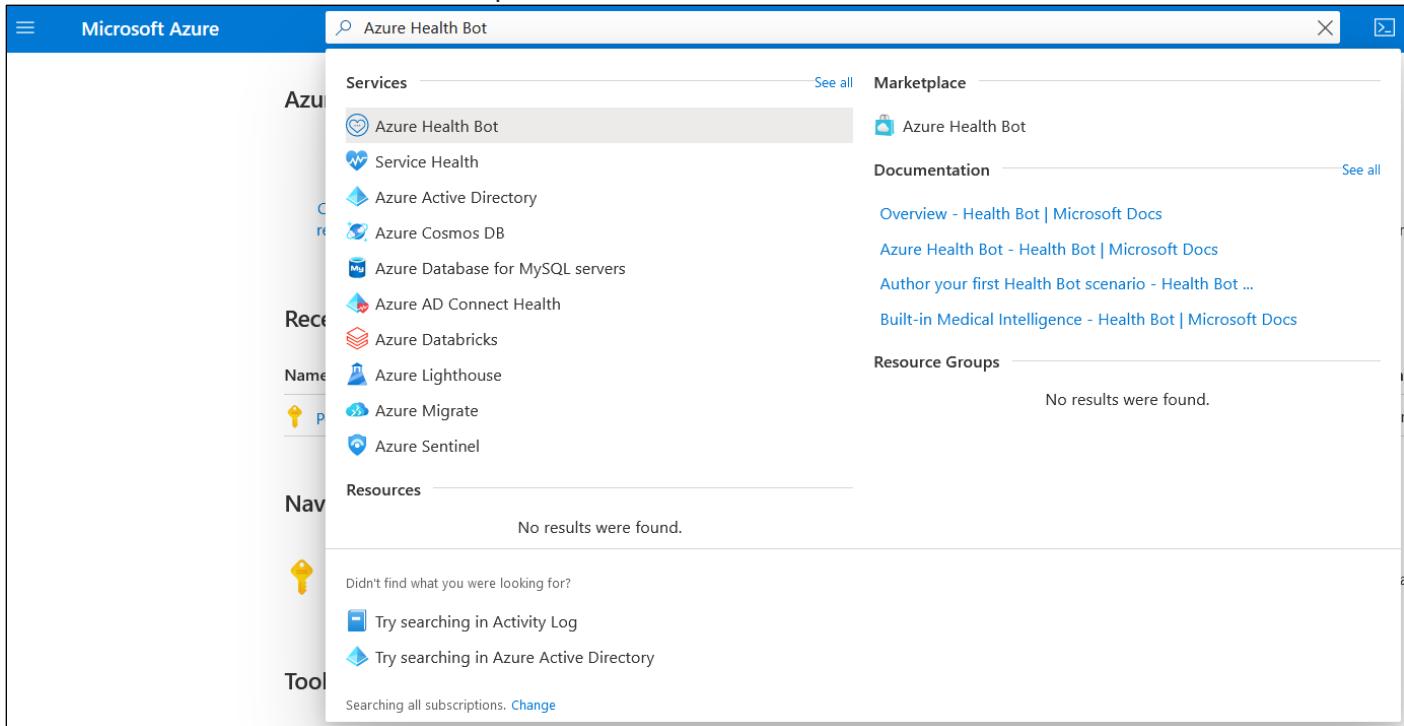
In this exercise, you will do the following:

- Set up Health Bot from Azure Portal
- Configure and enable the integration between Dynamics 365 Omnichannel and Health Bot
- Configure and enable Bot channel to obtain a Bot Id

**Azure Health Bot** empowers developers in healthcare organizations to build and deploy AI-powered, compliant, conversational healthcare experiences at scale. It combines built-in medical database with natural language capabilities to understand clinical terminology and can be easily customized to support your organization's clinical use cases. The service ensures alignment with industry compliance requirements and is privacy protected to HIPAA standards. To learn more about Azure Health Bot, please reference this [Azure Health Bot documentation](#).

## Task 1: Install Azure Health Bot in Azure Subscription

1. While logged in to your Microsoft 365 tenant, open a new tab in your internet browser incognito or in-private mode and navigate to Azure Portal at <https://portal.azure.com/>
2. Search for **Azure Health Bot** in the top search bar and **select** it from the search results.



3. Click **Create** button to create a new Azure Health Bot instance.

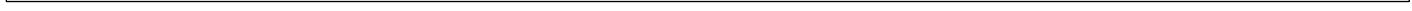
Microsoft Azure Search resources, services, and docs (G+ /)

Home > Azure Health Bot

Microsoft (PowerPlatformOpenHacks.onmicrosoft.com)

[+ Create](#) [Manage view](#) [Refresh](#) [Export to CSV](#) [Open query](#) | [Assign tags](#) | [Feedback](#)

Filter for any field... Subscription == all Resource group == all Location == all Add filter



4. You will be redirected to the Azure Health Bot page. Enter the following information:
  - a. **Subscription:** PowerPlatformOpenHacks Subscription
  - b. **Resource Group:** IndustryLabs
  - c. **Name:** iaduser[x]-healthbot (e.g., iaduser01-healthbot, using your assigned user)
  - d. **Region:** East US
  - e. **Plan:** Free (F0)

## Azure Health Bot

Basics Tags Review + create

Basics

**Project details**

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* ⓘ PowerPlatformOpenHacks Subscription

Resource group \* ⓘ IndustryLabs  
[Create new](#)

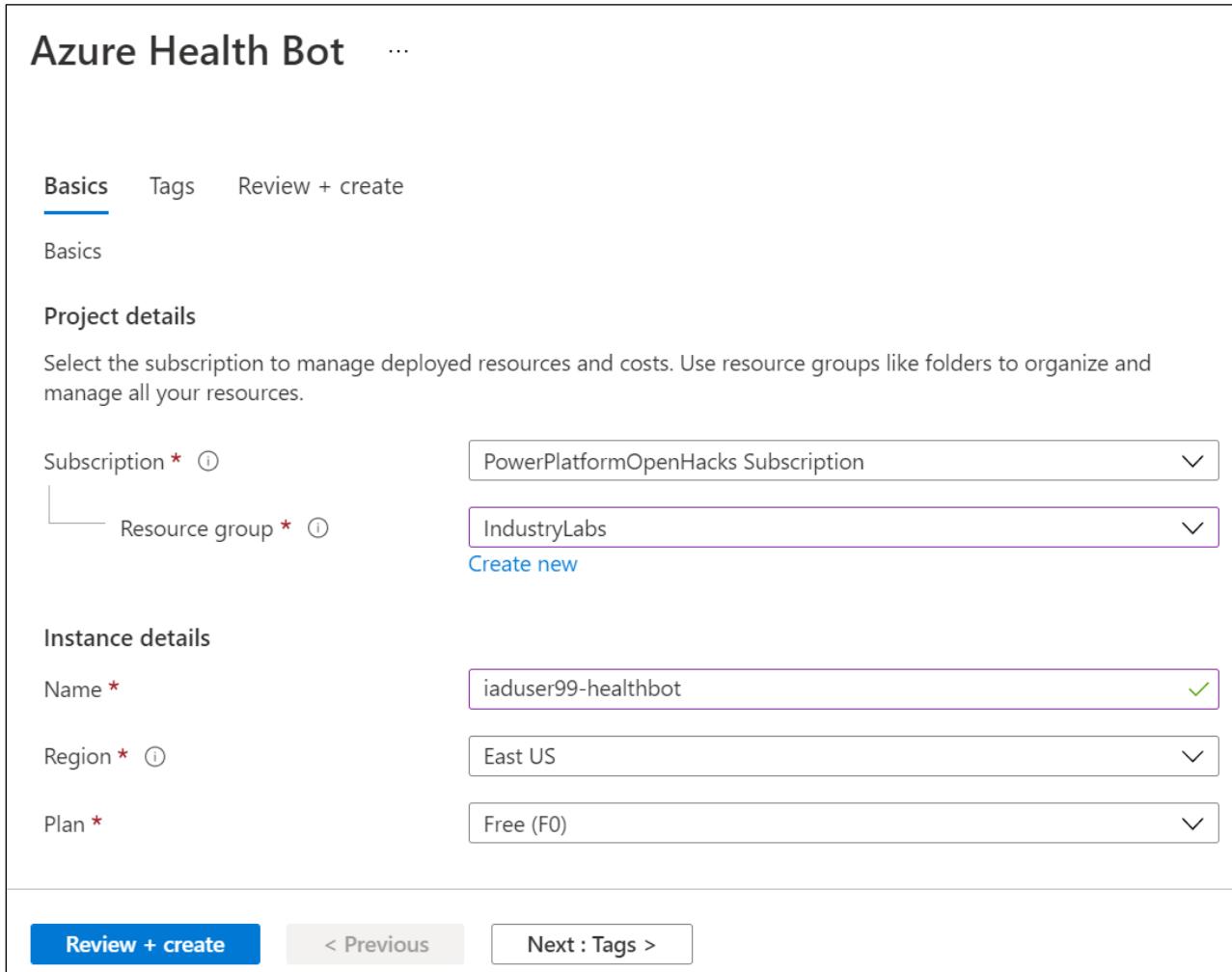
**Instance details**

Name \* iaduser99-healthbot

Region \* ⓘ East US

Plan \* Free (F0)

[Review + create](#) [< Previous](#) [Next : Tags >](#)



5. Select **Review + Create**.

- On the Review and create page, verify your details are correct as Azure validates your Health Bot. After validation passes, the create button will become enabled. Click **Create**.

*Note: It will take few seconds to run the backend process before the Create button is enabled.*

Azure Health Bot

Validation Passed

Basics Tags Review + create

TERMS

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. See the [Azure Marketplace Terms](#) for additional details.

**Basics**

Subscription	PowerPlatformOpenHacks Subscription
Resource group	IndustryLabs
Name	iaduser99-healthbot
Region	East US
Plan	Free (f0)

**Create** < Previous Next Download a template for automation

- You will be redirected to the **Deployment** page for your new Azure Health Bot.

Home >

Microsoft.HealthBot-20210915192409 | Overview

Deployment

Search (Ctrl+ /) <> Delete Cancel Redeploy Refresh

We'd love your feedback! →

Overview Inputs Outputs Template

Deployment is in progress

Deployment name: Microsoft.HealthBot-20210915192409  
Subscription: PowerPlatformOpenHacks Subscription  
Resource group: IndustryLabs

Start time: 9/15/2021, 7:31:03 PM  
Correlation ID: c5cf3406-622d-4aff-8668-134158fe55f9

Deployment details (Download)

Resource	Type	Status	Operation details
iaduser99-healthbot	Microsoft.HealthBot/healthBots	Created	<a href="#">Operation details</a>

8. When deployment is complete, the **Go to resource** button will enable. Please wait until deployment is complete for the Azure Health Bot, then select **Go to resource** when enabled.

The screenshot shows the Microsoft HealthBot Overview page for a deployment named "Microsoft.HealthBot-20210915192409". The status is "Your deployment is complete". Deployment details include a deployment name, subscription, and resource group. A "Go to resource" button is visible at the bottom.

9. You will be redirected to the **Resource** page for your new Azure Health Bot. Click the **Management portal** link in the Essential section to open your Azure Health Bot instance configuration page.

*Note: Please copy this Management portal link and store it to access the Health Bot later.*

The screenshot shows the Azure Resource page for the "lammahealthcare-bot-gcy" instance. In the "Essentials" section, the "Management portal" link is highlighted with a red box, containing the URL: <https://eastus.healthbot.microsoft.com/account/lammahealthcare-bot-gcy...>.

10. You will be navigated to your new Azure Health Bot instance homepage.

The screenshot shows the "Welcome to your Health Bot Instance" page. It features three main sections: "Select a template scenario", "Create a new scenario", and "Configure built-in capabilities". Each section includes a brief description and a "Manage from configuration section" link.

**Congratulations!** You have successfully created a new Health Bot instance in your Azure tenant.

## Task 2: Update Azure Health Bot Settings to Enable Dynamics 365 Integration

1. On the Azure Health Bot homepage, **expand** the side navigation bar to see the sitemap labels.

The screenshot shows the Azure Health Bot homepage. A red box highlights the three-line icon in the top-left corner of the sidebar, indicating it was expanded. The sidebar contains several icons and labels: Scenarios, Language, Configuration, Integration, Analytics, Users, and Resources. The main content area features a large heart icon and the text "Welcome to your Health Bot Instance". Below this, there are three main options: "Select a template scenario", "Create a new scenario", and "Configure built-in capabilities". Each option has a brief description and a corresponding icon. At the bottom of the main content area are two blue links: "Open scenarios template catalog" and "Create new scenario".

After expanding, you will see the sitemap labels next to the icons.

This screenshot is identical to the one above, but the side navigation bar is now fully expanded, showing the sitemap labels next to each icon. The labels are: Scenarios, Language, Configuration, Integration, Analytics, Users, and Resources. The rest of the interface, including the main content area with its welcome message and scenario creation options, remains the same.

2. Select **Configuration > Conversation** on the navigation bar.

The screenshot shows the Azure Health Bot navigation bar. The sidebar is expanded, showing the same seven categories: Scenarios, Language, Configuration, Integration, Analytics, Users, and Resources. The "Configuration" category is expanded, revealing its sub-categories: Medical, Compliance, and Conversation. The "Conversation" item is highlighted with a red box. The main content area is mostly blank, indicating the user is currently viewing the configuration settings for the Conversation section.

3. You will be landed in the **Interactions** tab.

The screenshot shows the Azure Health Bot interface. The top navigation bar includes the 'Azure Health Bot' logo, user name 'iaduser99-healthbot', and a 'Refresh' button. The left sidebar has a dark theme with white text and icons. It lists 'Scenarios', 'Language', 'Configuration' (which is expanded to show 'Medical', 'Compliance', and 'Conversation'), 'Integration', 'Analytics', 'Users', and 'Resources'. The 'Interactions' tab is selected and highlighted in blue. The main content area is titled 'Interactions' and contains the sub-instruction 'Configure built-in scenarios that interact with your end user.' Below this is a section titled 'Global defaults' which lists several default messages:

- Default reply for utterances that are not understood: Sorry, it seems I can't answer this.
- Default message when returning from interrupting scenarios: Now back to the previous topic...
- Default Error Message: Oops. Something went wrong and we need to start over.
- Default retry message (number prompts): I didn't recognize that as a number. Please enter a number.

4. Select **Human Handoff** tab in the Conversation settings.

The screenshot shows the Azure Health Bot interface with the 'Human Handoff' tab selected. The top navigation bar and sidebar are identical to the previous screenshot. The main content area is titled 'Human Handoff' and contains the sub-instruction 'Configure scenarios that allow handoff to human agents.' A 'Learn more' link is present. Below this is a section titled 'Human Handoff' with a toggle switch labeled 'Disabled' and an eye icon. It includes three configuration fields:

- End user timeout: Set to 10 minutes.
- Agent timeout: Set to 15 minutes.
- Waiting message: A placeholder message: 'We are connecting you to an agent. Please wait...'

5. Scroll to the bottom of the **Human Handoff** page. Under **Dynamics 365 Omnichannel**, toggle **Enabled** for **Bridge Messages**. This is required to allow communication and bridge messages between the Azure health Bot and Dynamics 365 Omnichannel for Customer Service.

Interactions   Navigation   Spelling   **Human Handoff**   Refresh

Agent connection message (2 of 2) ⓘ  
You can start chatting with the agent.

End of conversation message ⓘ  
Agent (agentName) has left the conversation.

All agents unavailable message ⓘ  
Sorry, no agents are currently available.

Connection error message ⓘ  
An error occurred while connecting you to an agent. Please try again later.

**Agent Authentication (Microsoft Teams)** ⓘ

Active Directory Tenant ID ⓘ  
Enter your active directory tenant ID

Azure Active Directory Group Object ID ⓘ  
Enter your authorised agent group ID

Application (client) ID ⓘ  
Enter your application (client) ID

Application (client) Secret ⓘ  
Enter your application (client) secret

**Online Meetings (Microsoft Teams)**

Default Meeting Organizer Object ID (optional) ⓘ  
Enter an object ID for default meeting organizer (optional)

**Dynamics 365 OmniChannel** ⓘ

Bridge Messages

6. Click **Save** in the top right.

Interactions   Navigation   Spelling   **Human Handoff**   Refresh Save Cancel changes

### Human Handoff

Configure scenarios that allow handoff to human agents. [Learn more](#)

7. Now let's enable the Health Bot for **Microsoft Teams** Channel.

8. Navigate to **Integration > Channels**.

- Scenarios >
- Language >
- Configuration >
- Integration**
  - Data connections
  - Authentication
  - Skills
  - Secrets
  - Channels**
- Analytics** >
- Users >
- Resources >

9. In the Channels list, select the toggle to **enable Microsoft Teams**.

The screenshot shows a list of channels for a Healthcare bot. The columns are Active, Channel, and Actions. The Microsoft Teams row has its Active toggle switched on, indicated by a blue circle. Other channels like Web Chat, DirectLine, Twilio, Facebook, Telegram, Alexa (preview), and WhatsApp (via Twilio - preview) have their toggles off, indicated by grey circles.

Active	Channel	Actions
<input checked="" type="checkbox"/>	Web Chat	<a href="#">View</a>
<input checked="" type="checkbox"/>	DirectLine	<a href="#">View</a>
<input checked="" type="checkbox"/>	Microsoft Teams	<a href="#">View</a> <a href="#">Test</a>
<input type="checkbox"/>	Twilio	
<input type="checkbox"/>	Facebook	
<input type="checkbox"/>	Telegram	
<input type="checkbox"/>	Alexa (preview)	
<input type="checkbox"/>	WhatsApp (via Twilio - preview)	

10. This will bring out a side window with your **Bot Id** information. **Copy and store** the BotId for later to use when creating the Dynamics 365 Application User.

The screenshot shows a modal dialog titled "Microsoft Teams Channel". It contains a message about connecting the bot to Microsoft Teams users through chat, a "Learn how" link, and a "Bot Id" section. The Bot Id is listed as "c7f90733-d6e3-4f3d-a95a-f28cfbc9b0b1" with a copy icon next to it. At the bottom are "Save" and "Close" buttons.

11. Select **Save**. This should enable Teams channel and your Microsoft Teams toggle should reflect accordingly.

The screenshot shows the same list of channels as before, but now the Microsoft Teams row has its Active toggle switched on, indicated by a blue circle, matching the state in the previous screenshot.

Active	Channel	Actions
<input checked="" type="checkbox"/>	Web Chat	<a href="#">View</a>
<input checked="" type="checkbox"/>	DirectLine	<a href="#">View</a>
<input checked="" type="checkbox"/>	Microsoft Teams	<a href="#">View</a> <a href="#">Test</a>
<input type="checkbox"/>	Twilio	
<input type="checkbox"/>	Facebook	
<input type="checkbox"/>	Telegram	
<input type="checkbox"/>	Alexa (preview)	
<input type="checkbox"/>	WhatsApp (via Twilio - preview)	

**Congratulations!** You completed the Azure Health Bot settings for integration with Microsoft Teams and Dynamics 365 Omnichannel for Customer Service.

## Task 3: Obtain Azure Application ID

In this task, you will be using an Azure Application ID already created in our Azure tenant called "**MCH Application Id**". Registering this Id establishes a trusted relationship between your Dynamics 365 app and the Microsoft identity platform. Using this Id, you will later create a Dynamics 365 Application User to bridge the authentication between Azure Health Bot and Power Apps.

1. Navigate back to the Azure Portal and search for **App Registrations** in the Search box.

The screenshot shows the Azure Portal search interface. The search bar at the top contains the text "App Registrations". Below the search bar, there are two main sections: "Services" and "Marketplace". The "Services" section is expanded, showing a list of services: App registrations, Event Grid Partner Registrations, App Configuration, App proxy, App Services, Function App, Application gateways, Application groups, Application Insights, and Application Services. The "App registrations" item is highlighted with a gray background. To the right of the services list, there is a "Documentation" section with links to "Quickstart: Register an app in the Microsoft identity ...", "Best practices for Azure AD application registration ...", "Remove limits on creating app registrations - Azure AD ...", and "Protected web API app registration - Microsoft identity ...". Below the documentation is a "Resource Groups" section with a note "No results were found.". At the bottom of the search results, there is a message "Didn't find what you were looking for?" followed by two suggestions: "Try searching in Activity Log" and "Try searching in Azure Active Directory". A note "Searching all subscriptions. Change" is also present at the bottom.

2. You will be landed in the App registration homepage on the Owned applications tab.

The screenshot shows the "App registrations" page in the Microsoft Azure portal. The top navigation bar includes the Microsoft Azure logo, a search bar, and links for "Home", "App registrations", and "...". Below the navigation is a toolbar with buttons for "New registration", "Endpoints", "Troubleshooting", "Refresh", "Download", "Preview features", and "Got feedback?". The main content area has three tabs: "All applications", "Owned applications" (which is underlined in blue), and "Deleted applications (Preview)". A search bar below the tabs contains the placeholder text "Start typing a name or Application ID to filter these results". At the bottom of the page, a message states "This account isn't listed as an owner of any applications in this directory." followed by a blue button labeled "View all applications in the directory".

3. Select the **All applications** tab.

The screenshot shows the 'App registrations' page in the Azure portal. The top navigation bar includes 'Home >', 'App registrations', and three more items. Below the navigation is a toolbar with 'New registration', 'Endpoints', 'Troubleshooting', 'Refresh', and 'Download'. A search bar says 'Start typing a name or Application ID to filter these results'. Under the tabs 'All applications' (which is underlined), 'Owned applications', and 'Deleted applications (Preview)', there is a list of four entries, each with a small 'MC' icon and the text 'Microsoft CRM Portals'.

4. To **search** for our Application Id, type “**MCH Application Id**” in the Search box.

The screenshot shows the 'App registrations' page with the search bar containing 'MCH Application Id'. The results list shows one entry with a 'MA' icon and the text 'MCH Application Id'.

5. Select the **MCH Application Id** app registration resource. **Copy and store** the **Application (client) ID** for later to use when creating the Dynamics 365 Application User.

*Note: ID values have been removed in the screenshot for privacy purposes.*

The screenshot shows the 'MCH Application Id' app registration details page. On the left is a sidebar with 'Overview', 'Quickstart', 'Integration assistant', 'Manage', 'Branding', and 'Authentication'. The main area has a search bar, a toolbar with 'Delete', 'Endpoints', and 'Preview features', and a table of details. The 'Essentials' section includes fields for 'Display name' (MCH Application Id), 'Application (client) ID' (highlighted with a red box), 'Object ID', 'Directory (tenant) ID', and 'Supported account types' (My organization only).

**Congratulations!** You have successfully obtained the MCH Application ID from Application Registrations in the Azure Portal.

## Exercise 2: Configure Omnichannel Live Chat

In this exercise, you will be configuring live chat for **Dynamics 365 Omnichannel for Customer Service**.

Omnichannel for Customer Service offers a suite of capabilities that extend the power of Dynamics 365 Customer Service Enterprise to enable organizations to instantly connect and engage with their customers across digital messaging channels.

In the following tasks, you will complete the following:

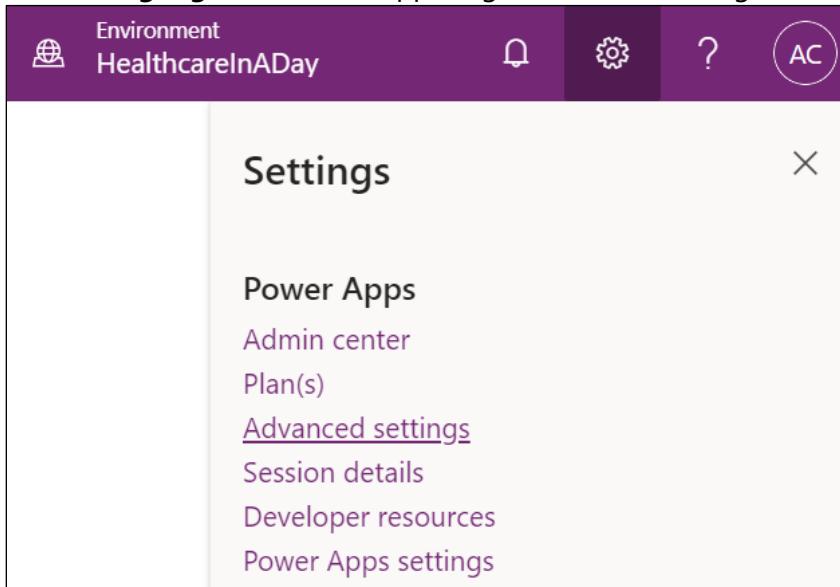
1. Assign Omnichannel Agent security role to your user
2. Create an Application User using the **MCH Application Id** and your **Bot Id**
3. Configure Queues and Workstreams for bot and agent users
4. Configure a Context Variable and Routing rule to route the message either to a Bot or Agent.

### Task 1: Assign Omnichannel Agent Security Role

1. While in the In-Private or Incognito window, navigate to [Power Apps](#).
2. Ensure the correct environment from the upper right **Environment** drop down is selected.



3. Select the **gauge icon** in the upper right corner and navigate to **Advanced Settings**.



4. A new window should open and navigate to Dynamics 365. It may take a while to load. If it's been longer than a minute, stop and reload the page. It should then load faster. It will land you in the Business Management section of Dynamics 365.

Dynamics 365 Settings Business Management SANDBOX

**Business Management**

Which feature would you like to work with?

 <b>Fiscal Year Settings</b> Set the start date, template, and display options for the fiscal year and fiscal period used for tracking sales goals.	 <b>Goal Metrics</b> Define and manage the kinds of goals that your organization tracks.
 <b>Business Closures</b> Create a list of holidays and other times when the business is closed.	 <b>Facilities/Equipment</b> Add facilities and equipment for service scheduling. Change information about resources or delete existing resources.
 <b>Queues</b> Create and manage service queues, and manage the membership of private queues. Establish criteria for automatic record creation and updates.	 <b>Resource Groups</b> Add new groups and new members to existing groups for service scheduling. Update group information and delete groups or group members.
 <b>Sales Territories</b> Create new sales territories and assign territory managers. Add and remove members, modify territory information, and delete territories.	 <b>Services</b> Add new services for service scheduling. Change service information and deactivate existing services.
 <b>Sites</b> Create new sites or office locations where service operations take place. Add and remove resources, change site information, or delete sites.	 <b>Subjects</b> Manage the subject hierarchy for your organization's products, literature, and articles.
 <b>Currencies</b> Add new currencies or change the exchange rates for existing currencies.	 <b>Connection Roles</b> Create, edit, and delete the standard labels used to define connections between records.
 <b>Automatic Record Creation and Update Rules</b> Create and manage rules for automatic record creation and updates. You can set up rules for either out-of-the-box entities or custom entities.	 <b>Rollup Queries</b> Go to your list of Rollup Queries that you can use to gather data about a group of related records.
 <b>LinkedIn Sales Navigator</b> Manage settings relating to LinkedIn Sales Navigator Integration	

5. On the top command bar next to Dynamics 365, select **Settings** to open the drop-down, then select **Security** in the third column under System.

Dynamics 365 Settings Business Management

**Settings**

<b>Business</b>	<b>Customization</b>	<b>System</b>
 Business Manageme...	 Customizations	 Administration
 Templates	 Solutions	 Security
 Product Catalog	 Microsoft AppSource	 Data Management
 Service Management	 Plug-In Trace Log	 System Jobs
 Sync Error	 Solutions History	 Document Manage...
		 Auditing

6. Under Security, select **Users**.

**Security**

Which feature would you like to work with?

 <b>Users</b> Add new users. Edit information about users and deactivate user records. Manage the teams, roles, and licenses assigned to users.
---

7. Switch the view drop down from Omnichannel Users to **Enabled Users** for the grid view so that your user will show in the list.

The screenshot shows a dropdown menu titled "Omnichannel Users". Under the heading "System Views", there is a list of user categories. The "Enabled Users" option is highlighted with a light blue background, indicating it is selected. Other options include "@Me", "Access Mode Interactive Users", "Administrative Access Users", "Administrators", "Agents", "All", "Application Users", "Associated Record Team Members", "Bot agents", "Bot Users", "By Me", "Disabled Users", "Disabled users consuming licenses", "Enabled Users" (which is selected), and "Full Access Users".

8. While in the Enabled User list, scroll to **find your user** or use the **Search bar**.

*Note: If you are in an official training, search for you assigned user – IAD User [x]*

The screenshot shows a search results grid in Dynamics 365. The search term "iad" is entered in the search bar at the top right. The results table has columns for Full Name, Position, Main Phone, Business Unit, Site, Title, and Primary Email. One result is shown: "IAD User 01" with the email "unq0ed694338a62465...".

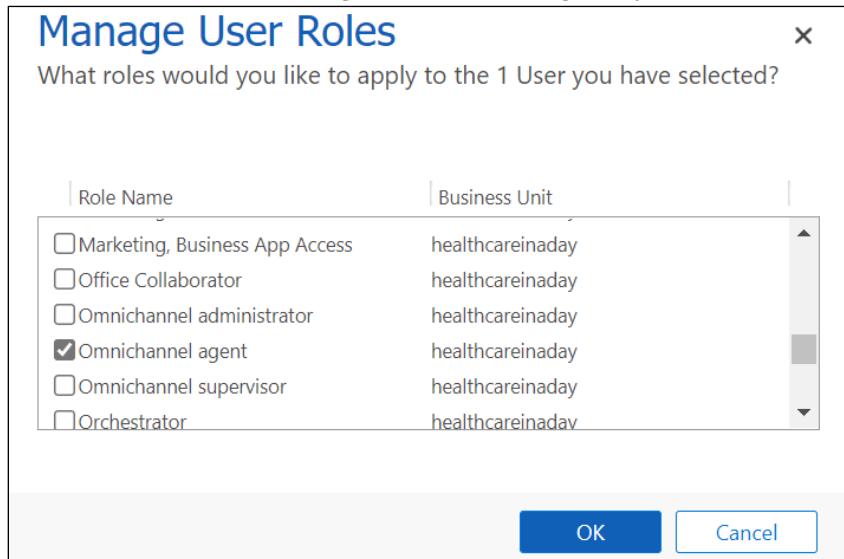
9. Select your user for the training and select **Manage Roles** on the top command bar.

The screenshot shows the Dynamics 365 top command bar. The "MANAGE ROLES" button is highlighted with a light blue background, indicating it is selected. Other buttons visible include "NEW", "EDIT", "APPROVE EMAIL", "REJECT EMAIL", "PROMOTE TO ADMIN", "CHANGE BUSINESS UNIT", and the "Dynamics 365" logo. The word "SANDBOX" is displayed on the right side of the bar.

The screenshot shows a search results grid in Dynamics 365. The user "IAD User 01" is selected, indicated by a checked checkbox in the first column. The results table has columns for Full Name, Position, Main Phone, Business Unit, and Site. The full name "IAD User 01" and the email "unq0ed694338a62465..." are visible.

10. Select the Omnichannel Agent roles to assign to your user and select **OK**.



**Congratulations!** You assigned the proper omnichannel agent role to your user to allow you to be a live agent in omnichannel.

## Task 2: Create Health Bot User in Dynamics 365 Customer Service

We need two users to configure in Omnichannel for Dynamics 365 Customer Service:

- **Health Bot User** – This is the Azure Health Bot user we created in the previous exercise.
- **Omnichannel Agent User** – This is your current user whom you are logged into Dynamics 365. This will allow you to be a live agent in Customer Service who receives messages from portal users through Azure Bot escalations. *Note: For official trainings, this is your assigned user, iaduser[x]*

In this task, you will create a **Bot User** which helps connect **Azure Health Bot** with **Omnichannel live Chat**.

1. Go to <https://admin.powerplatform.microsoft.com/>.
2. Select your Microsoft Cloud for Healthcare environment from the list

Environment	Type
HealthcareIndustryLabs	Sandbox

3. You will land on your environments detail page.

Environment URL	State
HealthcareIndustryLabs	Ready

Region	Refresh cadence
United States	Frequent

Type	Security group
Sandbox	[redacted]

Organization ID
[redacted]

4. Click the **Settings** button on the top command bar.

Open environment    **Settings**    Resources    Convert to production

Environments > HealthcareIndustryLabs

5. Expand **Users + permissions** and click **Application users**.

Search for a setting

- Product
  - Behavior, Features, Languages, Privacy + Security
- Business
  - Business closures, Calendar, Connection roles, Currencies
- Users + permissions**
  - Application users**
  - Business units
  - Hierarchy security
  - License To Role mapping
  - Mobile configuration
  - Positions
  - Security roles
  - Teams
  - Users
- Email
  - Email settings, Email tracking, Mailbox
- Integration
  - Document management settings, Sy
- Data management
  - Auto numbering, Automatic record c
- Encryption
  - Data encryption
- Resources
  - All legacy settings, Dynamics 365 Ap

6. Select (+) **New app user** button to create a new Application User.

Power Platform admin center

Environments > HealthcareIndustryLabs > Settings > **Application users**

Manage the non-interactive users in this environment so your apps can access Dataverse data. [Learn more](#)

Name	App ID	State
Dynamics Marketing Customer Experience Platform PR...	2220bbc4-4518-...	Active
FRE Omnichannel Omnichannel PVA Application	cd5f0174-51e9-4...	Active
Omnichannel Omnichannel for Customer Service	18cc9627-776c-4...	Active

**New app user**

7. Select (+) **Add an app** on the create screen that slides out on the right side.

Create a new app user ×

App \* \*  
+ Add an app

Business unit \* Select

Security roles(0) ✎

8. Paste the **Application ID** (the Application (client) ID you obtained in the Azure portal for the supplied MCH Application ID) into the search box and select the app from the list. Click **Add** at the bottom right.

← Add an app from Azure Active Directory ×

🔍 dfda9044-cb98-...  ×

There could be multiple reasons why your app may not show up in the list.  
[Learn more](#)

Name ↑	App ID
<span style="border: 1px solid red; border-radius: 50%; width: 20px; height: 20px; display: inline-block; vertical-align: middle;"></span> <span style="margin-left: 10px; border: 1px solid red; border-radius: 5px; padding: 2px;">MCH Application Id</span>	dfda9044-cb98-...

Add Cancel

9. Select a **Business unit** from the drop-down list (the options in the list will be unique for each Dynamics 365 environment). Click **Create** at the bottom right.

Create a new app user

App \*

MCH Application Id

Business unit \*

unqd8e...  
unqd8e...

Security roles(0)

Create Cancel

10. Return to the Dynamics 365 User page, switch the view to **Enabled Users**.

Omnichannel Users ▾

**System Views**

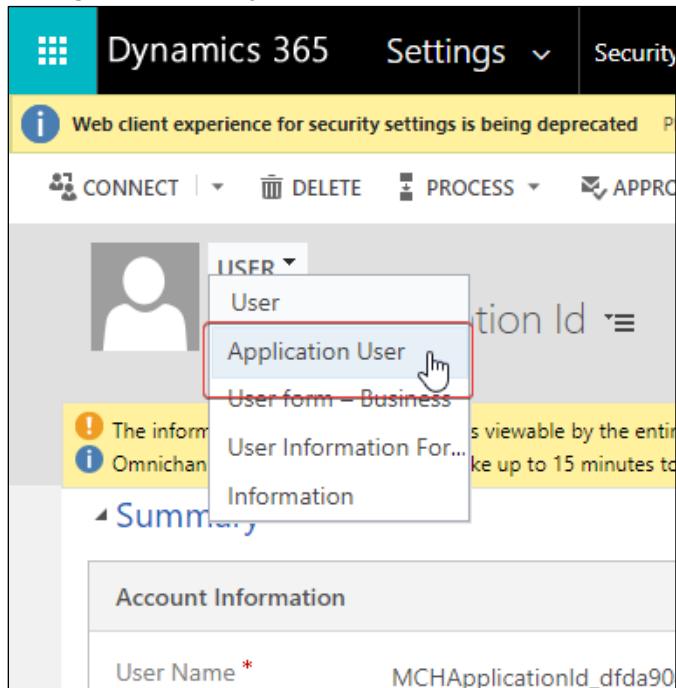
- @Me
- Access Mode Interactive Users
- Administrative Access Users
- Administrators
- Agents
- All
- Application Users
- Associated Record Team Members
- Bot agents
- Bot Users
- By Me
- Disabled Users
- Disabled users consuming licenses
- Enabled Users**
- Full Access Users

11. While in the Enabled User list, scroll to **find your App user** or use the **Search** bar. Double click on the user or select the row and click Edit.

*Note: If you are in an official training, search for the Application User – MCH Application ID*

Full Name	Position	Main Phone	Business Unit	Site	Title	Primary Email	
# MCH Application Id			unqd8e60411e999ec118			MCHApplicationId_dfd9044-c...	

12. Change the **form type** from User to **Application User** above the User Name.



13. You will see a new form appear that aligns to an Application User.

The screenshot shows a new form titled 'USER : APPLICATION USER'. At the top, it says '# MCH Application Id'. Below that, a yellow banner states 'The information provided in this form is viewable by the entire organization.' The form has two main sections: 'Summary' and 'User Information'. The 'Summary' section contains fields for 'User Name' (locked), 'Application ID' (locked), 'Application ID URI' (locked), and 'Azure AD Object ID' (locked). The 'User Information' section contains fields for 'Full Name' (locked) and 'Primary Email' (locked). Both 'Full Name' and 'Primary Email' fields show the value '# MCH Application Id'. The 'User type' field is set to 'Application user'.

14. In the **User Information** section, enter or select the following information and click the **Save** icon in the bottom right corner:
- User type:** Select **Bot application user**. This will *display a new field* to store the Bot application Id.
  - Bot application ID:** This is the Azure Health BotId you copied when enabling the Teams channel. This field is displayed once the User Type is selected to be Bot application user.

The screenshot shows the 'User Information' form. It includes fields for 'Full Name', 'Primary Email', 'User type' (with a dropdown menu open showing 'Application user' and 'Bot application user'), and 'Bot application ID'. The 'User type' dropdown has a red border around it, and the 'Bot application user' option is highlighted with a mouse cursor.

15. Select **Manage Roles** on the command bar.

The screenshot shows a user profile page with a navigation bar at the top containing 'CONNECT', 'DELETE', 'PROCESS', 'APPROVE EMAIL', 'REJECT EMAIL', 'REASSIGN RECORDS', 'MANAGE ROLES', and 'JOIN TEAMS'. The 'MANAGE ROLES' button is highlighted with a red border. Below the navigation bar, there is a user icon and the text 'USER : APPLICATION USER'. A note below says '# MCH Application Id'. A yellow banner at the bottom left of the page states: 'The information provided in this form is viewable by the entire organization.' The 'Summary' section is expanded, showing 'Account Information' with a 'User Name' field containing a long GUID.

16. Assign the **Omnichannel Agent role** to the Bot User as you did for your own user in the previous task. This will allow the bot to act as an omnichannel agent like your user.

The screenshot shows the 'Manage User Roles' dialog box. It asks 'What roles would you like to apply to the 1 User you have selected?'. A table lists available roles under 'Role Name' and 'Business Unit'. The 'Omnichannel agent' role is checked and highlighted with a red border. At the bottom are 'OK' and 'Cancel' buttons.

Role Name	Business Unit
<input type="checkbox"/> Marketing, Business App Access	healthcareinaday
<input type="checkbox"/> Office Collaborator	healthcareinaday
<input type="checkbox"/> Omnichannel administrator	healthcareinaday
<input checked="" type="checkbox"/> Omnichannel agent	healthcareinaday
<input type="checkbox"/> Omnichannel supervisor	healthcareinaday
<input type="checkbox"/> Orchestrator	healthcareinaday

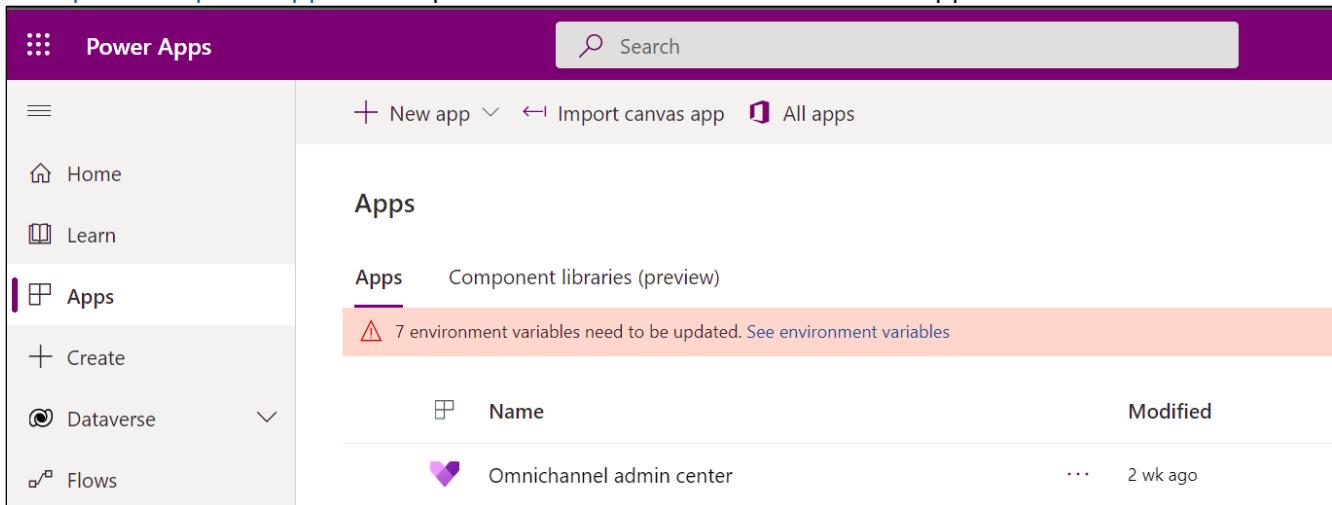
**Congratulations!** You successfully created a Bot User and assigned to it the Omnichannel Agent role.

## Task 3: Create and Configure Human Agent Queue

Queues are used to collect and distribute workload among agents. Agents are added as members to the queues and the workload is distributed among the agents based on assignment methods. Learn more at [Manage queues for unified routing](#).

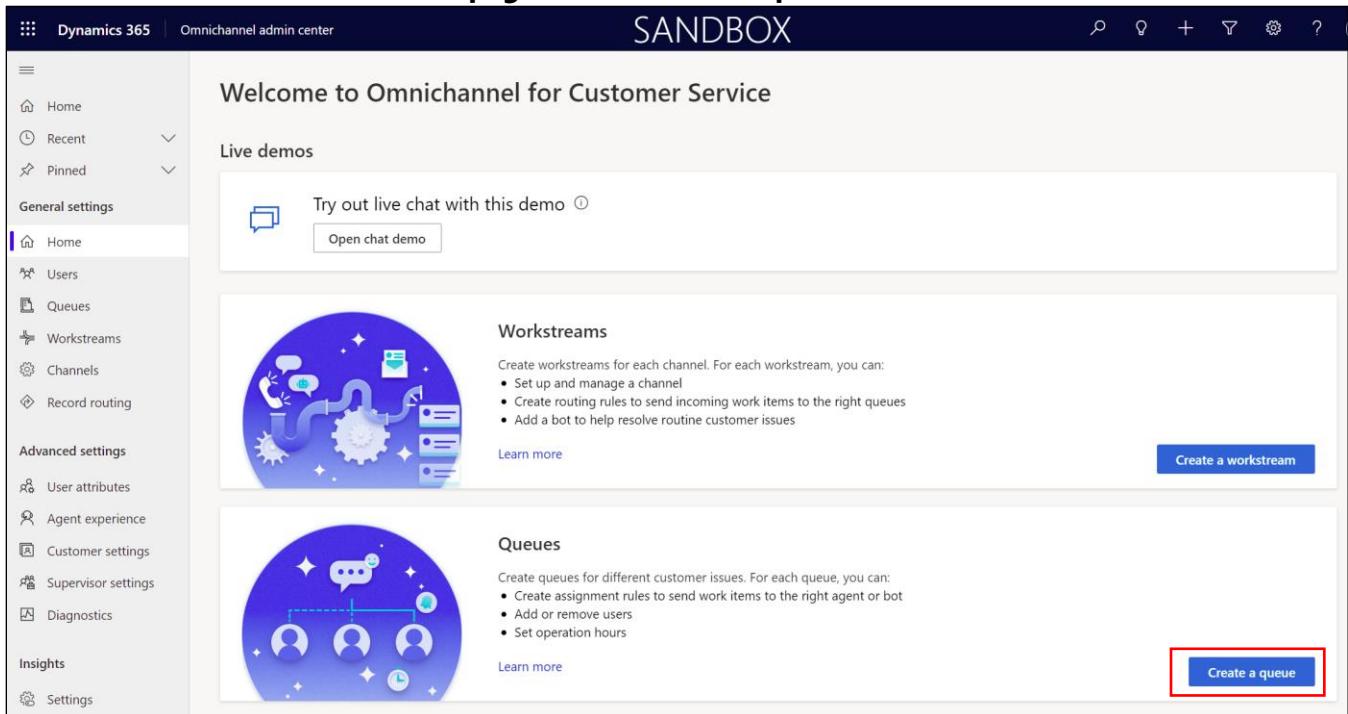
In this task, you will create the omnichannel queue necessary to communicate with a human agent.

1. In <http://make.powerapps.com>, open the **Omnichannel admin center** app.



The screenshot shows the Power Apps portal interface. On the left, there's a navigation bar with 'Power Apps' at the top, followed by 'Home', 'Learn', 'Apps' (which is selected), 'Create', 'Dataverse', and 'Flows'. The main area is titled 'Apps' and shows a list with one item: 'Omnichannel admin center'. The list has columns for 'Name' and 'Modified'. A red box highlights the 'Omnichannel admin center' entry.

2. You should be landed on the **Homepage**. Select **Create a queue** from the center shortcut.



The screenshot shows the 'Welcome to Omnichannel for Customer Service' page. On the left, there's a sidebar with 'General settings' (selected) and 'Advanced settings' sections. The main area has two sections: 'Workstreams' and 'Queues'. The 'Queues' section contains a 'Create a queue' button, which is highlighted with a red box. Other buttons in the 'Queues' section include 'Create workstream' and 'Learn more'.

3. You will now create a queue called "**Escalate to Human**" which will manage and redirect the incoming messages from a user to a Customer Service (human) Agent when Bot sends the user through to a live agent. Create the new Queue with the following details:
  - a. **Name:** Escalate to Human
  - b. **Type:** Messaging
  - c. **Group number:** [any number]

d. Click **Create**.

Create a queue

Queues are used to collect and distribute work among agents. Within queues, you can add users, set an assignment method, and add operation hours. [Learn more](#)

Name \*

Type \*

Group number \* ⓘ

Owner

**Create**

4. The new Queue record will open and contain a “Users” **subgrid**. Select **Add users** in the subgrid.

Dynamics 365 | Omnichannel admin center SANDBOX

Escalate to Human Edit

Users X Required

Add users to this queue

Work items in this queue will be assigned to these agents based on your work assignment and work distribution settings. [Learn more](#)

**+ Add users**

Assignment method [Learn more](#)

**Highest capacity** Read-only

Work items will be prioritized in the order they enter the queue. Among the agents who match skills, presence, and capacity, work will be assigned to the agent with the most capacity.

Operation hours Optional

**Set your operation hours**

Choose the days and hours for this queue to be active—agents will only receive work during these hours. If no operation hours are set, this queue will be active 24/7.

**+ Set operation hours**

General settings

- Home
- Recent
- Pinned

Advanced settings

- User attributes
- Agent experience
- Customer settings
- Supervisor settings
- Diagnostics

Insights

Settings

5. Search for your user and add it to the queue.

**Add Users**

Choose which agents can be assigned work items from this queue. [Learn more](#)

Name ↑	Role	Capacity profile	Capacity	Business unit
Omnichannel Power Virtual Ag...	Agent		2000000000	cloud4healthcare
<input checked="" type="checkbox"/> Allen Contoso	Agent			cloud4healthcare
# MCH Application Id	Agent		100	cloud4healthcare

6. The user is now added to the queue with the **agent** role.

**Escalate to Human** [Edit](#)

Messaging					
<a href="#">Type</a> Allen Contoso Owner					
Users					
Name ↑	Role	Capacity profile	Capacity	Business unit	Date added
Allen Contoso	Agent			cloud4healthcaredemo	04/10/2022

1 of 1

7. Select **Queues** on the left navigation bar and you will now see it listed in the queues subgrid.

**Queues**

Name	Group number ↓	Type	Owner
Escalate to Human	1	Messaging	Allen Contoso
Default messaging queue	2,147,483,647	Messaging	Allen Contoso
Default entity queue	2,147,483,647	Record	Allen Contoso

**Congratulations!** You have created the necessary queue to escalate to human agent and added your user to the messaging queue. Now we can create the workstream to initially route to a virtual bot along with routing rules to direct the user to Escalate to Human queue in the proper conditions.

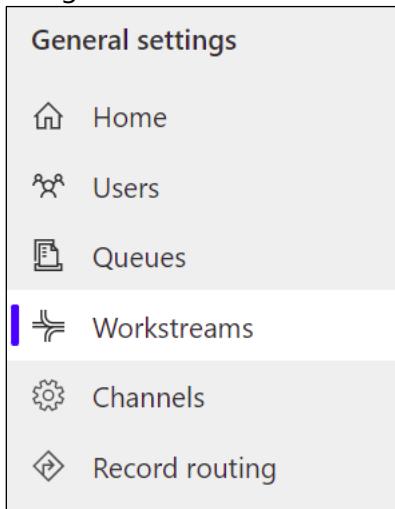
## Task 4: Create Workstream with Context Variables and Routing Rules

Workstreams are containers to enrich, route, and assign work items. A workstream is associated with a channel, such as live chat, voice, or case. After a bot is added to a workstream, the incoming work item is first routed to the selected bot at runtime based off classification rules. Learn more at [Create workstreams for unified routing](#).

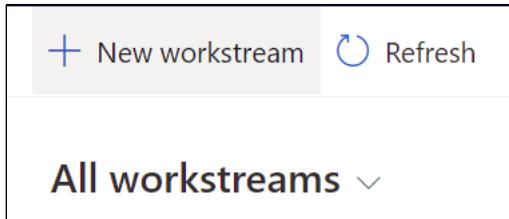
In this task, we will set up basic chat routing with a new workstream. This will allow for users to chat with a bot user initially and route to a live human agent in the proper situation. We will complete the following:

- **Create a new workstream**
- **Set up a live chat channel**
- **Add bot for initial routing:** Initial customer conversation is directed to the Azure Health Bot
- **Create context variable and routing rule to escalate to human agent:** When context variable **EscalateToAgent** is present and set to 1, we route to the "Escalate to Human" queue we previously set up with our user as an agent who can take over the conversation.

1. Navigate to **Workstreams** on the left navigation bar.



2. Select **+ New Workstream** on the command bar.



3. Enter the following details for the new workstream:

- a. **Name:** Chat Workstream
- b. **Type :** Messaging
- c. **Channel:** Chat
- d. **Work distribution mode:** Push
- e. Select **Create**

Create a workstream

A workstream is a collection of settings, including channel set up, routing rules, work distribution, and bots. Your workstream settings will be used to route customers to the right queues and agents. [Learn more](#)

**Name \***

**Owner**

**Type \***

**Channel \***

Make chats persistent  ⓘ

Signed-in customers and their agents can close and reopen their chat anytime and pick up the conversation where they left. We'll show a full conversation history every time the chat is reopened, creating one continuous experience. [Learn more](#)

**Work distribution mode**

Push

Incoming conversations will be assigned to agents automatically based on capacity and presence. You can also allow picking of open work items that go unassigned.

Pick

Incoming conversations will go to the open work items section of the agent dashboard. Agents will pick the conversations they work on.

**Create**

4. On the Chat Workstream record, you must set up your chat channel. Select **Set up chat** under Live chat.

**Chat Workstream** [Edit](#)

	Chat Channel	Messaging Type	Allen Contoso Owner										
<p><b>Live chat</b> <small>Required</small></p> <p><b>Set up your chat channel</b></p> <p>We'll guide you through each step of the process. When you're done, the settings you've turned on will show up here. <a href="#">Learn more</a></p> <p style="text-align: center;"><b>Set up chat</b></p>													
<p><b>Routing rules</b> <a href="#">Learn more</a></p> <p><small>Work classification (optional)</small> Add detailed information to incoming work items with classification rules. This allows for more precise routing and assignment. <a href="#">Learn more</a> <span style="float: right;"><b>+ Create ruleset</b></span></p> <p><small>Route to queues</small> Create rules to send incoming work items to the right queues. If no rules are created or no rules match, incoming work will be sent to the default queue. <a href="#">Learn more</a> <span style="float: right;"><b>+ Create ruleset</b></span></p>													
<p><b>Work distribution</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;"><small>Work distribution mode</small> <b>Push</b></td> <td style="width: 50%; text-align: right;"><small>See more</small></td> </tr> <tr> <td>Capacity</td> <td>Unit based 30</td> </tr> <tr> <td>Block capacity for wrap up</td> <td>Always block</td> </tr> <tr> <td>Allowed presences</td> <td>Available Busy</td> </tr> <tr> <td>Default skill matching</td> <td>None</td> </tr> </table>		<small>Work distribution mode</small> <b>Push</b>	<small>See more</small>	Capacity	Unit based 30	Block capacity for wrap up	Always block	Allowed presences	Available Busy	Default skill matching	None	<p><b>Bot</b> <small>Optional</small></p> <p><b>Add a bot</b></p> <p>All incoming work will be routed to the bot first. If needed, your bot will transfer customers to the right queues to speak with human agents.</p> <p><a href="#">Learn more</a> <span style="float: right;"><b>+ Add bot</b></span></p>	
<small>Work distribution mode</small> <b>Push</b>	<small>See more</small>												
Capacity	Unit based 30												
Block capacity for wrap up	Always block												
Allowed presences	Available Busy												
Default skill matching	None												

5. **Live Chat setup** screen will open. Enter the channel details as follows:

- Name:** Chat Widget
- Language:** English – United States
- Select **Next.**

Live chat setup

- Channel details
- Chat widget
- Behaviors
- User features
- Review and finish
- Success

### Channel details

Name \*

Language \*

Type \*

Channel \*

Back Next

6. On the following screen, toggle to enable **Proactive chat**. Here you may define any additional settings for the chat widget.

Live chat setup

- Channel details
- Chat widget
- Behaviors
- User features
- Review and finish
- Success

### Chat widget

This is the widget your customers will open to start a chat. Customize your widget below. [Learn more](#)

Title \*

Subtitle

Theme color \*

Logo URL \*

Agent display name

Widget position \*

**Proactive chat** On

Reconnect to previous chat

Show widget during operation hours

Back Next Cancel

7. Click **Next** to see the **Behavior** settings you can customize for your bot, including automated messages and surveys. No need to customize anything here now.

**Behaviors**

Select the features you'd like to auto-perform when a customer starts a chat.

**Custom automated messages**

Customize an automated message from a template for this workstream channel. Default automated message templates can be found in Global Settings, and are automatically applied to all workstreams. [Learn more](#)

Message trigger	Automated message	Status
You have no custom automated messages - select "add" to create one.		

**Pre-conversation survey**  Off

Create a survey for customers to fill out before chatting with a bot or agent. [Learn more](#)

**Post-conversation survey**  Off

Post conversation survey is currently supported through Dynamics 365 Customer Voice which is a comprehensive survey solution that builds on the current survey-creation experience of Microsoft Forms in Office 365.

ⓘ This will be operated by Dynamics 365 Customer Voice. [Learn more](#)

**Authentication settings**  Off

When this is on, customers will be required to sign in before they start a chat. Authentication settings can be

**Back** **Next** **Cancel**

8. Click **Next** to see the **User features** that can be defined for the bot. Nothing is needed here now.

**User features**

When turned on, these features can be used by agents and customer during a chat session.

**File attachments**  Off

Let customers and agents include attachments with their messages.

**Customer notifications**  Off

Let customers get notifications for incoming messages.

**Conversation transcripts**  Off

Let customers download or email their conversation transcripts. [Learn more](#)

By enabling this feature, you consent to your data being shared with third-party systems and flowing outside of your organization's compliance and geo boundaries (including Government Cloud environments). Learn more in the [Microsoft Privacy Statement](#).

**Voice and video calls**  Off

Let agents switch to voice or video calling during a chat session. [Learn more](#)

For Government Cloud environments, by enabling this feature, you consent to your data flowing outside of your organization's compliance and geo boundaries. Learn more in the [Microsoft Privacy Statement](#).

**Screen sharing**  Off

Let agents see and interact with the customer's screen. [Learn more](#)

By enabling this feature, you consent to your data being shared with third-party systems and flowing outside of your organization's compliance and geo boundaries (including Government Cloud environments). Learn more in the [Microsoft Privacy Statement](#).

**Back** **Next** **Cancel**

9. Review your settings and select **Create channel**.

The screenshot shows the 'Live chat setup' wizard. On the left, a vertical navigation bar lists steps: Channel details (checked), Chat widget (checked), Behaviors (checked), User features (checked), Review and finish (unchecked), and Success (unchecked). The main area is titled 'Review and finish'. It displays 'Channel Details' with Name: Chat Widget and Language: English - United States. Below that is 'Chat Widget' with Chat title: Let's chat, Chat subtitle: We're Online, Chat color: Blue, Logo URL: https://oc-cdn-ocprod.azureedge.net/livechatwidget/images/chaticon.svg, and Proactive chat: Enabled. At the bottom are 'Back' and 'Create channel' buttons.

10. Once the chat channel is successfully created, **copy the script** of the chat widget, and save it somewhere to add it to your website later. Select **Done** to close the wizard.

The screenshot shows the 'Live chat setup' wizard. The navigation bar on the left has all steps checked: Channel details, Chat widget, Behaviors, User features, Review and finish, and Success. A message box at the top right says 'Chat Channel Created' with the subtext 'We have successfully created your chat channel'. Below it, 'What's Next?' says 'Now that your channel has been setup, you can now look at the following next steps'. Under 'Add Widget To Your Website', it says 'Copy and paste this snippet to add your chat widget to any webpage. You can get this snippet anytime from your chat channel settings.' A code snippet is shown in a box:

```
<script id="Microsoft_Omnichannel_LCWidget" src="https://oc-cdn-ocprod.azureedge.net/livechatwidget/scripts/LiveChatBootstrapper.js" data-app-id="73a43dc1-c5fa-4661-bdde-927caa0bef4" data-lcw-version="prod" data-org-id="bbb0ecf5-7280-4b01-abef-261b1f44b48b" data-org-url="https://unqbbb0ecf572804b01abef261b1f44b-crm.omnichannelengagementhub.com"></script>
```

A red box highlights the 'Copy' button at the bottom of the code box.

11. In your new **Chat Workstream** record, select **Add Bot** to add the Azure Health bot for initial routing.

Chat Workstream [Edit](#)

Chat Channel	Messaging Type	Owner
Chat widget	Allen Contoso	Owner

1. Chat Widget [▼](#)

[+ Add chat channel](#)

Language: English - United States

Chat widget [Edit](#) [Delete](#)

[Copy code snippet](#)

< 1 of 1 channels >

Routing rules [Learn more](#)

[Work classification \(optional\)](#) [+ Create ruleset](#)  
Add detailed information to incoming work items with classification rules. This allows for more precise routing and assignment. [Learn more](#)

[Route to queues](#) [+ Create ruleset](#)  
Create rules to send incoming work items to the right queues. If no rules are created or no rules match, incoming work will be sent to the default queue. [Learn more](#)

Work distribution [See more](#)

Work distribution mode: Push	Capacity: Unit based
Capacity: 30	Block capacity for wrap up: Always block
Allowed presences: Available	Default skill matching algorithm: Busy
None	

Bot [Optional](#)

Add a bot  
All incoming work will be routed to the bot first. If needed, your bot will transfer customers to the right queues to speak with human agents.

[Learn more](#) [+ Add bot](#)

[+ Add bot](#)

Show advanced settings

12. Find and **select** your bot. Click **Add**. *For official trainings, your bot should be called "MCH Application Id".*

Add from existing [X](#)

Smart assist bots [Refresh](#) [Filter by name](#)

Name	Id	Created by	Created on ↑
# MCH Application Id	0708c73c-ce11-45e4-b3c1-595f15c0d316	Allen Contoso	04/14/2022
Omnichannel Power Virtual Agents Omnidirectional PVA Application	87f51aaa-5e4e-4dcf-b4f6-1aab011faef	# PowerVirtualAgentsProd	04/13/2022

13. This should open the advanced settings and display your bot in the **smart assist bots** subgrid.

**Advanced settings**

**Sessions**

**Default** Chat session - default  
This determines which tabs to display when an agent begins a new session, and which agent scripts will be used.

**Agent notifications**

**Incoming unauthenticated** Chat - incoming unauthenticated - default  
**Incoming authenticated** Chat - incoming authenticated - default  
**Consult** Chat consult - default  
**Transfer** Chat - transfer - default  
**Supervisor assign** Chat - supervisor assign - default

**Context variables**

**Add a context variable**  
Add custom context that can be used to define routing rules. [Learn more](#)

**+ Add context variable**

**Smart assist bots**

Name	Id
# MCH Application Id	0708c73c-ce11-45e4-b3c1-595f15c0d316

1 of 1

**Quick replies**

**Add quick replies**  
Associate quick replies to allow agents to use in this workstream. If the quick replies are not associated with any workstream, they are available for all workstreams.  
[Learn more](#)

**+ Add quick replies**

14. You can also access the advanced settings at the bottom of the record page by selecting **Show advanced settings**.

**Chat Workstream** [Edit](#)

Channel	Type	Owner
Chat	Messaging	Allen Contoso

**1. Chat Widget**

**Language** English - United States

**Chat widget** [Copy code snippet](#) [Edit](#) [Delete](#)

< 1 of 1 channels >

**Routing rules** [Learn more](#)

**Work classification (optional)**  
Add detailed information to incoming work items with classification rules. This allows for more precise routing and assignment. [Learn more](#) [+ Create ruleset](#)

**Route to queues**  
Create rules to send incoming work items to the right queues. If no rules are created or no rules match, incoming work will be sent to the default queue. [Learn more](#) [+ Create ruleset](#)

**Work distribution**

Work distribution mode	Push
Capacity	Unit based 30
Block capacity for wrap up	Always block
Allowed presences	Available Busy
Default skill matching algorithm	None

**Bot** Optional

**Add a bot**  
All incoming work will be routed to the bot first. If needed, your bot will transfer customers to the right queues to speak with human agents.  
[Learn more](#) [+ Add bot](#)

**Show advanced settings**

15. Now we want to define a new context variable and routing rule. Select **+ Add Context variable**.

The screenshot shows the 'Advanced settings' section of a workstream configuration. It includes four main panels:

- Sessions**: A panel for managing sessions, showing a 'Default' entry for 'Chat session - default'.
- Agent notifications**: A panel for managing agent notifications, listing categories like 'Incoming unauthenticated', 'Incoming authenticated', 'Consult', 'Transfer', and 'Supervisor assign' each associated with a 'Chat' icon and a 'default' entry.
- Context variables**: A panel titled 'Add a context variable' with a sub-instruction 'Add custom context that can be used to define routing rules.' It features a blue button labeled '+ Add context variable' which is highlighted with a red box.
- Smart assist bots**: A panel showing a single entry '# MCH Application Id' with 'Name' and 'Id' columns, and the value '0708c73c-ce11-45e4-b3c1-595f15c0d316'.

16. In the context variable flyout, select **+ Add** to add new context variable.

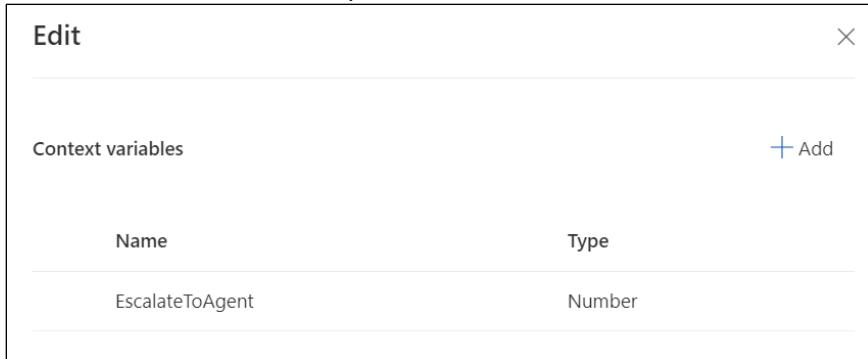
The 'Edit' flyout window has a header 'Edit' and a close button 'X'. It contains a section titled 'Context variables' with a blue '+ Add' button. Below this are two columns: 'Name' and 'Type'.

17. Create the new Context Variable with the following details:

- Name:** EscalateToAgent
- Type:** Number
- Click Create**

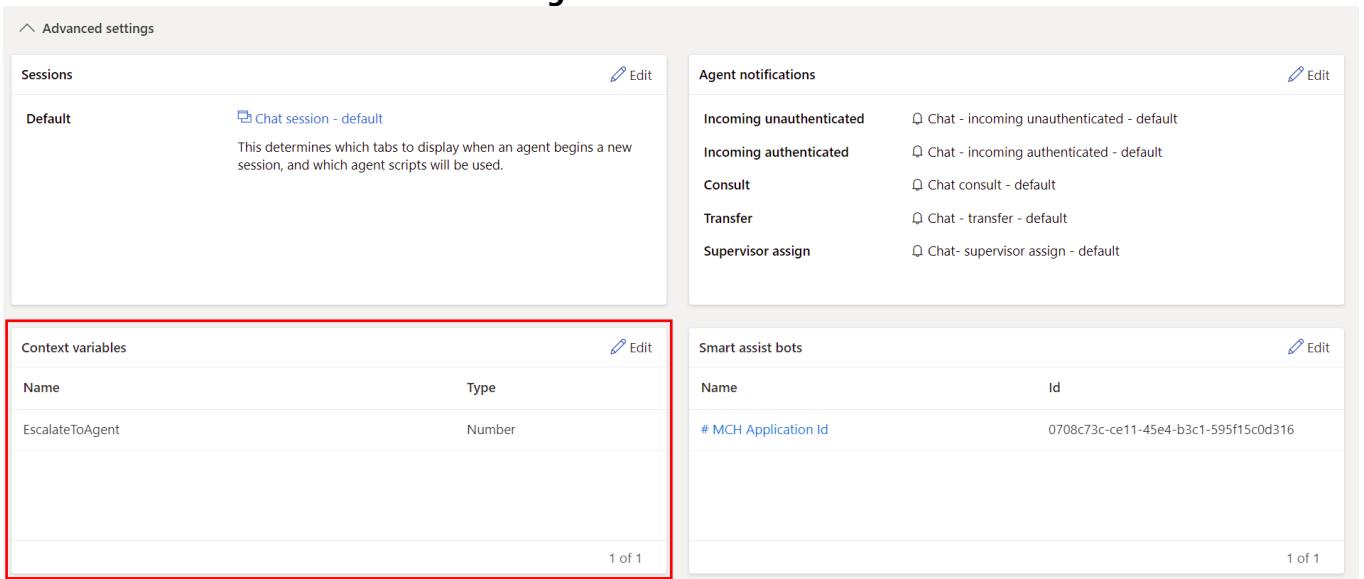
The dialog box is titled 'Add context variable' and has a close button 'X'. It contains two input fields: 'Name \*' with the value 'EscalateToAgent' and 'Type \*' with the value 'Number'. At the bottom are 'Create' and 'Cancel' buttons.

18. Close the context variable panel.



The screenshot shows a modal window titled 'Edit' with a header 'Context variables'. At the top right is a blue 'Add' button with a plus sign. Below it is a table with two columns: 'Name' and 'Type'. There is one row containing the entry 'EscalateToAgent' under 'Name' and 'Number' under 'Type'.

19. You should now see the new **EscalateToAgent** context variable in the live chat workstream.



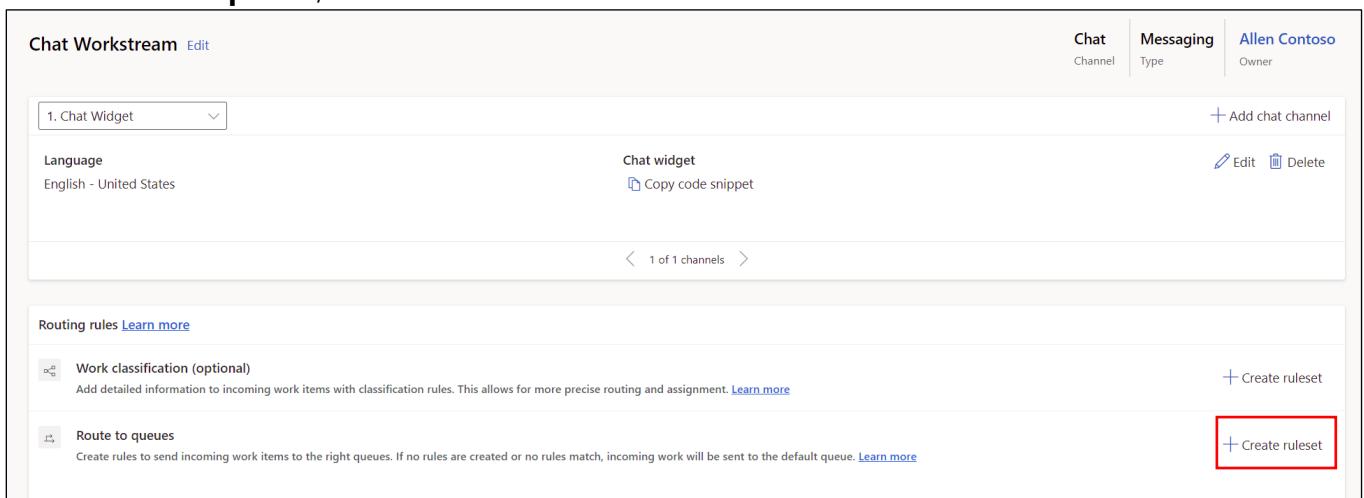
The screenshot shows the 'Advanced settings' page for a 'Chat Workstream'. It includes sections for 'Sessions' (with a 'Default' tab selected), 'Agent notifications', 'Context variables' (which contains the 'EscalateToAgent' entry), and 'Smart assist bots'. A red box highlights the 'Context variables' section.

20. Select Advanced Settings to collapse to the main page.



A screenshot of a button labeled 'Advanced settings' with a downward arrow icon, indicating it has been collapsed.

21. Under **Route to queues**, select **+Create ruleset**.



The screenshot shows the 'Chat Workstream' configuration page. It includes sections for 'Chat Workstream' settings, 'Routing rules', 'Work classification (optional)', and 'Route to queues'. In the 'Route to queues' section, there is a blue '+ Create ruleset' button, which is highlighted with a red box.

22. Create the new route-to-queues ruleset with the following details:

- Name:** Human Agent
- Description:** Escalate to human agent
- No Conditions.

**Create route-to-queues ruleset** ×

After you name this ruleset, you can start creating rules.

**Name \***  
Human Agent

**Description**  
Escalate to human agent

[Learn more](#) Create Cancel

23. In the new Human Agent queue ruleset, select **+ Create rule**.

Route to queues Allen Contoso  
Owner

[← Human Agent](#) [Edit](#) + Create rule

Decision list

+  
Create a route to queue rule + Create rule

24. Name the new rule **Human Agent Rule**.

**Create route to queue rule**  
Add conditions and select the queue to route to. [Learn more](#)

**Rule Name \***  
Human Agent Rule

**🔗 Root record: Conversation**  
Root record is the starting record for the conditions and the output below. Starting from the root record you can navigate to its related records and attributes.

25. Under conditions, choose “**Add related entity**” from the dropdown.

Conditions

And ▾

+ Add ▾

+ Add row

= Add group

Route to **Add related entity**

Select queue \*

This screenshot shows the 'Route to' configuration section. It includes a dropdown menu with options like 'Add', 'Add row', and 'Add group'. Below this is a red box highlighting the 'Add related entity' button. A dropdown menu for 'Select queue' is also visible.

26. In the first two drop downs, Choose **Context item value** and **Contains data**. In the inline condition choose **EscalateToAgent Equals 1**.

Conditions

And ▾

Context item value (Con... ▾ Contains data ▾ ...)

And ▾

EscalateToAgent Equals 1 ...

+ Add ▾

+ Add ▾

This screenshot shows a detailed view of the 'Conditions' section. It uses an 'And' logic structure. The first condition is 'Context item value (Contains data)'. The second condition is another 'And' logic structure containing 'EscalateToAgent Equals 1'. There are also '+ Add' buttons for further conditions.

27. In the Route to queues section, choose **Escalate to Human** queue created previously.

Route to queues

Escalate to Human \*

This screenshot shows the 'Route to queues' configuration section. It has a dropdown menu with 'Escalate to Human' selected, indicated by a red box.

28. The configured rule set is shown below. Select **Create**.

**Create route to queue rule**

Add conditions and select the queue to route to. [Learn more](#)

Rule Name \*

**Root record:** Conversation

Root record is the starting record for the conditions and the output below. Starting from the root record you can navigate to its related records and attributes.

**Conditions**

**Route to queues**

**Create** **Cancel**

29. The Chat Workstream now has a Human Agent ruleset that will escalate to a human agent when the EscalateToAgent context variable is set to 1.

Route to queues				Allen Contoso Owner
<a href="#">← Human Agent</a> <a href="#">Edit</a>			<a href="#">+ Create rule</a> <a href="#">Refresh</a> <a href="#">Search</a>	
Before sending a work item to a queue, we'll match rules and operating hours in priority order. If none of the queues are currently in operation, the work item will be sent to the earliest operating queue. <a href="#">Learn more</a>				
Order	Rule name	Condition	Queue	
1	Human Agent Rule	EscalateToAgent equals 1	Escalate to Human	

**Congratulations!** You have created a new Workstream with the proper live chat channel, smart assist bot, and routing rule that will allow customers to begin conversation with a health bot and escalate to a human agent.

# Exercise 3: Embed Health Bot in Power Apps Portal

In this exercise, you will be embedding the **Omnichannel Chat Widget** into the Power Apps Customer self-service portal using Portal Management configuration. In your environment, we created a Lamna Healthcare Company Portal using the **Customer self-service portal** template before deploying Microsoft Cloud for Healthcare. Now we will configure the chat widget to show on the customer website.

**Customer self-service portal:** A customer self-service portal enables customers to access self-service knowledge, support resources, view the progress of their cases, and provide feedback.

**Portal Management:** Application to help you get started with the advanced portal configuration. In this walk-through, you will learn how to configure Chat widget in **Portal Management** app.

1. In <http://make.powerapps.com>, open the **Portal Management** app.

The screenshot shows the 'Apps' section of the Portal Management app. At the top, there are two tabs: 'Apps' (which is selected) and 'Component libraries (preview)'. Below the tabs, a message says '⚠️ 10 environment variables need to be updated. See environment variables'. There is a table with two rows. The first row has a 'Name' column with 'Lamna Healthcare Patient Portal'. The second row has a 'Name' column with 'Portal Management', which is highlighted with a grey background.

Name
Lamna Healthcare Patient Portal
Portal Management

2. Select **Content Snippets** in the left navigation pane

The screenshot shows the left navigation pane of the Portal Management app. The 'Content' section is expanded, and 'Content Snippets' is selected, indicated by a blue vertical bar on the left. Other options in the 'Content' section include 'Basic Forms'. Higher-level categories like 'Website' and 'Settings' are also visible.

- Home
- Recent
- Pinned
- Website
- Websites
- Page Templates
- Redirects
- Site Markers
- Site Settings
- Website Bindings
- Settings
- Content
- Content Snippets
- Basic Forms

1. In **Active Content Snippets**, type “**Chat**” in the **Search** box and press enter.

The screenshot shows a search results page for "Active Content Snippets". A search bar at the top contains the word "chat". Below the search bar, there are five columns: Name, Website, Content Snippet Language, Type, and Value. Two records are listed:

Name	Website	Content Snippet Language	Type	Value
Chat Widget Code	Customer Self-Service	---	HTML	---
Chat Widget Code	Healthcare Patient Portal	---	HTML	<script id="Microsoft_Omnichannel_LCWidget" src...

2. You will see two **Chat Widget Code** records retrieved in the list.

Click to open the Chat Widget Code record related to **Customer Self-service**.

The screenshot shows the "Active Content Snippets" list. The second record, "Chat Widget Code" under "Customer Self-Service", is highlighted with a blue background. The other record, "Chat Widget Code" under "Healthcare Patient Portal", is shown below it.

3. In the **Chat Widget Code** record associated with Customer self-service, select **Value (HTML) > HTML** Tab and then paste the **Chat Widget Code snippet** that you copied and stored in the previous task.

The screenshot shows the "Chat Widget Code" record details. The "General" tab is selected. The "Value (HTML)" tab is also visible. The "Value (HTML)" tab has a "Designer" and "HTML" option, with "HTML" selected. The code area contains the copied snippet:

```
1 rsion="prod" data-org-id="b7eccfa4-b990-4c1e-9486-5d5ec922bbf4" data-org-url="https://unqb7eccfa4b9904c1e94865d5ec922b-crm.omnichannelengagementhub.com"></script>
```

4. Click **Save & Close**.

The screenshot shows a modal dialog box with four buttons: a left arrow, a document icon, a "Save" button, and a "Save & Close" button. The "Save & Close" button is highlighted. The title of the dialog is "Chat Widget Code" and the subtitle is "Content Snippet".

5. Now open the other **Chat Widget Code** associated with the **Healthcare Patient Portal** website.

Active Content Snippets	
	Website
Chat Widget Code	Customer Self-Service
<b>Chat Widget Code</b>	Healthcare Patient Portal

6. In the **Chat Widget Code** record associated with the Healthcare Patient Portal, paste in **Value (HTML)** the same **Chat Widget Code snippet** that you copied and stored previously and added to the customer self-service chat widget code. Replace any value that may have already populated the field.

**Chat Widget Code**  
Content Snippet

General Administration Related

Name	* Chat Widget Code
Website	* <a href="#">Healthcare Patient Portal</a>
Display Name	Chat Widget Code
Type	HTML
Content Snippet Language	---

Value (HTML)  
Designer | **HTML**

```
1 rsion="prod" data-org-id="b7eccfa4-b990-4c1e-9486-5d5ec922bbf4" data-org-url="https://unqb7eccfa4b9904c1e94865d5ec922b-crm.omnichannelengagementhub.com"></script>
```

7. Select **Save and Close**.

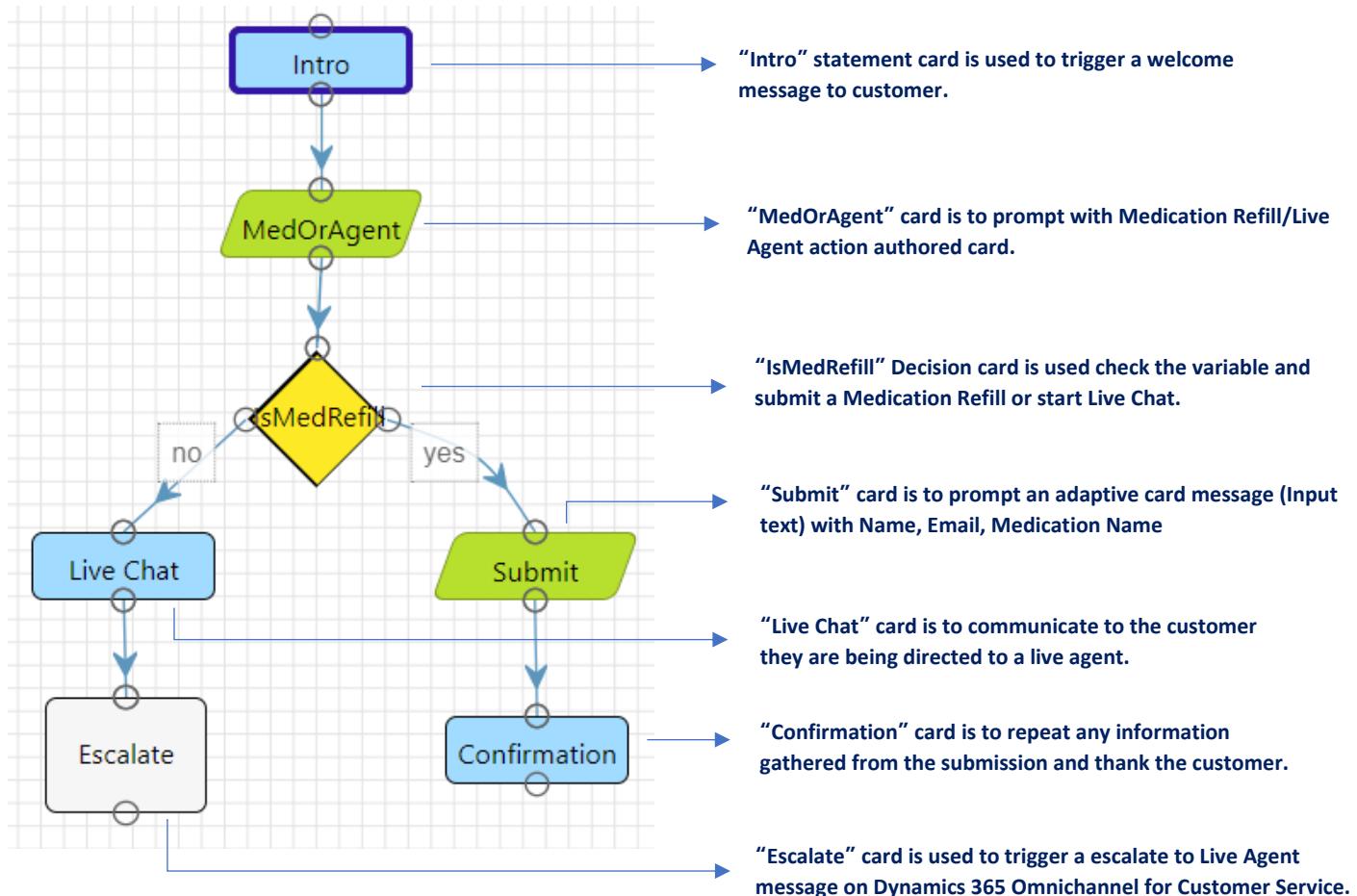
**Congratulations!** In this exercise you have successfully updated the chat widget in the Power App Portal Content Snippets. With this configuration, the Health Bot will be visible on the Power Apps portal for both the customer self-service template and the healthcare patient portal template.

# Exercise 4: Extend Azure Health Bot with Custom Scenarios

**Dynamics 365 Omnichannel** integration allows the patient to interact with **Azure Health Bot** using the Dynamics 365 chat widget to access the medical knowledge and your custom scenarios. It also, allows the escalation of a bot conversation to a live agent to continue the interaction. When escalating a conversation, Dynamics passes along the conversation history and the context to the agent.

In this exercise, you will be doing the following:

1. Designing the below Health Bot Scenario called **MCH\_PatientService**

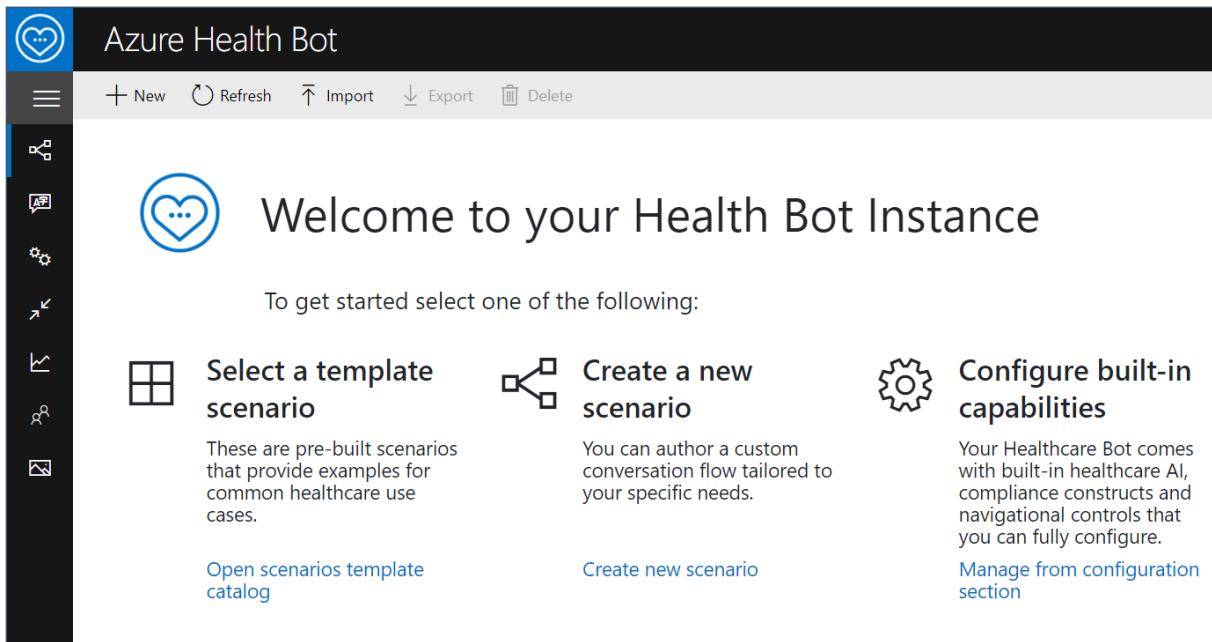


2. Design another Health Bot Scenario called **MCH\_PatientServiceWelcome**. This scenario holds the starting statement which will allow the user to invoke the **MCH\_PatientService** scenario.
3. Set the **Automatic Welcome Scenario** to be the **MCH\_PatientServiceWelcome** custom scenario you create. This will begin the scenario when a user first interacts with the Health Bot.

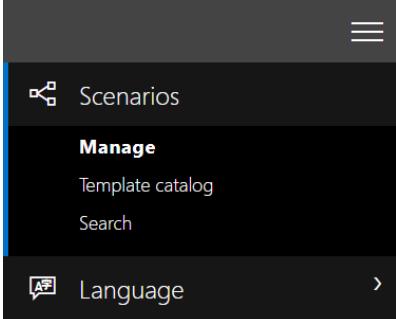
## Task 1: Create MCH\_PatientService Scenario

In this task, you will create the **MCH\_PatientService** bot scenario using the designer canvas.

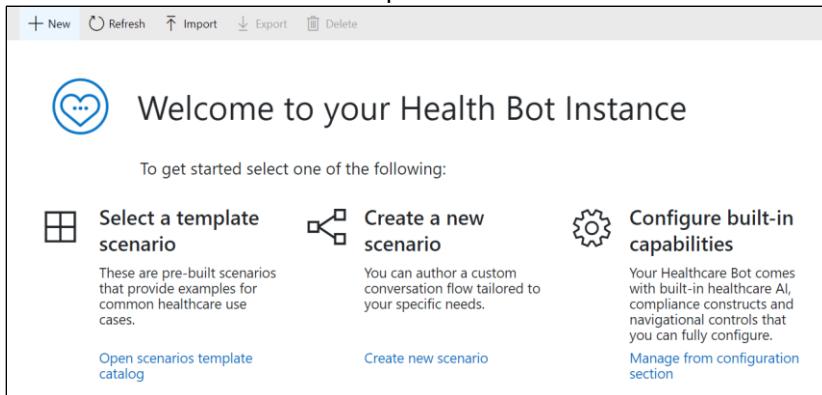
Navigate back to the Azure Health Bot instance homepage where you set the bot settings. This is the portal management link you copied from the Azure portal.



1. Click to Expand the Side navigation bar. Navigate to **Scenario > Manage**.



2. Click **+ New** button on the top ribbon.



3. Provide the following details for the new health bot scenario:

- Name:** MCH\_PatientService
- Scenario ID:** MCH\_PatientService

**New Scenario**

**Name\*** ⓘ  
MCH\_PatientService

**Description** ⓘ

**Scenario ID\*** ⓘ  
MCH\_PatientService

**Returning Message** ⓘ

**Interrupting scenario** ⓘ

**Breaking scenario** ⓘ

**Create** **Cancel**

4. Now let's design the scenario conversation. It should navigate you directly to the designer. If not, select the MCH\_PatientService scenario in **Scenarios > Manage** to edit.

The screenshot shows the Azure Health Bot Designer interface. The top bar includes the Azure Health Bot logo, user name (iaduser99-healthbot), and various icons. The main area is titled "MCH\_PatientService". The left side features a toolbar with icons for Save, Run, Set run arguments, Create Snapshot, and Exit, followed by a grid of step icons: Prompt, Yes/No, Statement, Branch, Switch, Begin, Replace, End, Data Conn., Skill, Global, Assign, Action, Wait, and LU. A search bar is positioned above the grid. The right side shows a "Web Chat v4" window with an "Info" tab active. The "Info" tab displays the following details for the scenario:

- Name: MCH\_PatientService
- Scenario ID: MCH\_PatientService
- Description:
- Returning Message:
- Interrupting: Disabled
- Breaking: Disabled

At the bottom, there are tabs for "Code" and "Designer" (which is selected), and a status bar showing the date and time (2021-09-16T09:28:44Z).

## Step 1: Add Bot Introduction Statement

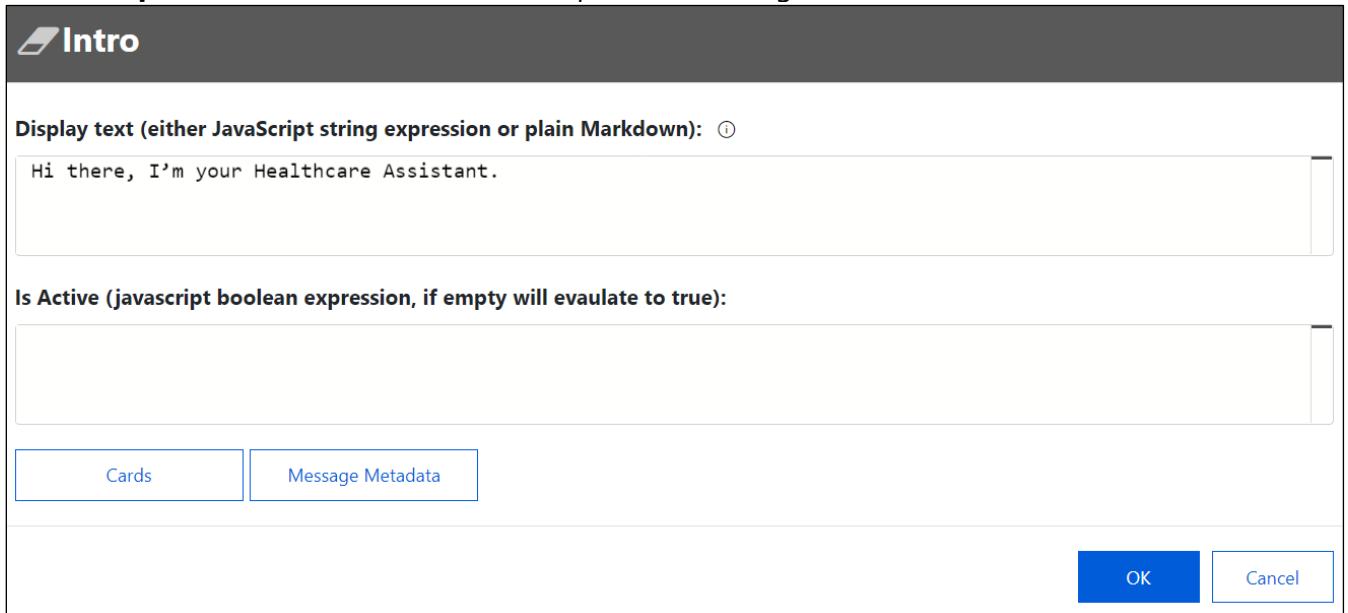
1. Add a beginning **Statement** to the scenario by **selecting** the icon **and dragging** Statement icon onto the editor.



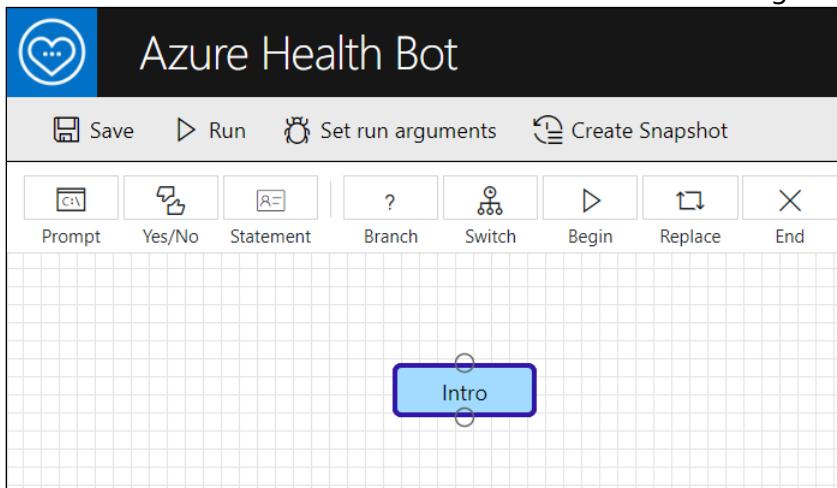
2. Enter the **Display Text**:

Hi there, I'm your Healthcare Assistant.

3. Select the **pencil** next to Statement in the top bar and Change Title to "**Intro**". Click **OK**.



4. Click **OK**. You will see the intro statement added to the designer canvas. Double click anytime to edit.



## Step 2: Add Statement for Medication Request or Live Agent

This section prompts two buttons Medication Refill and Live Agent. When user click any one of the buttons it will set the appropriate text to the variable MedicationOrAgent.

1. Select **Prompt** icon and drag down onto canvas



2. Enter the following details:

- a. **Display Text:** Would you like to request a medication refill or chat with a live agent?
- b. **Variable name:** MedicationOrAgent
- c. **Data type:** string
- d. Rename title to **MedOrAgent**.
- e. Click **Cards** button.

The dialog has a title bar 'MedOrAgent'. The 'Display text' field contains 'Would you like to request a medication refill or chat with a live agent?'. The 'Variable name' field is 'MedicationOrAgent'. The 'Variable Data Type' dropdown is set to 'string'. The 'Is Active' field is empty. The 'Suggestions' field is empty. At the bottom, there are 'Cards' and 'Message Metadata' buttons, and 'OK' and 'Cancel' buttons.

3. Select **Add Card**.

The dialog has a title bar 'Cards'. It contains an 'Add Card' button, a 'Layout' dropdown set to 'vertical', and 'OK' and 'Cancel' buttons at the bottom.

4. Select Card Type as **HeroCard**. Leave title blank as we already prompted with display text.
5. Click **Add Action** button twice to add two actions:
  - a. For the first action, select the following:
    - i. Action type: imBack
    - ii. Action value: MedicationRefill
    - iii. Action title: "Medication Refill"
  - b. For the second action, fill in the following:
    - i. Action type: imBack
    - ii. Action value: LiveAgent
    - iii. Action title: "Live Agent"

**Card**

**Card Type:**  
HeroCard

**Image Url:**  

**Title:**

**Sub Title:**

**Actions**

Action type	Action value	Action title	X
imBack	MedicationRefill	"Medication Refill"	
imBack	LiveAgent	"Live Agent"	

**Add Action**

**OK** **Cancel**

6. Click Ok three times to get back to designer

## Cards

HeroCard :: undefined :

[Add Card](#)

**Layout:**

vertical

[OK](#) [Cancel](#)

## MedOrAgent

**Display text (either JavaScript string expression or plain Markdown):** ⓘ  
Would you like to request a medication refill or chat [with](#) a live agent?

**Variable name (to store the input from the user):**  
MedicationOrAgent

**Variable Data Type:**  
string

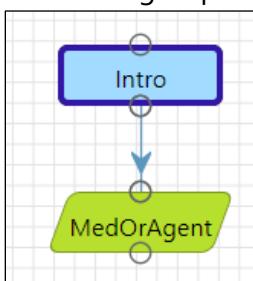
**Is Active (javascript boolean expression, if empty will evaluate to true):**

**Suggestions (javascript string array expression):**

[Edit Cards](#) [Message Metadata](#)

[OK](#) [Cancel](#)

7. Connect Intro and Appointment boxes. Select the bottom circle on intro and drag it to the top circle on the new prompt. An arrow will automatically appear when you try to connect Intro and MedOrAgent boxes using ellipse pointer.



8. Select **Save**.



9. Select **Run** to see the output in the WebChat on the right.

A screenshot of the Azure Health Bot interface. The "Run" button in the navigation bar is highlighted in blue. To the right, a "Web Chat v4" window is open, showing a conversation with "Lamina Healthcare Support". The messages are:

- Hi there, I'm your Healthcare Assistant.
- Would you like to request a medication refill or chat with a live agent?

Below the messages, there are two buttons:

- Medication Refill**
- Live Agent**

The timestamp "Just now" is shown at the bottom left of the chat window. At the bottom, there's a text input field with the placeholder "Type your message" and a microphone icon.

### Step 3: Add MedicationOrAgent Decision Branch

This section checks whether the user has clicked Medication Refill or Live Agent with the help of the variable MedicationOrAgent. It will redirect the message accordingly.

1. Add a **Branch** to the designer canvas.



2. Enter the following in the **javascript Boolean expression**:

scenario.MedicationOrAgent === "MedicationRefill"

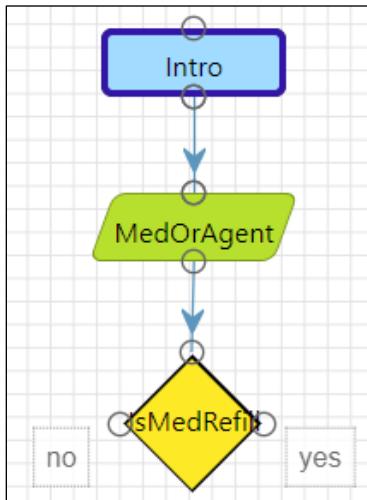
3. Rename to **IsMedRefill**. Select **OK**.

**IsMedRefill**

```
javascript boolean expression:
scenario.MedicationOrAgent === "MedicationRefill"
```

**OK**    **Cancel**

4. Select and drag the bottom circle of the **MedOrAgent** prompt to the top circle of the **IsMedRefill** branch decision to connect them.



#### Step 4: Prompt User to Enter Data for Medication Refill Option

1. Add a **Prompt** element. This will be used to display the Form data (using Adaptive Card) to capture Patient name, email, and phone to create an appointment.



2. Add the following details:
  - Variable name:** formData
  - Variable Data Type:** Object
  - Change Title to **Submit**
  - Do not add any display text since the adaptive card will display instead

## Submit

Display text (either JavaScript string expression or plain Markdown): [ⓘ](#)

Variable name (to store the input from the user):

Variable Data Type:

Is Active (javascript boolean expression, if empty will evaluate to true):

Maximum number of retries:

[Edit Cards](#)

[Message Metadata](#)

[OK](#)

[Cancel](#)

3. Click **Cards** button > **Add Card** > **Adaptive Card**.
4. Refer to the lab resources file **AdaptiveCardForMedicationRefill.txt** and copy the json content and paste it in the json section of your card.

# Card

Card Type:

AdaptiveCard

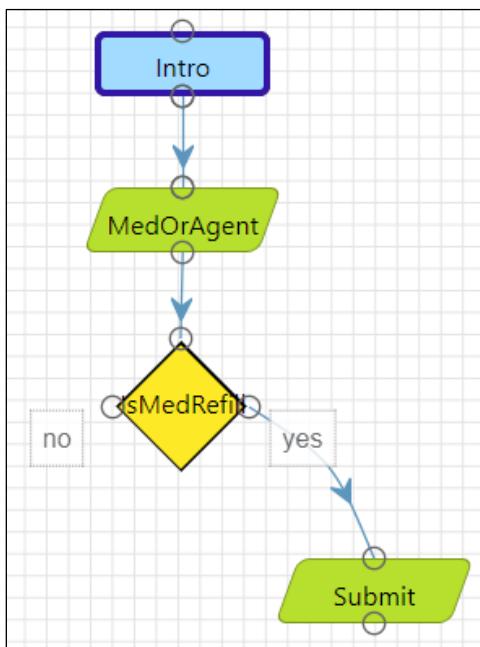
Design your own Adaptive Card

```
1 [
2   "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
3   "type": "AdaptiveCard",
4   "version": "1.0",
5   "body": [
6     {
7       "type": "ColumnSet",
8       "columns": [
9         {
10           "type": "Column",
11           "width": 2,
12           "items": [
13             {
14               "type": "TextBlock",
15               "text": "Tell us about yourself",
16               "weight": "bolder",
17               "size": "medium"
18             },
19             {
20               "type": "TextBlock",
21               "text": "We just need a few more details to get your Medication refill.",
22               "isSubtle": true,
23               "wrap": true
24             },
25             {
26               "type": "TextBlock",
```

OK

Cancel

5. Select **OK** three times to get back to the designer.
6. **Connect** the **Yes** condition of the **IsMedRefill** branch to the **Submit** prompt.



7. **Save** and **run** your current scenario. If you don't save the scenario first, it won't run with updates since the last save. If you haven't saved at all, it won't recognize any conversation.



8. You should see the below output when running the conversation and selecting "Medication Refill" card when prompted to show the AdaptiveCard.

A screenshot of a Web Chat v4 session. The top bar shows "Web Chat v4" and "en-us". The message list shows a message from "Bot" at "Just now" with the text "Tell us about yourself" and a message from the user at "Just now" with the text "We just need a few more details to get your Medication refill. Don't worry, we'll never share or sell your information." Below this, there are input fields for "Your name" (placeholder "Last, First"), "Your email" (placeholder "youremail@example.com"), and "Medication Requested". Each input field has a sub-field for "Medication Name". A large "Submit" button is centered below the "Medication Requested" field. At the bottom, there is a message input field with a paperclip icon and the placeholder "Type your message", and a microphone icon.

## Step 5: Add Confirmation Statement

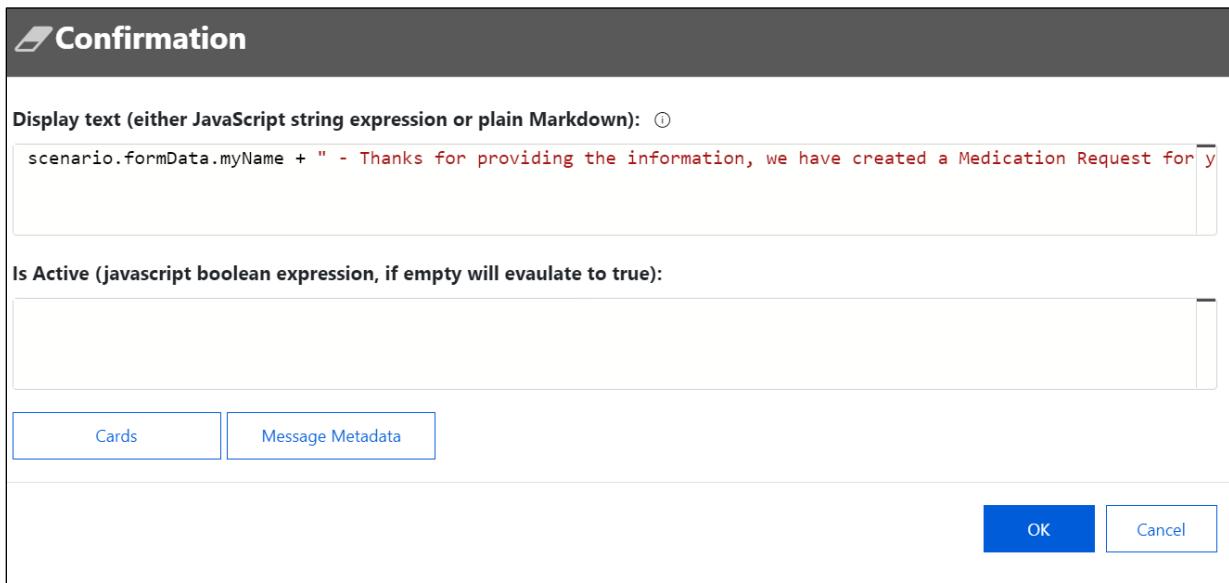
1. Add a **Statement** element.



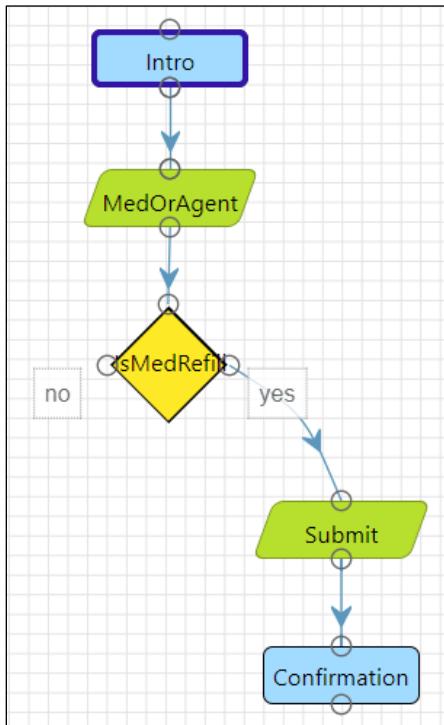
2. Add **Display text** as the following:

```
scenario.formData.myName + " - Thanks for providing the information, we have created a Medication Request for you regarding the following medication: " + scenario.formData.myMedReq
```

3. Rename the statement to **Confirmation**.



4. Connect the Submit step to the Confirmation step in the designer canvas.



5. Select **Save** and **Run** to see your scenario in the webchat.

Web Chat v4

en-us Just now

Bot

**Tell us about yourself**

We just need a few more details to get your Medication refill.

Don't worry, we'll never share or sell your information.

Your name

Jensen, Casey

Your email

caseyjensen@contoso.com

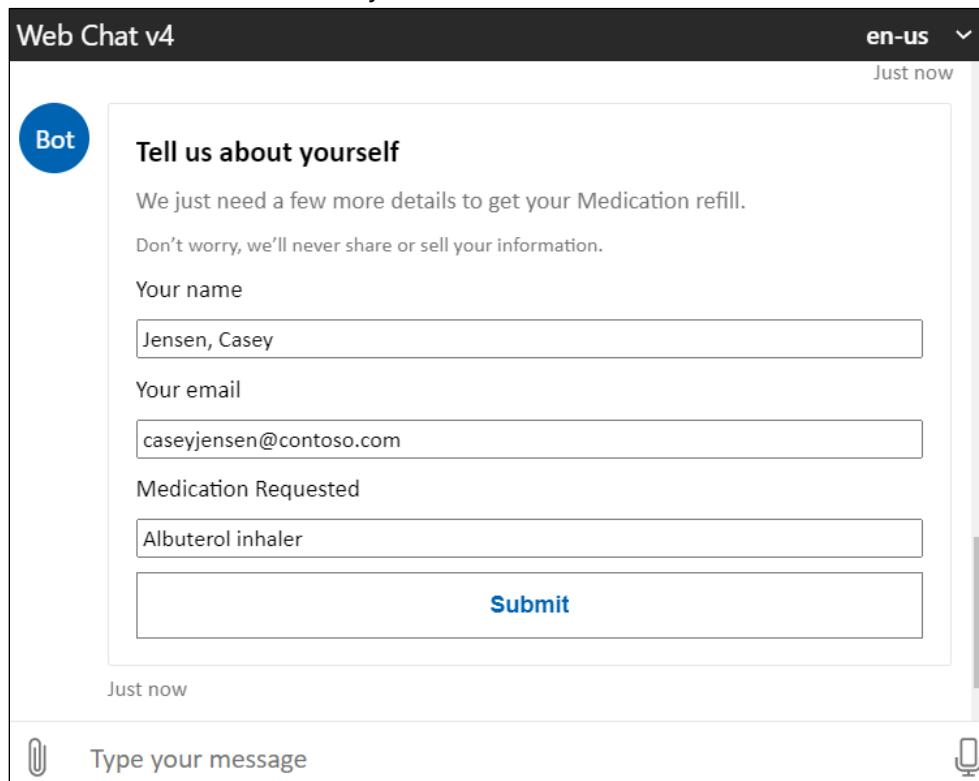
Medication Requested

Albuterol inhaler

**Submit**

Just now

Type your message



6. Fill in information for the request and click **Submit** to see the confirmation text.

Web Chat v4

en-us

Don't worry, we'll never share or sell your information.

Your name

Jensen, Casey

Your email

caseyjensen@contoso.com

Medication Requested

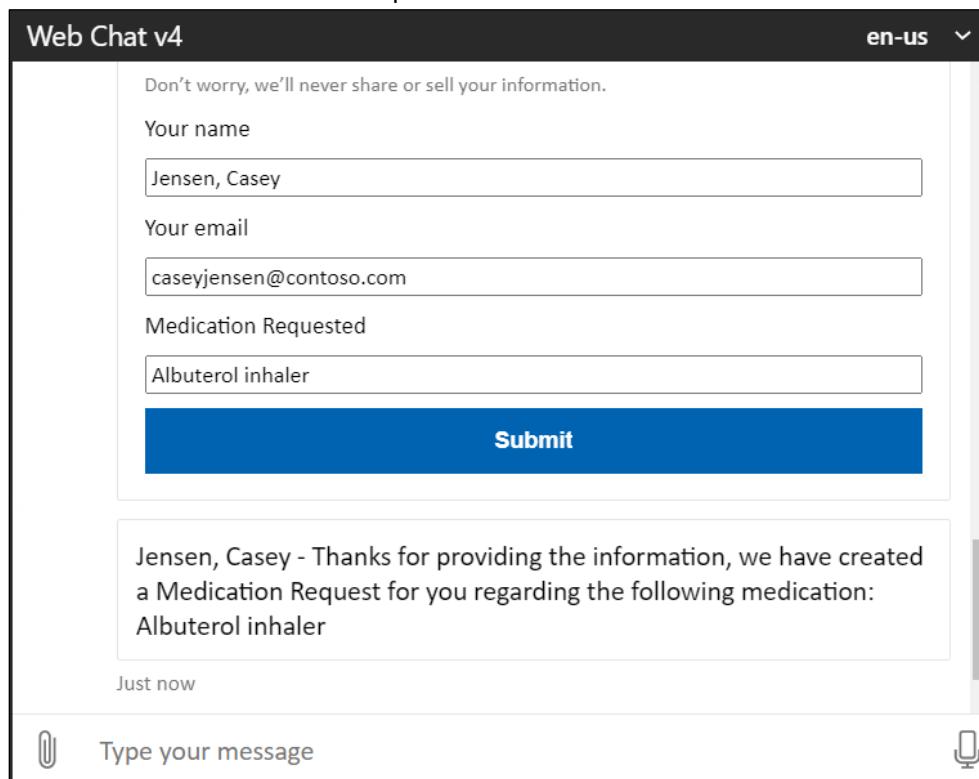
Albuterol inhaler

**Submit**

Jensen, Casey - Thanks for providing the information, we have created a Medication Request for you regarding the following medication:  
Albuterol inhaler

Just now

Type your message

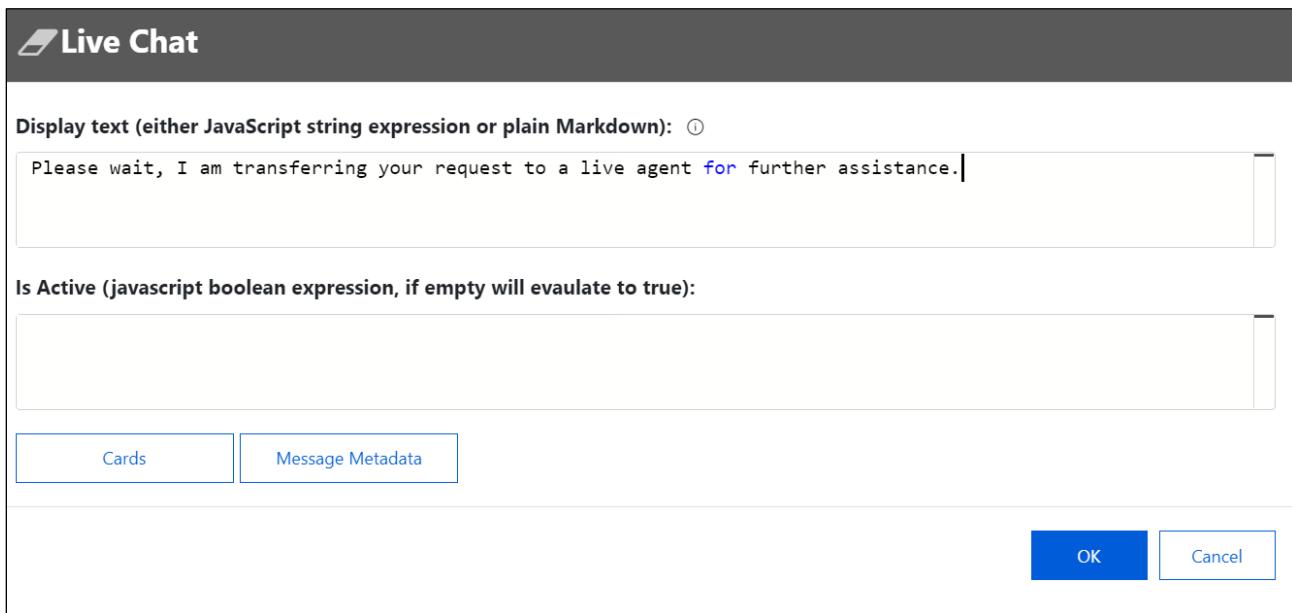


## Step 6: Invoke Live Agent Action

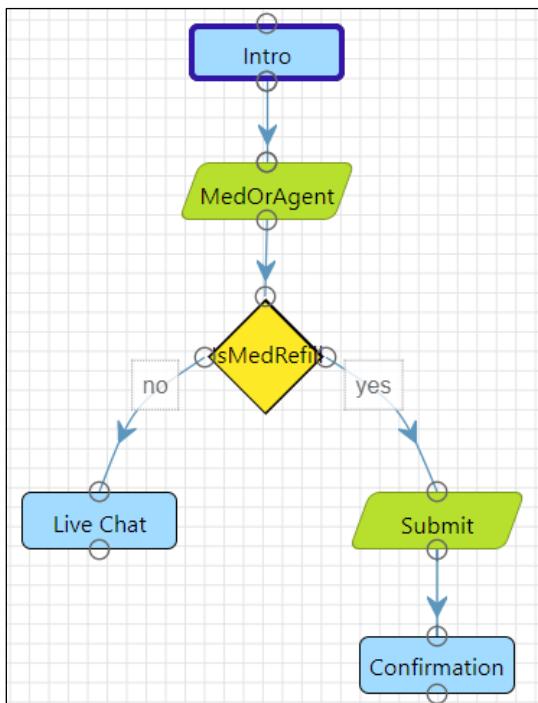
1. Add a **Statement** element to the canvas.



2. Enter **Display Text**: Please wait, I am transferring your request to a live agent for further assistance.
3. Rename the statement to **Live Chat**.

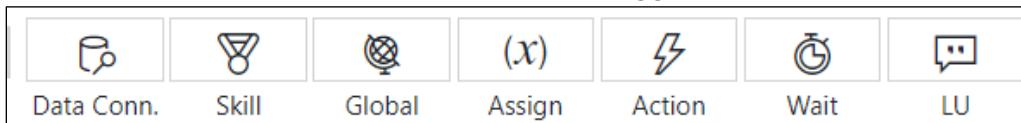


4. Click **OK** to return to the designer page.
5. **Connect** the **No** decision of the **IsMedRefill** branch to the **Live Chat** statement.



## Step 7: Add Action to Invoke Escalation

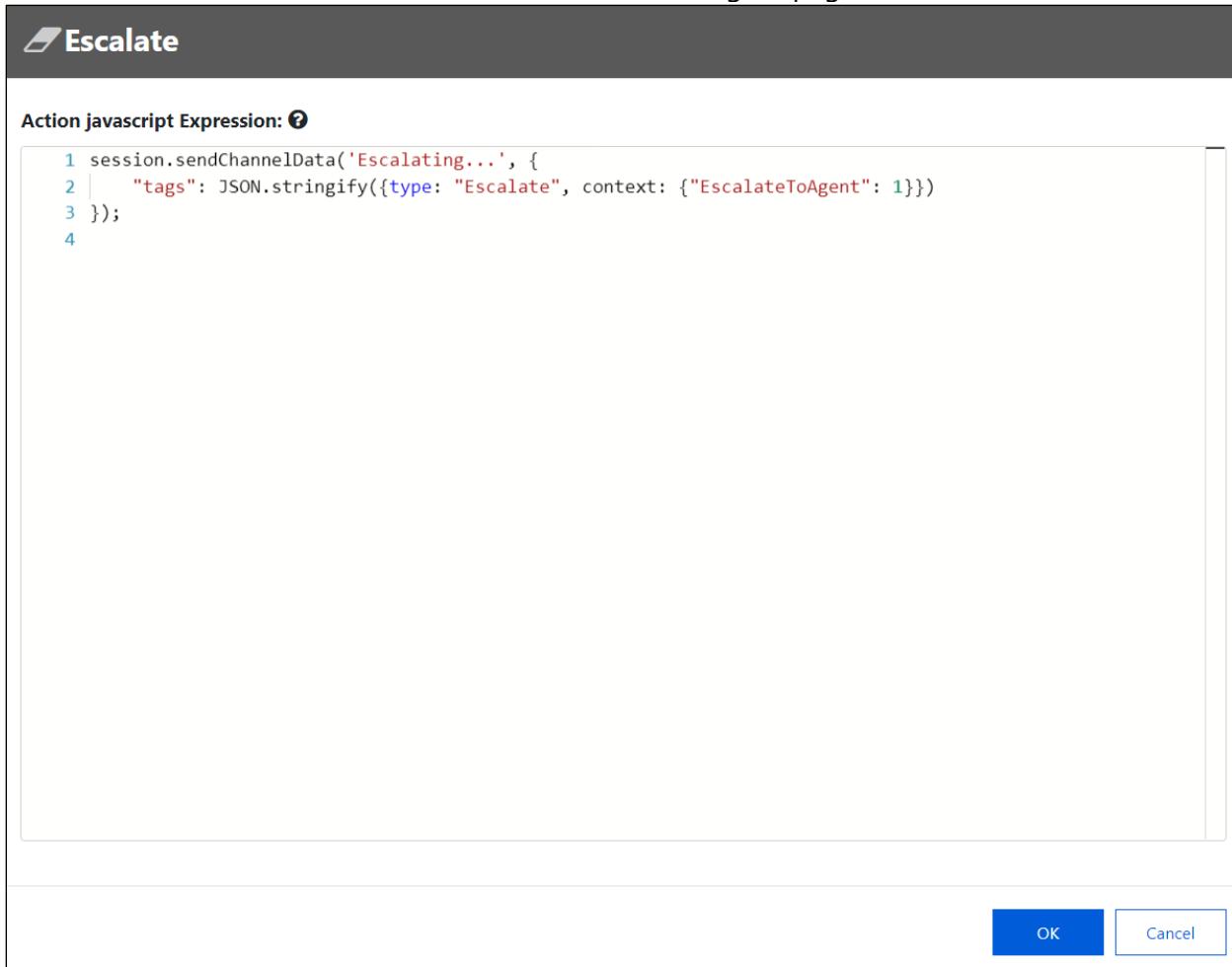
6. Add an **Action** element to the canvas, used to trigger an escalation to Omnichannel Live Agent



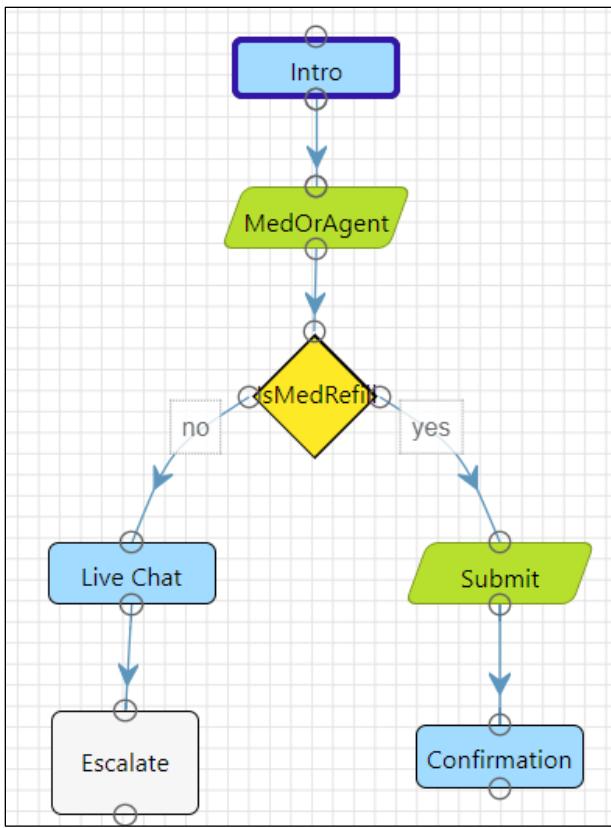
7. Add the following code in the action, which will trigger the Live agent chat:

```
session.sendChannelData('Escalating...', {  
    "tags": JSON.stringify({type: "Escalate", context: {"EscalateToAgent": 1}})  
});
```

8. Name the action **Escalate**. Click **OK** to return to the designer page.

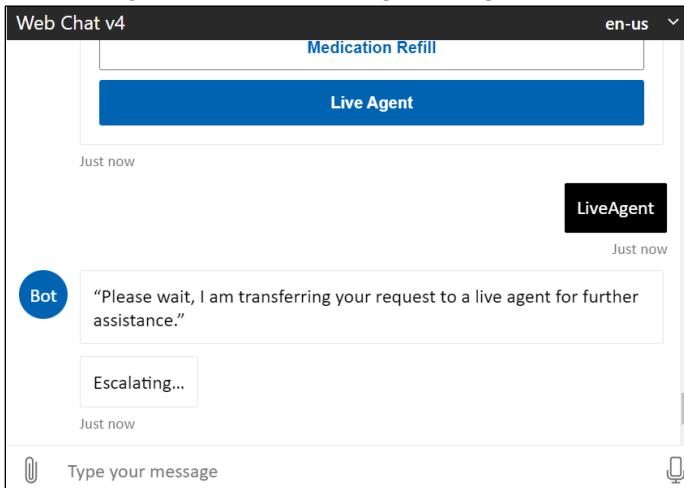


9. **Connect** the **Live Chat** to the new **EscalateToAgent** action. You completed the final connection!

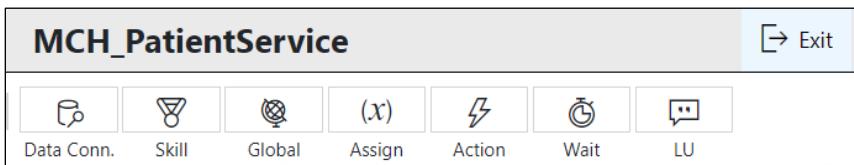


10. **Save** and **run** your scenario to see the full scenario output.

11. Test all logical paths. Selecting Live Agent in the authored card should show the escalation action.



12. **Exit** the MCH\_PatientService scenario editor.



## Task 2: Create MCH\_PatientServiceWelcome Scenario

In this task, you will create another bot scenario called **MCH\_PatientServiceWelcome** to invoke the **MCH\_PatientService** scenario.

1. On the Azure Health Bot scenarios page, select **+New** to create another new scenario

The screenshot shows the 'Scenario Management' page with the following interface elements:

- Top navigation bar with buttons: + New, Refresh, Import, Export, Delete.
- Title: Scenario Management.
- Sub-instruction: Create and manage custom scenarios for your bot instance. [Learn more](#).
- Table header: Active, Name, Scenario ID, Description.
- Table data:
  - One row is visible with the name 'MCH\_PatientSer...', Scenario ID 'MCH\_PatientServ...', and a partially visible description.

2. Provide the following details for the new scenario:

- a. **Name:** MCH\_PatientServiceWelcome
- b. **Scenario ID:** MCH\_PatientServiceWelcome
- c. Select **Create**.

The 'New Scenario' dialog box contains the following fields:

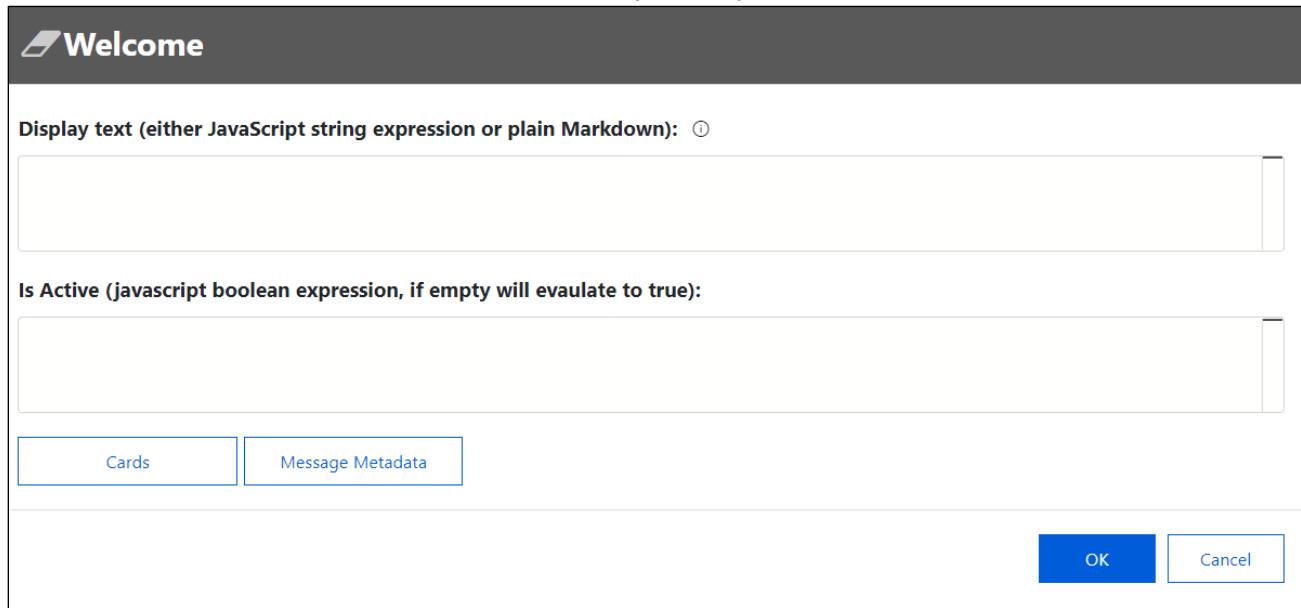
- Name\***: MCH\_PatientServiceWelcome
- Description**: (empty)
- Scenario ID\***: MCH\_PatientServiceWelcome
- Returning Message**: (empty)
- Interrupting scenario**: (switch is on)
- Breaking scenario**: (switch is off)

At the bottom are two buttons: **Create** (highlighted in blue) and **Cancel**.

3. On the scenario editor designer, add a **Statement** element.

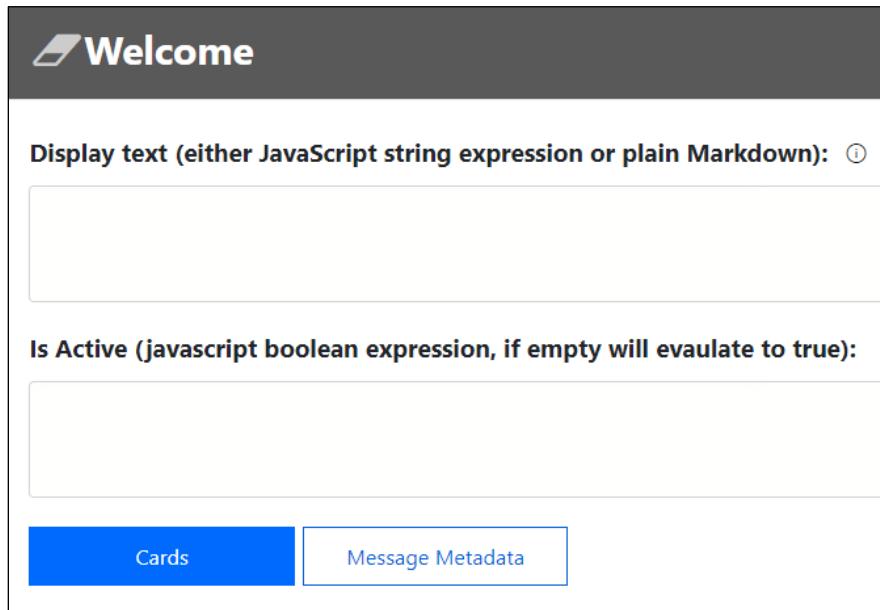


4. Rename the statement **Welcome**. Do not add any Display text as we will show it in the card instead.



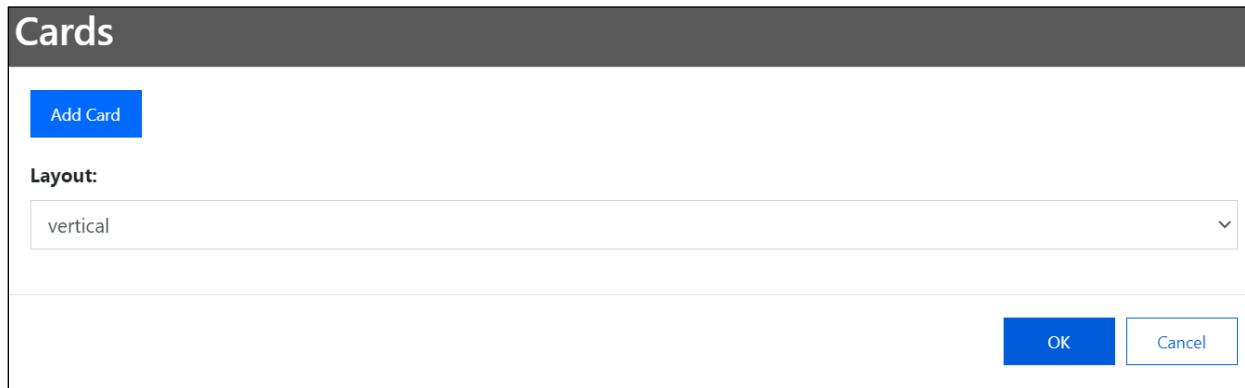
The screenshot shows the configuration dialog for a 'Welcome' statement. At the top, there's a title bar with a pencil icon and the word 'Welcome'. Below it is a section labeled 'Display text (either JavaScript string expression or plain Markdown):' with a text input field. Underneath is another section labeled 'Is Active (javascript boolean expression, if empty will evaluate to true):' with a text input field. At the bottom of the dialog are two buttons: 'Cards' (highlighted in blue) and 'Message Metadata'. In the bottom right corner are 'OK' and 'Cancel' buttons.

5. Select **Cards**.



This screenshot is identical to the one above, showing the 'Welcome' configuration dialog. The difference is that the 'Cards' button at the bottom has been clicked, turning blue, while the 'Message Metadata' button remains white.

6. Select **Add Card**.



The screenshot shows the 'Cards' configuration dialog. The title bar says 'Cards'. Below it is a 'Add Card' button. A 'Layout:' section contains a dropdown menu set to 'vertical'. At the bottom are 'OK' and 'Cancel' buttons.

7. Choose **HeroCard**. Add **Title**: Welcome to Lamna Healthcare Patient Service Portal

**Card**

**Card Type:**  
HeroCard

**Image Url:**  
 

**Title:**  
Welcome to Lamna Healthcare Patient Service Portal

**Sub Title:**

**Actions**

Add Action

**OK** **Cancel**

8. Select **Add Action** and provide the following details:

- Action type:** imBack
- Action value:** "begin MCH\_PatientService"
- Action title:** "Lamna Healthcare Support"

**Card**

**Card Type:**  
HeroCard

**Image Url:**  
 

**Title:**  
Welcome to Lamna Healthcare Patient Service Portal

**Sub Title:**

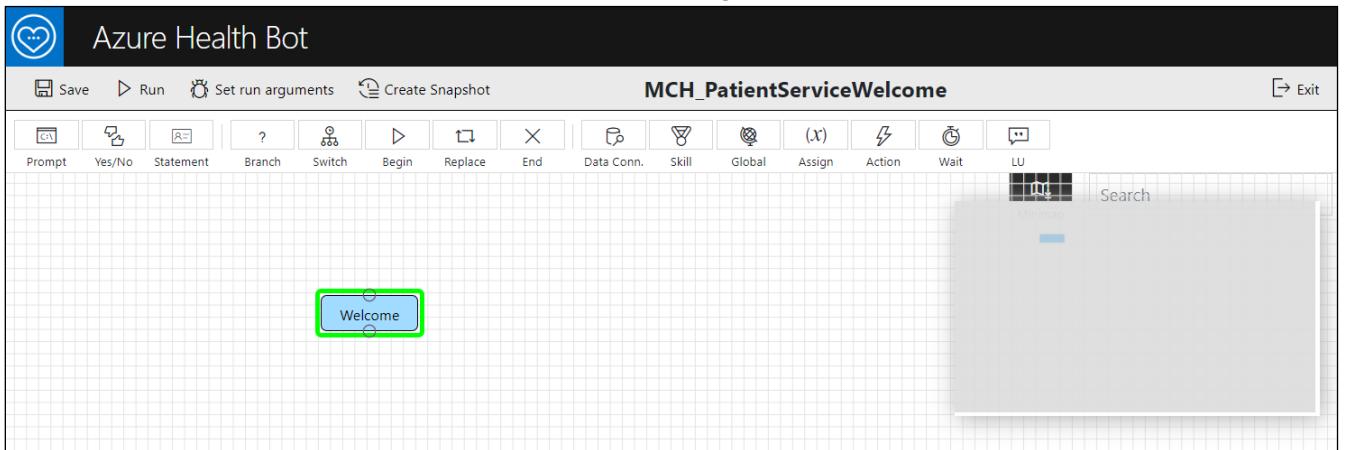
**Actions**

Action type	Action value	Action title
imBack	"begin MCH_PatientService"	"Lamna Healthcare Support" 

Add Action

**OK** **Cancel**

9. Click **OK** and view your completed scenario. This will be used to kick off the conversation and allow the other MCH\_PatientService scenario to be invoked through the authored card.



5. **Save** and **run** to test your bot scenario **MCH\_PatientServiceWelcome** scenario in the Web Chat.

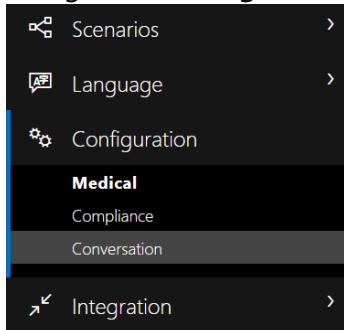


6. **Exit** the scenario designer.

## Task 3: Configure Welcome Scenario as Automatic

In this task, we will set the MCH\_PatientServiceWelcome to be the "Automatic Welcome Scenario" in settings. This will always trigger the welcome scenario when a user starts a conversation with the **Azure Health Bot**.

1. Navigate to **Configuration > Conversation**



2. In the **Interactions** tab, scroll down to the **Automatic Welcome** section.

A screenshot of the Azure Health Bot interactions page. The top navigation bar shows the bot name "iaduser99-healthbot". The left sidebar is identical to the one in the previous screenshot. The main content area is titled "Interactions" and contains sections for "Default retry message (date prompt)" and "Default retry message (attachment prompt)". Below these is the "Automatic welcome" section, which includes a sub-section for "Automatic welcome message" with a large text input field and an "Automatic welcome scenario" dropdown. The dropdown currently shows "\*\* scenario not selected \*\*" with a dropdown arrow. A note at the bottom states: "Important: Works only with scenarios that display a single statement step".

3. In the **Automatic welcome scenario** dropdown, select the **MCH\_PatientServiceWelcome** scenario.

A screenshot of the "Automatic welcome" configuration dialog. It shows the "Automatic welcome message" input field and the "Automatic welcome scenario" dropdown. The dropdown now shows "MCH\_PatientServiceWelcome" with a dropdown arrow. The "Important" note from the previous screenshot is also present here.

## Task 4: Test Health Bot Escalation from Power Apps Portal to Dynamics 365 Omnichannel

1. Navigate to Power Apps and click to open **Lamna Healthcare Patient Portal**.

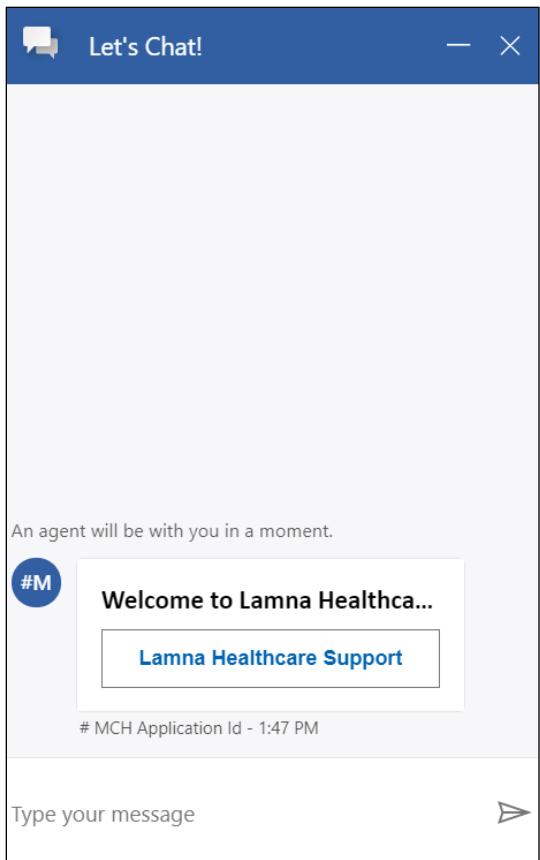
The screenshot shows the Microsoft Power Apps portal interface. On the left is a navigation sidebar with options like Home, Learn, Apps, Create, Data, Flows, Chatbots, AI Builder, and Solutions. The 'Apps' section is selected. The main area displays a list of apps under the heading 'Apps Component libraries (preview)'. A yellow banner at the top right states: 'Your trial portal app will expire in 10 days. To keep it, convert it to production.' Below this, there is a table with columns: Name, Modified, and Owner. The 'Name' column lists several apps, with 'Lamna Healthcare Patient Portal' highlighted by a red box.

Name	Modified	Owner
Lamna Healthcare Patient Portal	2 wk ago	SYSTEM
Patient Service Center	23 h ago	K Venkat
Customer Service Hub	1 wk ago	SYSTEM
Healthcare Administration	2 wk ago	K Venkat
Portal Management	2 wk ago	K Venkat

2. You should see the Health Bot "Let's Chat" button in the lower right-hand corner of the screen. This means the chat widget was successfully embedded into the Customer Self-service portal.

The screenshot shows the Contoso Customer Self-Service portal. At the top, there is a header with 'Contoso, Ltd.' and navigation links for Knowledge Base, Forums, My Support, and a user profile for 'Autumn Atkins'. Below the header is a banner with the text 'CONTOSO CUSTOMER SELF - SERVICE'. The main content area features a 'Most Popular' section with three categories: 'Most Popular Articles', 'Most Recent Articles', and 'Top Rated Articles'. At the bottom right of the page, there is a blue 'Let's Chat' button with a speech bubble icon and the text 'We're Online'.

3. When you click the chat widget, bot will trigger a welcome scenario message we created and set as the default welcome message (**MCH\_PatientServiceWelcome**).



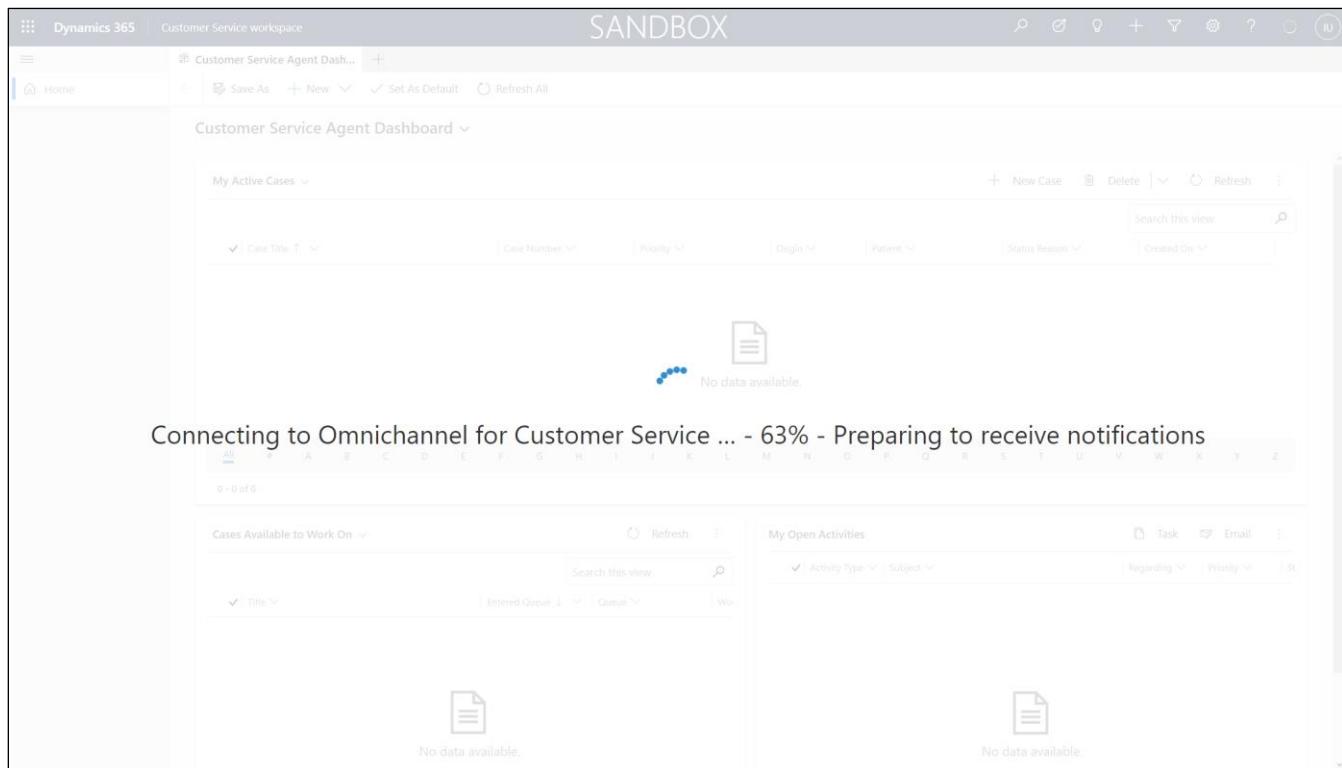
4. Navigate back to Power Apps and open **Customer Service Workspace**.

Icon	Name
Checkmark	Customer Service workspace
Heart	Customer Service Hub

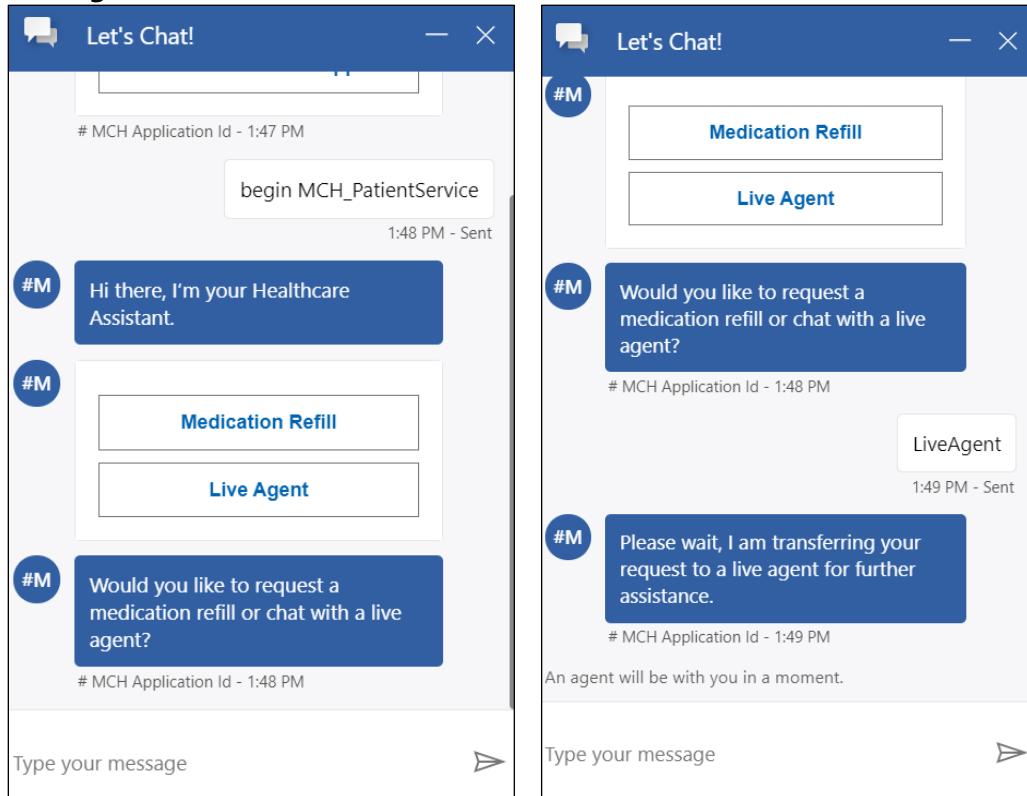
**Note:** Omnichannel for Customer Chat Widget will work only when you see the presence status is enabled.

There should be a splash loading screen that goes through multiple steps and then displays the status indicator as available once loaded. (Status is enabled when green with checkmark in circle)

Splash screen:

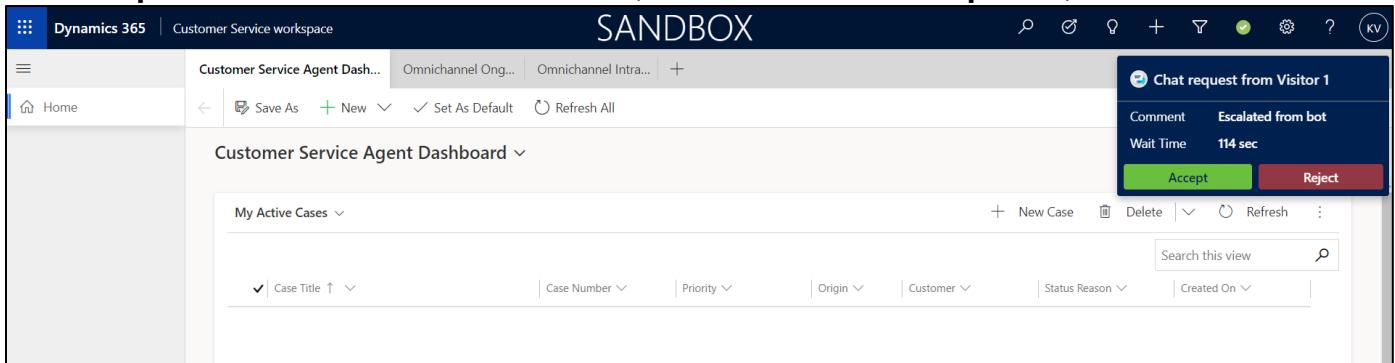


5. In the Health Bot in Lamna Healthcare Patient Portal, click **Lamna Healthcare Support** button, then the **Live Agent** button to witness the escalation into Omnichannel to chat with a live agent (your user!)

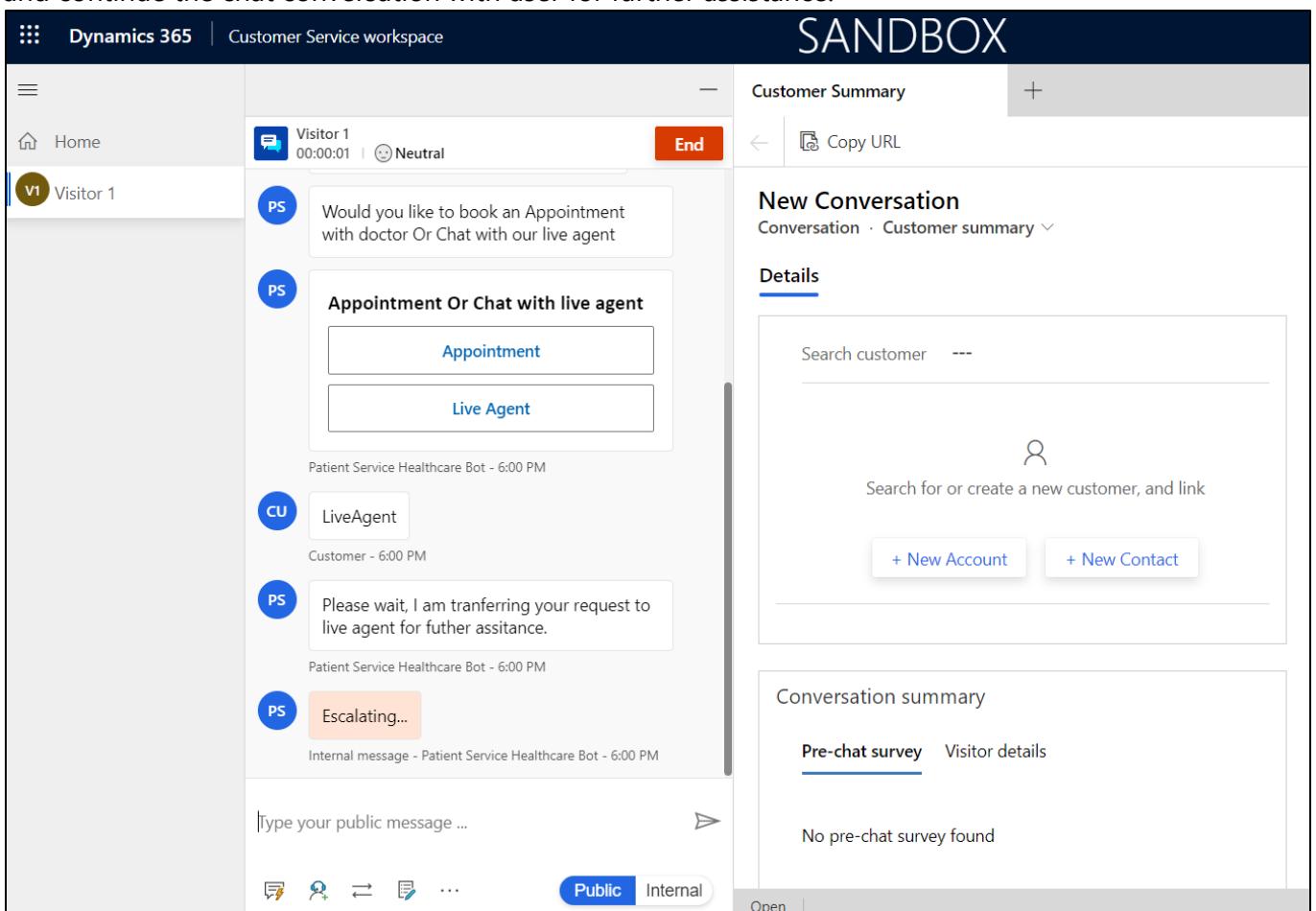


6. Navigating back to Omnichannel for Customer Service, your user as the **Live Agent** should receive an incoming notification with **Accept/Reject** options for that chat.

7. Click **Accept** to connect and chat with customer (In this case chat with the **patient**).



8. As soon as Live Chat Agent accepts the incoming chat notification, Omnichannel for Customer Service has opened a **Live Chat Widget** and Agent would be able to see the entire bot conversation with user and continue the chat conversation with user for further assistance.



**Congratulations!** You have successfully extended the Azure Health Bot with custom scenarios and tested the end-to-end escalation scenario from a patient using the Azure Health Bot in Power Apps Portals to chatting with a Live Agent in Omnichannel for Customer Service.

# Summary

**Nice work!** You have completed **Lab 04 – Azure Health Bot**.

In this lab, you learned how to do the following:

- Set up Azure Health Bot
- Configure Dynamics 365 Customer Service Omnichannel Live Chat
- Embed Azure Health Bot in a Power Apps Portal
- Extend Azure Health Bot with custom scenarios