



# Microsoft Cloud for Healthcare Industry Labs

## Lab 04: Azure Health Bot

Step-by-Step Lab

September 2021

# Contents

<b>Overview .....</b>	<b>3</b>
Learning Objectives.....	3
Prerequisites .....	3
Azure Health Bot.....	3
Industry Prioritized Scenarios .....	4
Atkins Family 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 Omnichannel Queues .....	23
Task 4: Update Live Work Stream with Context Variables and Routing Rules .....	27
Task 5: Create Chat Widget for Health Bot.....	31
<b>Exercise 3: Embed Health Bot in Power Apps Portal .....</b>	<b>33</b>
<b>Exercise 4: Extend Azure Health Bot with Custom Scenarios .....</b>	<b>36</b>
Task 1: Create MCH_PatientService Scenario.....	37
Task 2: Create MCH_PatientServiceWelcome Scenario .....	54
Task 3: Configure Welcome Scenario as Automatic .....	58
Task 4: Test Health Bot Escalation from Power Apps Portal to Dynamics 365 Omnichannel .....	59
<b>Summary .....</b>	<b>63</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.

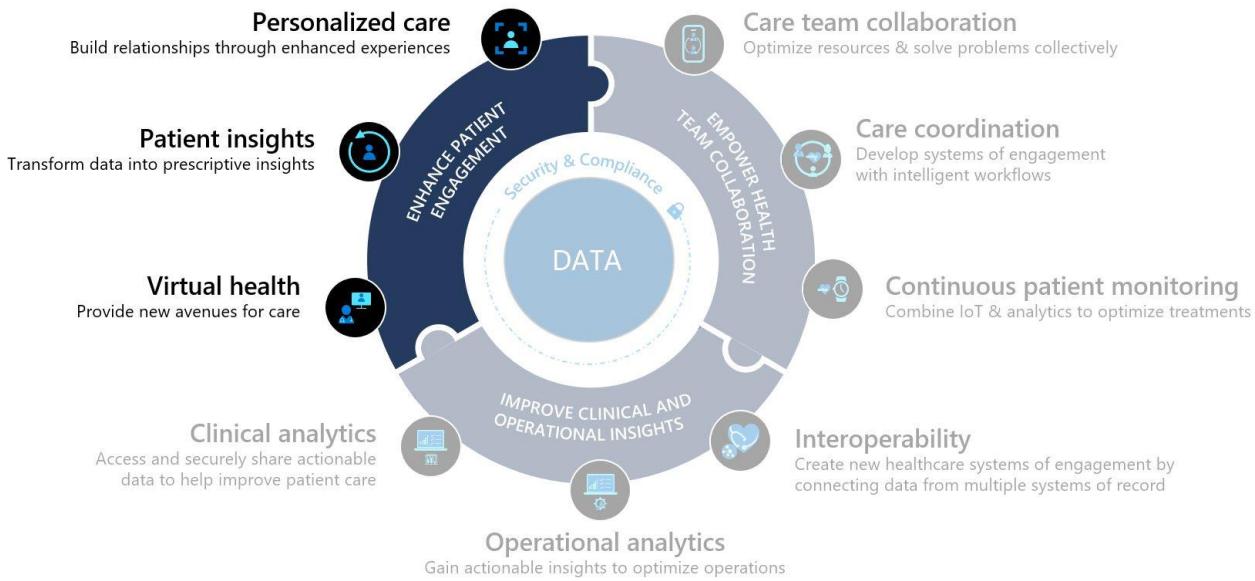
Using the service, healthcare organizations can build a "health bot instance" and integrate it with their systems that patients, nurses, doctors, and other representatives interact with. Building an instance allows you to:

- 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.



## Atkins Family 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, such as entering 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

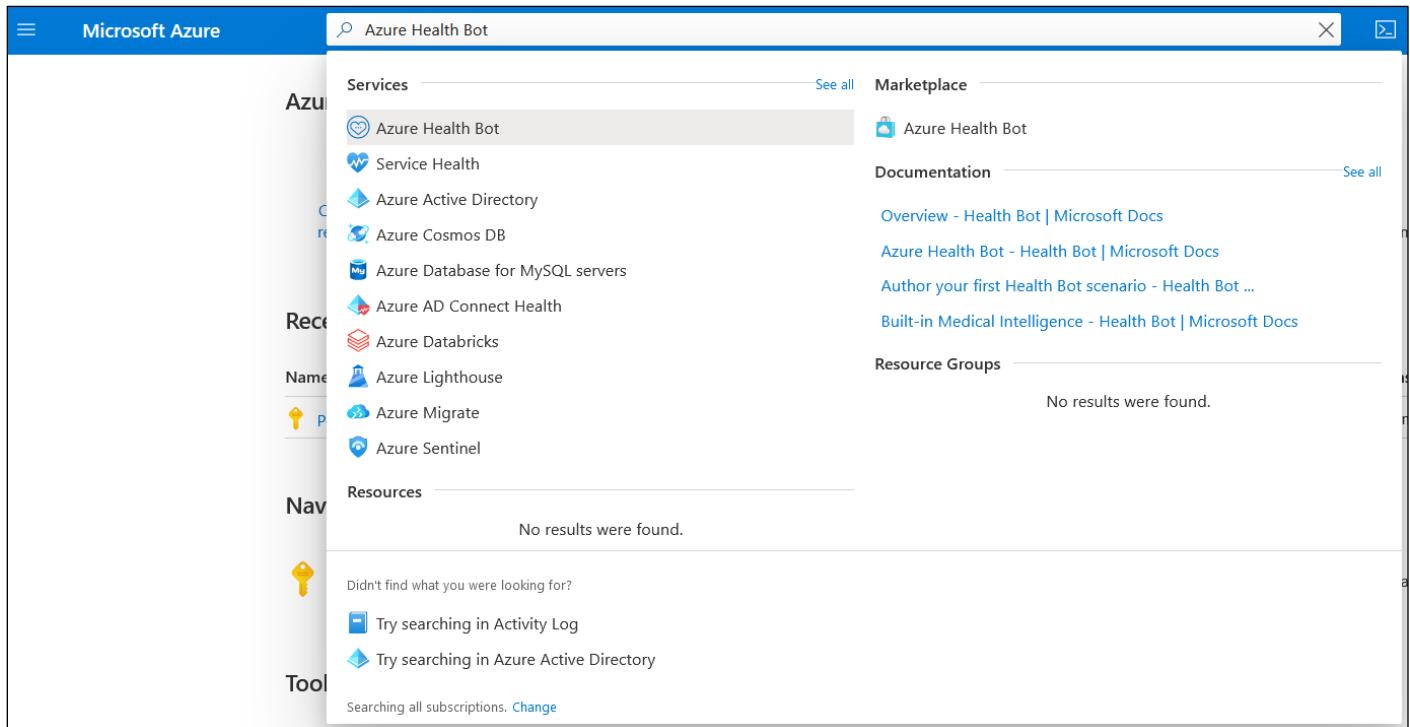
In this exercise, you will do the following:

- 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

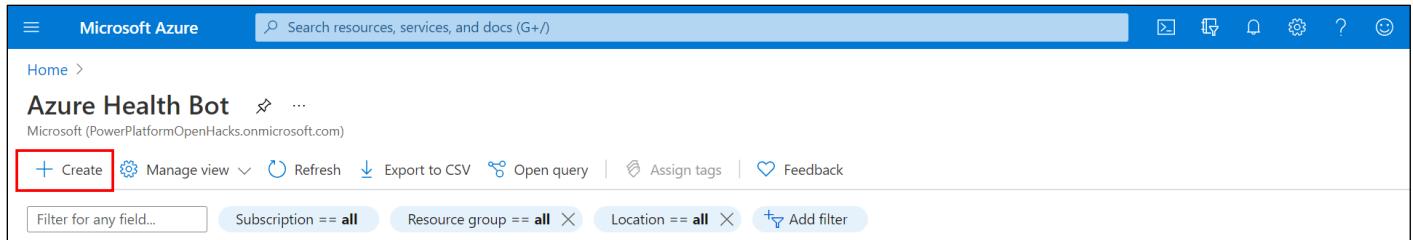
**Azure Health Bot** empowers developers in healthcare organizations to build and deploy AI-powered, compliant, conversational healthcare experiences at scale. It combines built-in medical database with natural language capabilities to understand clinical terminology and can be easily customized to support 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.
3. Select "**Azure Health Bot**" from the search results.



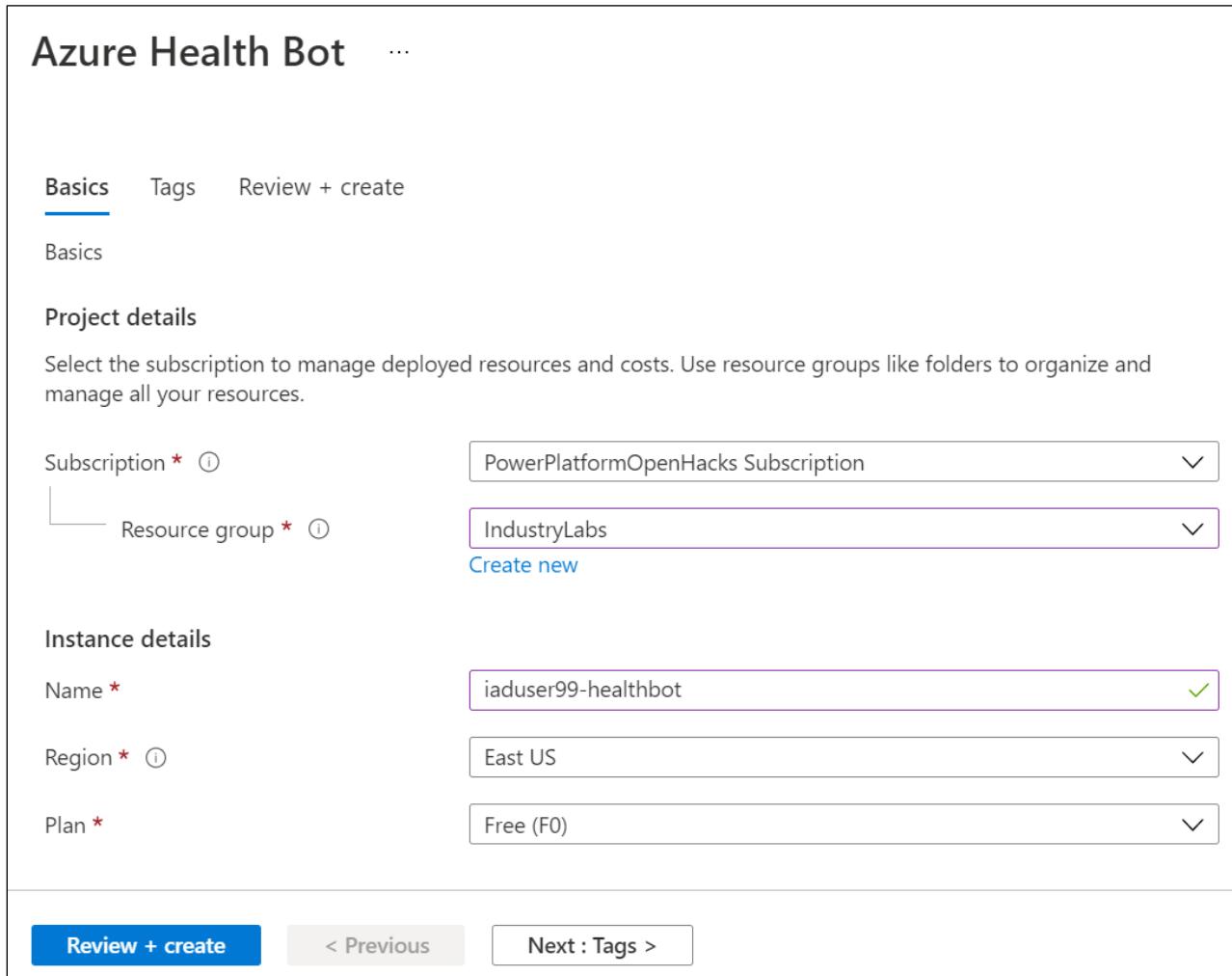
4. Click **Create** button to create a new Azure Health Bot instance.



The screenshot shows the Microsoft Azure portal interface. At the top, there's a search bar and several navigation icons. Below the header, the URL 'Azure Health Bot' is visible, followed by a note: 'Microsoft (PowerPlatformOpenHacks.onmicrosoft.com)'. A red box highlights the 'Create' button, which is located in the top-left corner of the main content area. Other buttons in the row include 'Manage view', 'Refresh', 'Export to CSV', 'Open query', 'Assign tags', and 'Feedback'. Below the main row are filter options: 'Filter for any field...', 'Subscription == all', 'Resource group == all', 'Location == all', and 'Add filter'.

5. You will be redirected to the Azure Health Bot page. Enter the following information:

- Subscription:** PowerPlatformOpenHacks Subscription
- Resource Group:** IndustryLabs
- Name:** iaduser[x]-healthbot (e.g., iaduser01-healthbot)
- Region:** East US
- Plan:** Free (F0)



The screenshot shows the 'Basics' tab of the Azure Health Bot creation form. The 'Subscription' dropdown is set to 'PowerPlatformOpenHacks Subscription'. The 'Resource group' dropdown is set to 'IndustryLabs' with a 'Create new' link below it. In the 'Project details' section, the 'Name' field is filled with 'iaduser99-healthbot'. Under 'Instance details', the 'Region' is 'East US' and the 'Plan' is 'Free (F0)'. At the bottom, there are navigation buttons: 'Review + create' (highlighted in blue), '< Previous', and 'Next : Tags >'.

6. Select **Review + Create**.

- On the Review and create page, verify your details are correct as Azure validates your Health Bot. When the create button is enabled after validation passes, click **Create**.

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

The screenshot shows the 'Review + create' step of the Azure Health Bot creation wizard. At the top, a green bar indicates 'Validation Passed'. Below it, tabs for 'Basics', 'Tags', and 'Review + create' are visible, with 'Review + create' being the active tab. The 'TERMS' section contains a detailed legal agreement. The 'Basics' section displays the following configuration:

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

At the bottom, there are buttons for 'Create' (in blue), '< Previous', 'Next', and 'Download a template for automation'.

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

The screenshot shows the 'Overview' page for the Microsoft.HealthBot-20210915192409 deployment. The left sidebar includes 'Overview' (selected), 'Inputs', 'Outputs', and 'Template'. The main area displays deployment details:

- 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

A 'Deployment details' table is shown:

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

9. 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.

10. 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.*

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

**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 blue heart icon and the text "Welcome to your Health Bot Instance". Below this, there are 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.

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

This screenshot is identical to the one above, but the sidebar is fully expanded, revealing 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 central welcome message and scenario options, remains the same.

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

The screenshot shows the Azure Health Bot navigation bar. A red box highlights the "Conversation" item under the "Configuration" category in the sidebar. The other items in the sidebar are: Scenarios, Language, Configuration, Medical, Compliance, Integration, Analytics, Users, and Resources. The main content area is mostly blank, indicating the user has navigated away from the homepage.

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 main menu on the left has sections: Scenarios, Language, Configuration (with Medical, Compliance, Conversation sub-sections), Integration, Analytics, Users, and Resources. The 'Interactions' tab is selected and highlighted in blue. The central content area is titled 'Interactions' with the sub-instruction 'Configure built-in scenarios that interact with your end user.' Below this is a 'Global defaults' section containing five entries: 'Default reply for utterances that are not understood' (value: Sorry, it seems I can't answer this.), 'Default message when returning from interrupting scenarios' (value: Now back to the previous topic...), 'Default Error Message' (value: Oops. Something went wrong and we need to start over.), 'Default retry message (number prompts)' (value: I didn't recognize that as a number. Please enter a number.), and 'Default timeout message' (value: We are connecting you to an agent. Please wait...).

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 left sidebar are identical to the 'Interactions' screen. The central content area is titled 'Human Handoff' with the sub-instruction 'Configure scenarios that allow handoff to human agents.' Below this is a 'Human Handoff' configuration section. It features a toggle switch labeled 'Disabled' which is currently off. There are three dropdown menus: 'End user timeout' set to '10 minutes', 'Agent timeout' set to '15 minutes', and 'Waiting message' containing the placeholder text 'We are connecting you to an agent. Please wait...'. A 'Learn more' link is also present in the main content area.

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**.

Active	Channel
<input checked="" type="checkbox"/>	Web Chat
<input checked="" type="checkbox"/>	DirectLine
<input checked="" type="checkbox"/>	Microsoft Teams
<input checked="" type="checkbox"/>	Twilio
<input checked="" type="checkbox"/>	Facebook
<input checked="" type="checkbox"/>	Telegram
<input checked="" type="checkbox"/>	Alexa (preview)
<input checked="" 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.

### Microsoft Teams Channel

Connect your Health Bot to interact with Microsoft Teams users naturally through chat. [Learn how](#)

**Bot Id**

c7f90733-d6e3-4f3d-a95a-f28cfbc9b0b1 [Copied!](#)

[Save](#) [Close](#)

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

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 checked="" type="checkbox"/>	Twilio	
<input checked="" type="checkbox"/>	Facebook	
<input checked="" type="checkbox"/>	Telegram	
<input checked="" type="checkbox"/>	Alexa (preview)	
<input checked="" 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 results for "App Registrations". 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 Azure App registrations homepage. The top navigation bar includes the Microsoft Azure logo, a search bar, and links for Home, App registrations, and more. Below the navigation is a toolbar with buttons for New registration, Endpoints, Troubleshooting, Refresh, Download, Preview features, and Got feedback?. There are three tabs at the top: All applications, Owned applications, and Deleted applications (Preview). A search bar below the tabs contains the placeholder "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 "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 allows filtering by 'Display name' or 'Application ID'. The main area is titled 'All applications' with tabs for 'Owned applications' and 'Deleted applications (Preview)'. A search input field contains placeholder text: 'Start typing a name or Application ID to filter these results'. Below the search is a section for 'Display name' containing four entries, each with a small colored square icon (MC) 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 populated with 'MCH Application Id'. The results list shows one item: 'MCH Application Id' with a purple square icon and the display name '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. The left sidebar has sections for 'Overview', 'Quickstart', 'Integration assistant', 'Manage', 'Branding', and 'Authentication'. The 'Overview' section is currently selected. The main content area shows the 'Essentials' group with fields: 'Display name : MCH Application Id', 'Application (client) ID : [redacted]', 'Object ID : [redacted]', 'Directory (tenant) ID : [redacted]', and 'Supported account types : My organization only'. The 'Application (client) ID' field is highlighted with a red rectangle.

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

## Exercise 2: Configure Omnichannel Live Chat

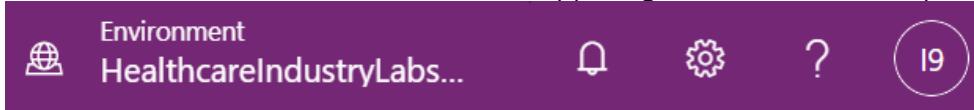
In this exercise, you will be configuring live chat for **Dynamics 365 Omnichannel for Customer Service**. Omnichannel for Customer Service offers a suite of capabilities that extend the power of Dynamics 365 Customer Service Enterprise to enable organizations to instantly connect and engage with their customers across digital messaging channels.

In the following tasks, you will complete the following:

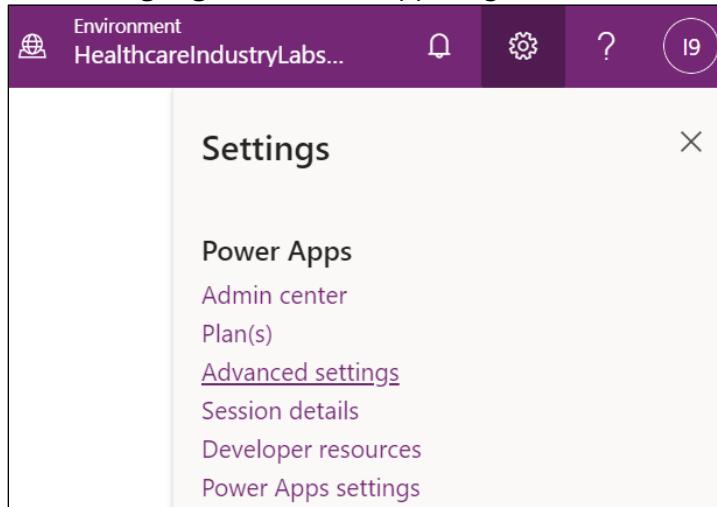
1. Assign Omnichannel agent security role
2. Create an Application User using the **MCH Application Id** and your **Bot ID**
3. Configure Queues for Bot and Agent Users
4. Configure a Context Variable and Routing rule to route the message either to a Bot or Agent.

### 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 the current selection), 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. At the top, there is a search bar containing the text "iad". Below the search bar is a header row with columns: "Full Name ↑", "Position", "Main Phone", "Business Unit", "Site", "Title", and "Primary Email". The main body of the grid contains one row for "IAD User 01", which includes the full name "IAD User 01", the email address "unq0ed694338a62465...", and the site "IADUser01@PowerPlatformOp...".

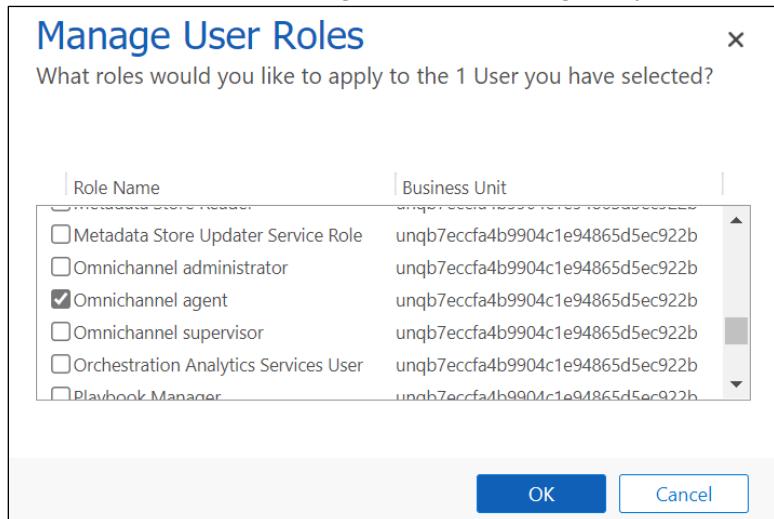
9. Select your user for the training and select **Manage Roles** on the top command bar.

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 SECURITY. The word "Sandbox" is also visible on the right side of the bar.

The screenshot shows a search results grid. At the top, there is a search bar containing the text "iad". Below the search bar is a header row with columns: "Full Name ↑", "Position", "Main Phone", "Business Unit", and "Site". The main body of the grid contains one row for "IAD User 01", which includes the full name "IAD User 01", the email address "unq0ed694338a62465...", and the site "IADUser01@PowerPlatformOp...". The checkbox next to the user's name is checked, indicating they are selected.

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 internal 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. While in Dynamics 365 in the User page, switch the view to **Application Users**.

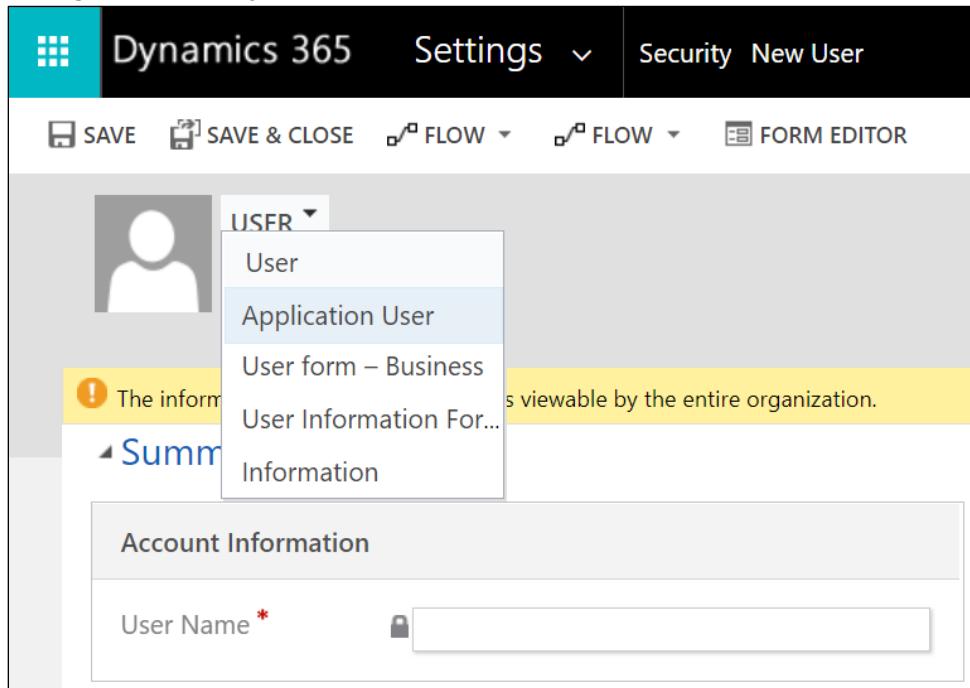
The screenshot shows the Dynamics 365 User search interface. The left sidebar has a tree view with 'System Views' expanded, showing options like '@Me', 'Access Mode Interactive Users', 'Administrative Access Users', 'Administrators', 'Agents', and 'All'. Below 'System Views' is a section titled 'Application Users' with the following options: 'Associated Record Team Members', 'Bot agents', 'Bot Users', 'By Me', 'Disabled Users', 'Disabled users consuming licenses', and 'Enabled Users'. The 'Bot Users' option is highlighted with a blue background.

2. Select (+) New button to create a new Application User.

The screenshot shows the Dynamics 365 Application Users list page. The top navigation bar includes 'Dynamics 365', 'Settings', 'Security', and a 'Sandbox' indicator. Below the navigation is a toolbar with buttons for '+ NEW', 'PROMOTE TO ADMIN', 'EMAIL A LINK', 'FLOW', 'RUN REPORT', 'EXCEL TEMPLATES', and 'EXPORT TO EXCEL'. The main area is titled 'Application Users' and displays a table of users. The columns are 'Full Name' (sorted by last name), 'Application I...', 'Azure AD Obj...', and 'Application I...'. The table contains three rows of data:

Full Name	Application I...	Azure AD Obj...	Application I...
Dynamics Marketing Customer Experience Platfo...	2220bbc4-45...	0814cd9d-ef1...	api://prod.cxp
FRE Omnichannel Omnichannel PVA Application	cd5f0174-51e...	902ad22f-fff1...	cd5f0174-51e...
Omnichannel Omnichannel for Customer Service	18cc9627-77...	26631d04-a8...	18cc9627-77...

3. Change the **form type** from User to **Application User** above the New User name.



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

The screenshot shows the 'New User' form for an Application User. At the top, it displays 'USER : APPLICATION USER' and the title 'New User'. A yellow warning message states: 'The information provided in this form is viewable by the entire organization.' The form is divided into sections: 'Summary' and 'User Information'. The 'Summary' section contains fields for 'User Name' (locked), 'Application ID \*' (disabled), 'Application ID URI' (locked), and 'Azure AD Object ID' (locked). The 'User Information' section contains fields for 'Full Name \*' (locked), 'Primary Email' (locked), and 'User type' (set to 'Application user').

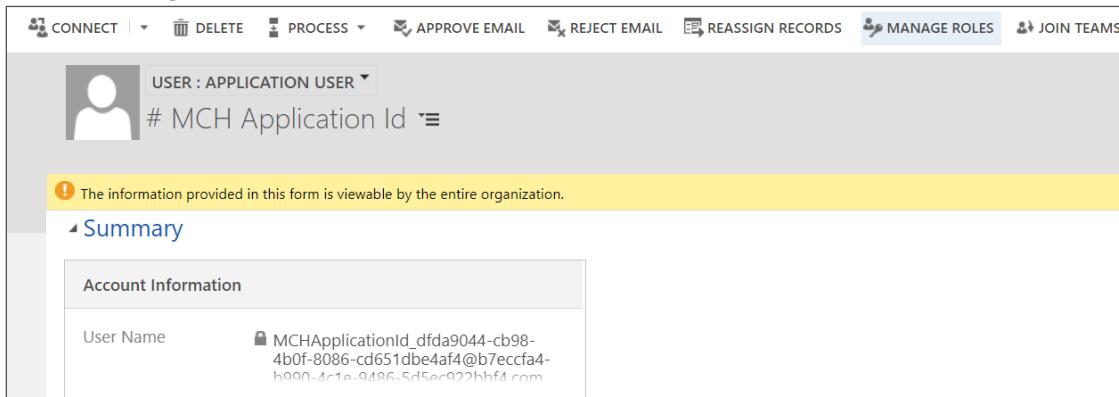
- On the **New User** record, enter or select the following information:
  - Application ID:** This is the Application (client) ID you obtained in the Azure Portal for the supplied MCH Application ID.
  - User type:** Select **Bot application user**. This will *display a new field* to store the Bot application 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 'New User' form for 'USER : APPLICATION USER'. At the top, there's a note: 'The information provided in this form is viewable by the entire organization.' Under the 'Summary' section, there are two tabs: 'Account Information' and 'User Information'. In the 'User Information' tab, the 'User type' is set to 'Bot application user' and the 'Bot application ID' field contains the value 'ef9a0771-20b2-4ba3-ab6b-f120e486c880'.

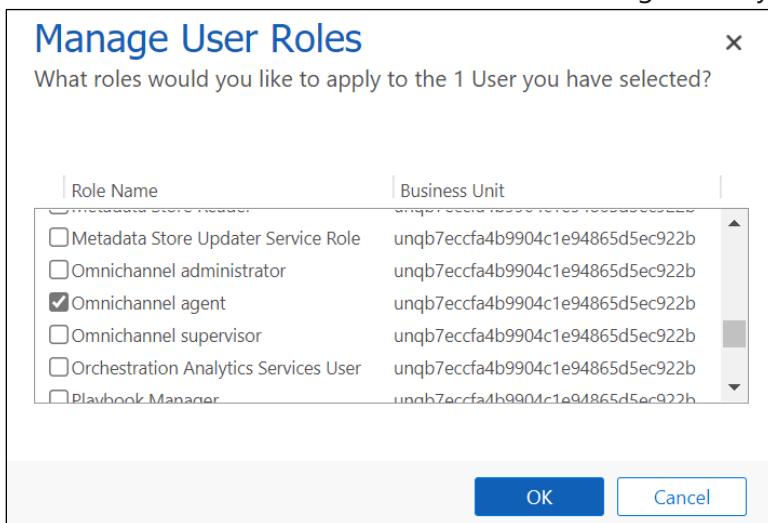
- Click **Save**. It will auto populate the other values in the record.

The screenshot shows the 'New User' form for 'USER : APPLICATION USER'. The 'User type' is set to 'Bot application user'. The 'Bot application ID' field contains the placeholder '# MCH Application Id'. In the 'User Information' tab, the 'Primary Email' field also contains the same placeholder '# MCH Application Id'.

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



8. 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.



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

## Task 3: Create and Configure Omnichannel Queues

In this task, you will create and configure the omnichannel queues necessary to communicate with the correct bot or agent depending on the situation.

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

Apps

Apps Component libraries (preview)

⚠️ 10 environment variables need to be updated. See [environment variables](#)

Icon	Name
Heart	Omnichannel admin center
Checkmark	Omnichannel Administration

2. Select **Queues** on the left navigation bar.

- Home
- Recent
- Pinned
- Channels
- Entity Records
- Chat
- Queues & Users
  - Queues
  - Users
  - Bots
  - Skills

3. Open **Default Messaging Queue**.

Omnichannel queues ▾		
✓ Name ↑ ▾	Queue type ▾	Created On ▾
Default entity queue	Entity	8/19/2021 6:26 PM
✓ Default messaging queue	Messaging	8/19/2021 6:26 PM

4. We will now associate the Default messaging queue with the Bot User so it will respond to incoming messages from customers without agent (human) intervention.

Select **Add Existing User** on the **User (Agents)** subgrid to add the Bot user you previously created.

The screenshot shows the 'Default messaging queue' details page. On the left, there's a 'SUMMARY' section with fields: Name (Default messaging queue), Priority (2,147,483,647), Queue type (Messaging), and Owner (empty). Below this is a note about preview terms. On the right, there's a 'Users (Agents)' subgrid with columns for Full Name, Capacity, and Business Unit. A button labeled 'Add Existing User' is visible at the top of the subgrid. The message 'No data available.' is displayed below the subgrid header.

5. In the Lookup Records pane, search for your **Bot User** (MCHApplicationId) created in the earlier task.

The screenshot shows the 'Lookup Records' pane with a search bar containing 'MCH'. The results list shows one record: '# MCH Application Id' followed by its value. There are buttons for 'Add' and 'Cancel' at the bottom.

6. Select the record from the list and click **Add**.

The screenshot shows the 'Lookup Records' pane with the same search results as the previous screenshot. The record '# MCH Application Id' is highlighted. The 'Add' button at the bottom is highlighted in blue, indicating it is the next action to be taken.

7. You should now see the Bot User (MCH Application Id) in the Users (Agents) list.

*Note: If your user does not populate after adding, make sure you have assigned the bot user the correct omnichannel agent security role (it may take up to 15 minutes for changes to take effect)*

SUMMARY	
Name	* Default messaging queue
Priority	* 2,147,483,647
Queue type	* Messaging
Owner	* Bot User (MCH Application Id)
<small>By using this feature, you acknowledge that this feature is in preview and you agree to the <a href="#">Preview Terms</a>. <a href="#">Learn more</a></small>	
Operating Hours	---

Users (Agents)	
# MCH Application Id	unqb7eccfa4b9904c1e94865d5ec922b..

8. Go back to the **Omnichannel queues** grid. Click **+ New** to create a new Queue.

<input checked="" type="checkbox"/> Show Chart <input type="button" value="+ New"/> <input type="button" value="Delete"/>   <input type="button" value="Refresh"/> <input type="button" value="Email a Link"/>   <input type="button" value="Flow"/>	
Omnichannel queues ▾	
<input checked="" type="checkbox"/> Name ↑	Queue type ▾
Default entity queue	Entity
Default messaging queue	Messaging

9. Give the new Queue the following details:

- Name:** "Escalate To Human"
- Priority:** 1 (lower than default queue)
- Click Save.**

SUMMARY	
Name	* Escalate To Human
Priority	* 1
Queue type	* Messaging
Owner	* IAD User 99
<small>By using this feature, you acknowledge that this feature is in preview and you agree to the <a href="#">Preview Terms</a>. <a href="#">Learn more</a></small>	
Operating Hours	---

10. A Users (Agents) **subgrid should appear** on the right and your **user should be automatically added** to the list. If your user account is not on the list, add it through the Add Existing User button now.

The queue **Escalate To Human** is created to 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.

The screenshot shows the 'Escalate To Human' Queue configuration page. On the left, there's a 'SUMMARY' section with fields: Name (\* Escalate To Human), Priority (\* 1), Queue type (\* Messaging), and Owner (\* IAD User 99). Below this is a note about preview features. On the right, there's a 'Users (Agents)' subgrid with columns: Full Name (sorted by ascending), Capacity (sorted by descending), and Business Unit (sorted by descending). One user, 'IAD User 99', is listed. At the top right of the subgrid, there are buttons for 'Add Existing User' and more options.

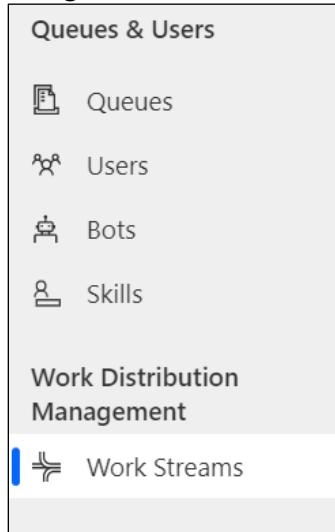
**Congratulations!** You have created the necessary queue to escalate to human agent and added the appropriate users to each messaging queue.

## Task 4: Update Live Work Stream with Context Variables and Routing Rules

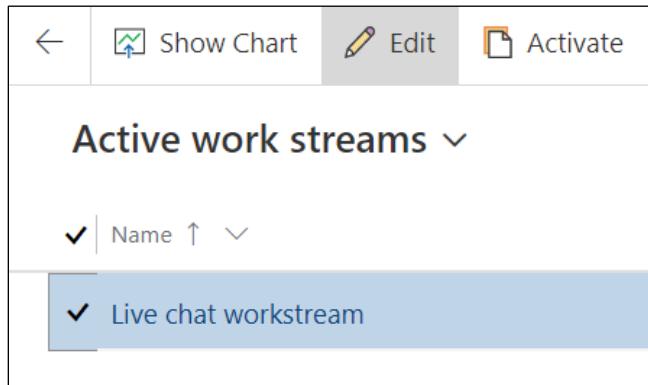
In this task, we will set up basic chat routing. This will allow for users to chat with a bot user in certain cases and a live human agent in other scenarios. The routing rules will allow chat to behave as follows:

- **Route to Bot:** Initial customer conversation is through Health Bot in the default messaging queue. When the chat bot is first opened, route to Default queue which only contains the bot user (agent).
- **Human Routing Rule:** When context variable **EscalateToAgent** is present and set to 1, we route to the queue that has only human users (agents) who can take over conversation.

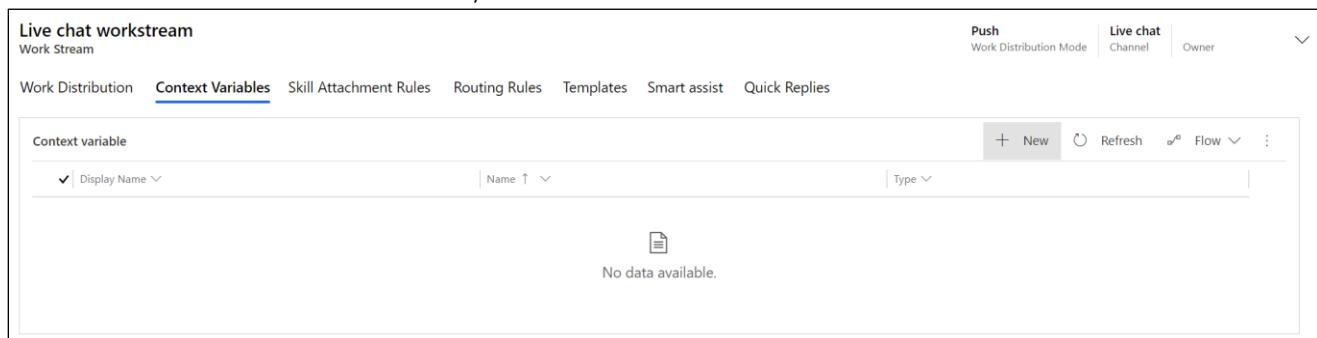
1. Navigate to **Work Streams**.



2. Select and edit the **Live chat workstream**.



3. In the **Live chat workstream** record, select the **Context Variables** tab. Select **+ New**.



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

- a. **DisplayName:** EscalateToAgent
- b. **Name:** EscalateToAgent
- c. **Type:** Number

Quick Create: Context variable

Display Name	<b>EscalateToAgent</b>
Name	* EscalateToAgent
Owner	*  IAD User 99
Type	* Number

**Save and Close** | **Cancel**

5. Click **Save and Close**.

6. You should now see the new Context Variable in the Live chat workstream.

Live chat workstream

Work Stream

Push  
Work Distribution Mode | Live chat | Channel | Owner

Work Distribution    **Context Variables**    Skill Attachment Rules    Routing Rules    Templates    Smart assist    Quick Replies

Context variable

+ New	Refresh	Flow	⋮
Display Name	Name ↑	Type	⋮
EscalateToAgent	EscalateToAgent	Number	⋮

7. Select the **Routing Rules** tab. Click **+ Add** to create a new routing rule.

Live chat workstream

Work Stream

Push  
Work Distribution Mode | Live chat | Channel | Owner

Work Distribution    Context Variables    Skill Attachment Rules    **Routing Rules**    Templates    Smart assist    Quick Replies

Rule items

+ Add	Flow	Run Report	⋮
Name	Description	Modified On	⋮

No data available.

8. Create the new Health Bot routing rule with the following details:

- Name:** ToHealthcareBot
- Queue:** Default messaging queue
- No Conditions.

The screenshot shows the 'New Rule Item' dialog for a routing rule named 'ToHealthcareBot'. The 'General' tab is selected. In the 'General Information' section, the name is set to 'ToHealthcareBot', owner is 'IAD User 99', work stream is 'Live chat workstream', and queue is 'Default messaging queue'. The 'Condition' section is empty, showing an 'OR' operator and a '+ Condition' button.

9. Select **Save & Close**. On the Live chat workstream, click **+ Add** to add another new Routing Rule.

The screenshot shows the 'Live chat workstream' page with the 'Routing Rules' tab selected. A single rule item 'ToHealthcareBot' is listed, created by 'IAD User 99' on 9/16/2021 at 12:39 AM. The list includes columns for Name, Description, and Modified On.

10. Create the new Omnichannel Agent routing rule with the following details:

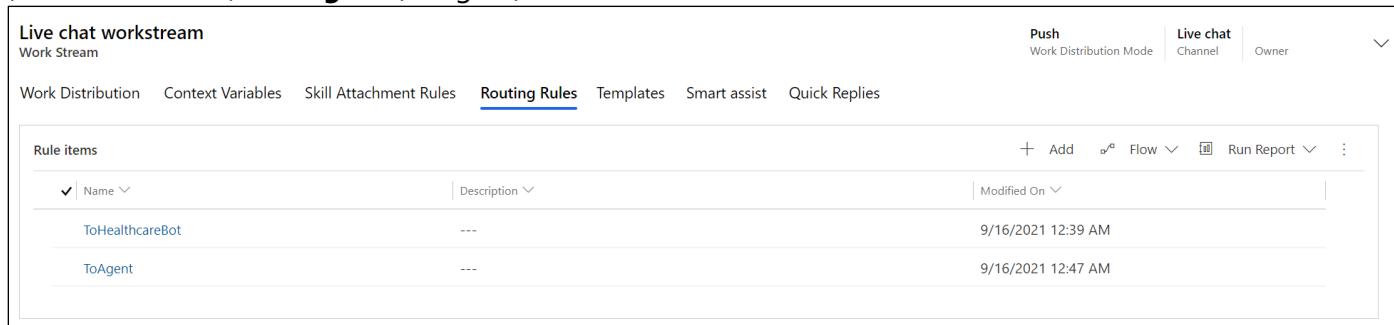
- Name:** ToAgent
- Queue:** EscalateToHuman
- Add Condition:** Context Variable "EscalateToAgent = 1"

The screenshot shows the 'New Rule Item' dialog for an omnichannel agent routing rule named 'ToAgent'. The 'General' tab is selected. In the 'General Information' section, the name is set to 'ToAgent', owner is 'IAD User 99', work stream is 'Live chat workstream', and queue is 'Escalate To Human'. The 'Condition' section contains a single condition: 'Context variable Equals 1'.

The screenshot shows the 'Condition' configuration dialog. It displays the same condition: 'Context variable Equals 1'. The dialog includes buttons for AND, OR, Ungroup, Delete, and Reset.

11. Select **Save & Close**.

12. On the Live chat workstream, you should now see the two **Routing Rules** we created for **Bot** (ToHealthcareBot) and **Agent** (ToAgent).



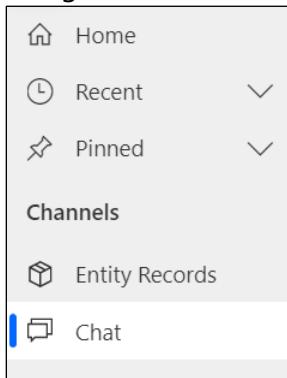
The screenshot shows the 'Live chat workstream' interface with the 'Work Stream' tab selected. At the top, there are tabs for 'Work Distribution', 'Context Variables', 'Skill Attachment Rules', 'Routing Rules' (which is underlined, indicating it's the active tab), 'Templates', 'Smart assist', and 'Quick Replies'. On the right side, there are buttons for 'Push' (Work Distribution Mode), 'Live chat Channel', 'Owner', and a dropdown menu. Below the tabs, there's a section titled 'Rule items' with a table. The table has columns for 'Name' (with a dropdown arrow), 'Description' (with a dropdown arrow), and 'Modified On'. There are two entries: 'ToHealthcareBot' was modified on 9/16/2021 12:39 AM, and 'ToAgent' was modified on 9/16/2021 12:47 AM. At the bottom of the table are buttons for '+ Add', 'Flow', 'Run Report', and a more options menu.

Name	Description	Modified On
ToHealthcareBot	---	9/16/2021 12:39 AM
ToAgent	---	9/16/2021 12:47 AM

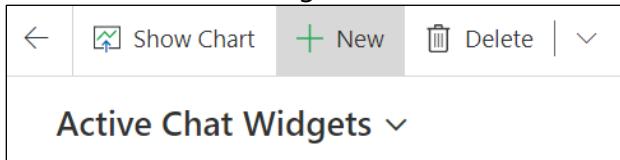
**Congratulations!** You have created the proper context variable and routing rules that will allow customers to begin conversation with a health bot and escalate to a human agent.

## Task 5: Create Chat Widget for Health Bot

1. Navigate to Chat.



2. Select +New Chat Widget.



3. Give the Chat Widget a Name (ex: Patient Portal Chat Widget).

New Chat Widget

General settings Surveys Conversation options Design Location

Name \* Patient Portal Chat Widget

Language \* English - United States

Agent display name Full name

Authentication settings ---

File attachments

Enable file attachments for customers No

Enable file attachments for agents No

Customer waiting

Show position in queue No

Show average wait time No

Conversation Mode

Select mode Live Chat

Live chat conversations happen in real time. When the chat ends, the session ends and the chat history is not maintained.

Work distribution

Work stream \* Live chat workstream

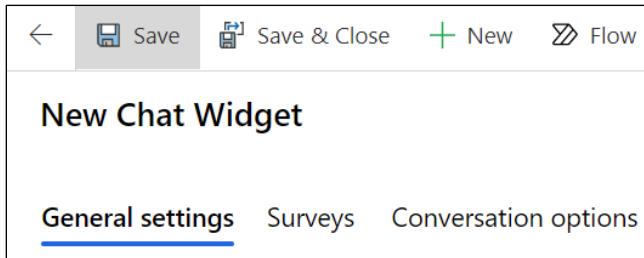
Chat Transcripts

By enabling this feature, you acknowledge that your data may flow outside your organization's compliance and geo boundaries. This includes Government Cloud environments. Learn more [here](#) and in the [Microsoft Privacy Statement](#).

Allow download of transcript \* No

Allow email of transcript \* No

4. Click Save.



- After the record is saved, a **Widget Code Snippet** will be generated. **Copy** the code snippet and store it for later use.

Patient Portal Chat Widget  
Chat Widget

General settings Automated messages Surveys Conversation options Design Location

**General information**

Name	* Patient Portal Chat Widget
Language	* English - United States
Agent display name	Full name
Authentication settings	---

**Conversation Mode**

Select mode	Live Chat
-------------	-----------

Live chat conversations happen in real time. When the chat ends, the session ends and the chat history is not maintained.

**Work distribution**

Work stream	* Live chat workstream
-------------	------------------------

**File attachments**

Enable file attachments for customers	No
Enable file attachments for agents	No

**Customer waiting**

Show position in queue	No
Show average wait time	No

**Chat Transcripts**

By enabling this feature, you acknowledge that your data may flow outside your organization's compliance and geo boundaries. This includes Government Cloud environments. Learn more [here](#) and in the [Microsoft Privacy Statement](#).

Allow download of transcript	* No
Allow email of transcript	* No

**Code snippet**

```
<script id="Microsoft_Omnichannel_LCWid" src="https://oc-cdn-ocprod.azureedge.net/livechatwidget/t/scripts/LiveChatBootstrapper.js" data-app-id="cfaa18bc-49f2-4223-a353-f4010b932ec7" data-lcw-version="prod" data-org-id="b7eccfa4-b990-4c1e-9486-5d5ec922bbf4" data-org-url="https://unpb7eccfa4b9904c1e94865d5ec922bbf4.crm.omnichannelengagementhub.com"></script>
```

**Congratulations!** In this exercise, you have successfully configured Customer Service Omnichannel Live chat by creating the necessary Users, Queues, Work Streams, Context Variables, Routing Rules, and Chat Widget. These all work together and allow patients to chat with a virtual health bot with the option to escalate up to a human agent if needed.

# 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 and a checkmark icon.

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 its 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 it, a table lists two records:

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, indicating it is selected.

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 below it. The "Value (HTML)" field contains the following code:

```
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 confirmation dialog box with the title "Chat Widget Code" and the message "Content Snippet". It includes standard save options: "Save" and "Save & Close".

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

Active Content Snippets ▾	
✓ Name ↑ ▾	Website ▾
Chat Widget Code	Customer Self-Service
✓ Chat Widget Code	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**.

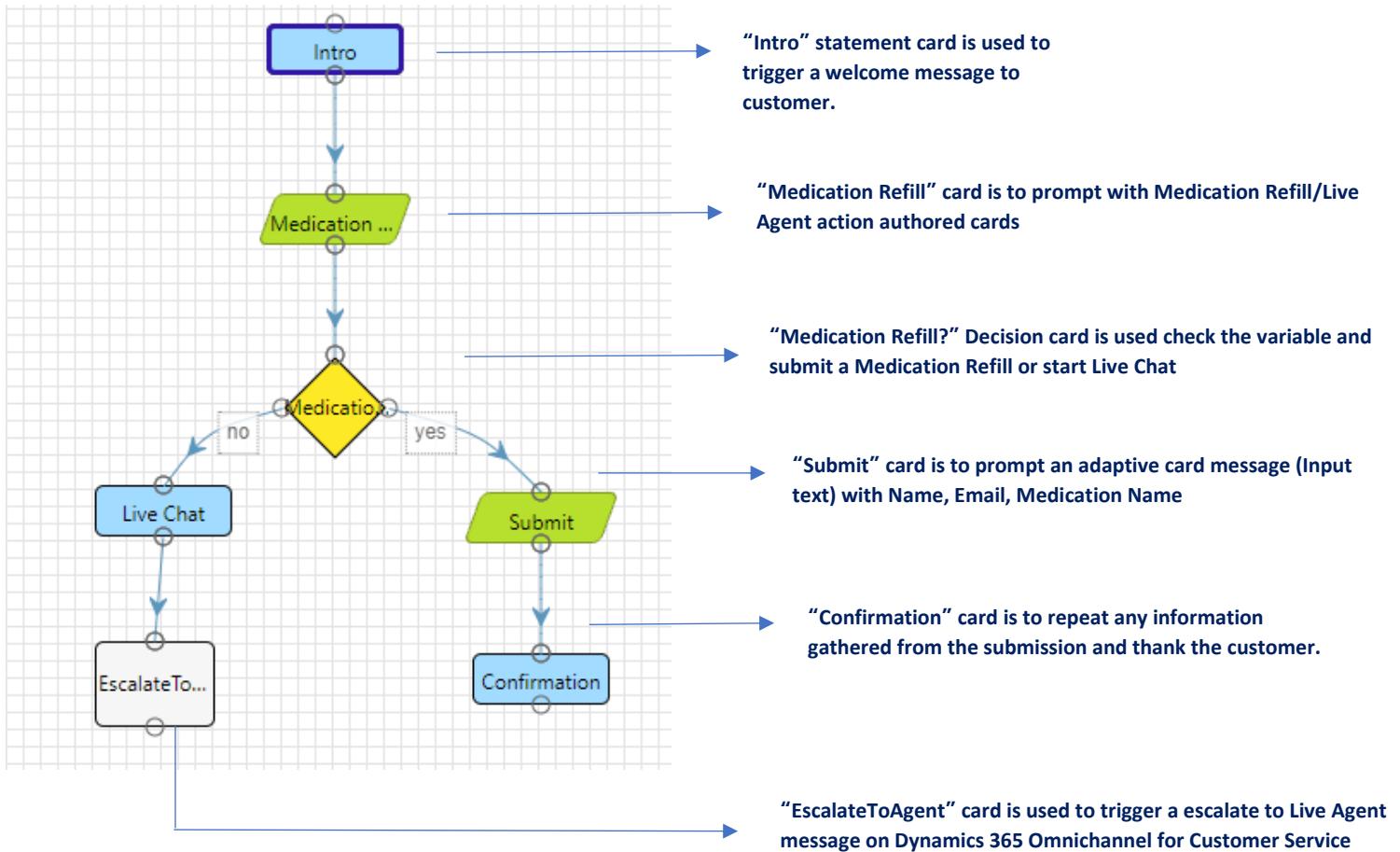
**Congratulations!** In this exercise you have successfully updated the chat widget in the Power App Portal Content Snippets. With this configuration, the Health Bot will be visible on the Power Apps portal, for both the customer self-service template and the healthcare patient portal template.

# 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**

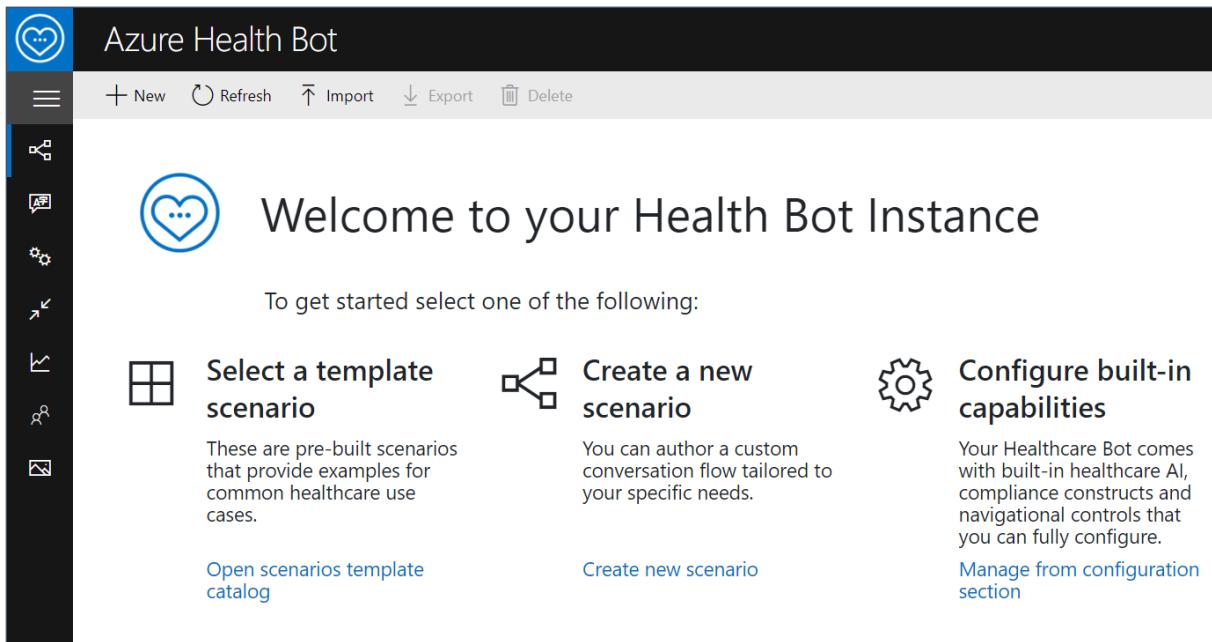


2. Design another Health Bot Scenario called **MCH\_PatientServiceWelcome**. This scenario holds the starting statement which will allow the user to invoke the **MCH\_PatientService** scenario.
3. Set the **Automatic Welcome Scenario** to be the **MCH\_PatientServiceWelcome** custom scenario you create.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

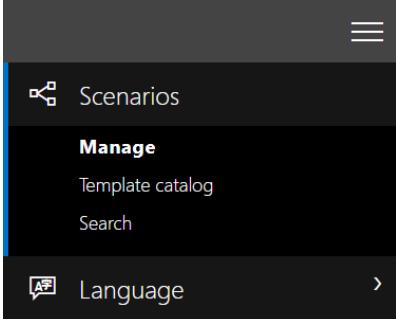
## Task 1: Create MCH\_PatientService Scenario

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

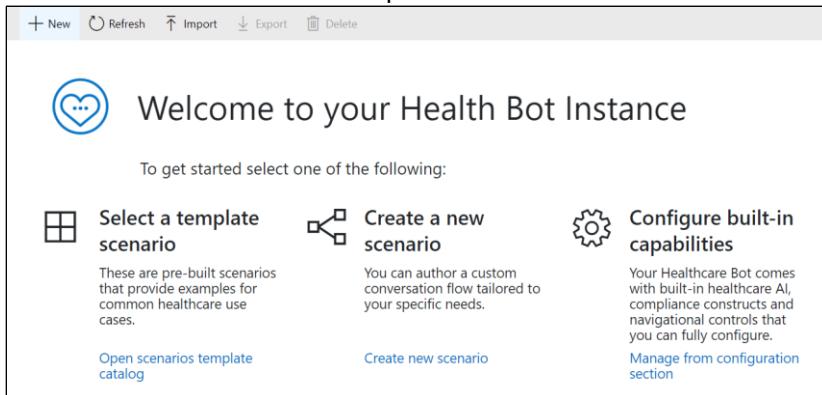
Navigate back to the Azure Health Bot instance homepage where you set the bot settings. This is the portal management link you copied from the Azure portal.



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.

Code **Designer** 2021-09-16T09:28:44.131Z + 100% ↴

**MCH\_PatientService**

Save Run Set run arguments Create Snapshot Exit

Prompt Yes/No Statement Branch Switch Begin Replace End Data Conn. Skill Global Assign Action Wait LU

Search

Type your message

Info Watch Immediate

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

## Step 1: Add Bot Introduction Statement

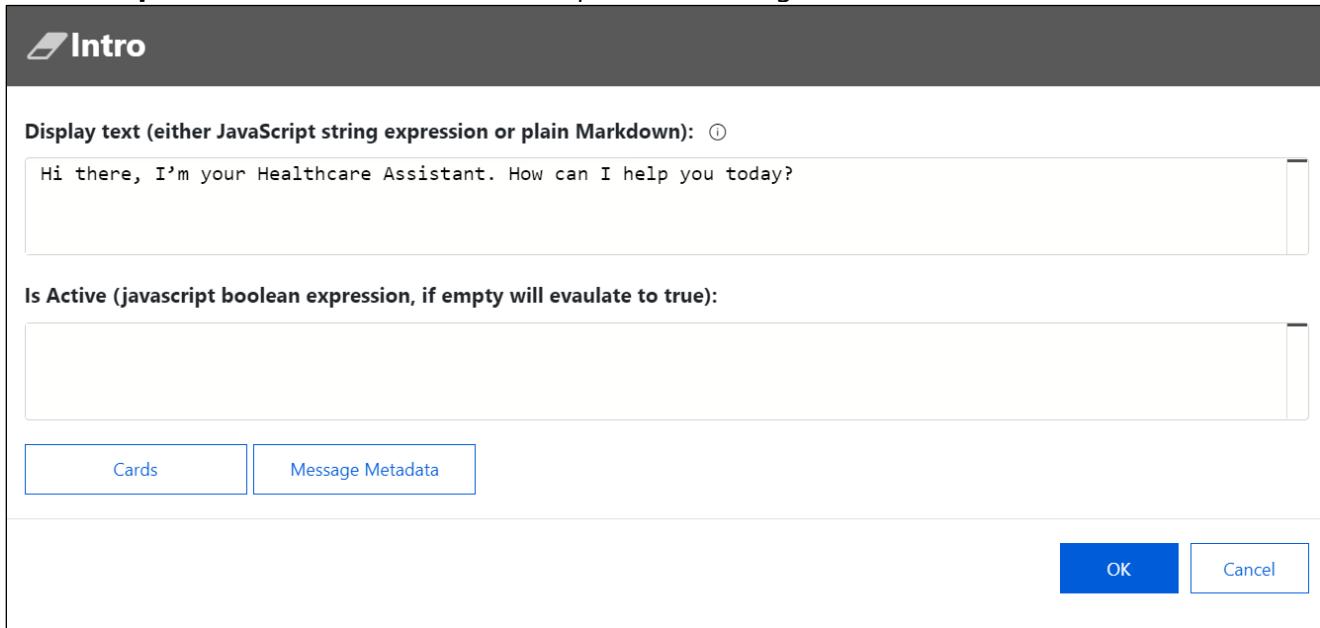
1. Add a beginning **Statement** to the scenario by **selecting** the icon **and dragging** Statement icon onto the editor.



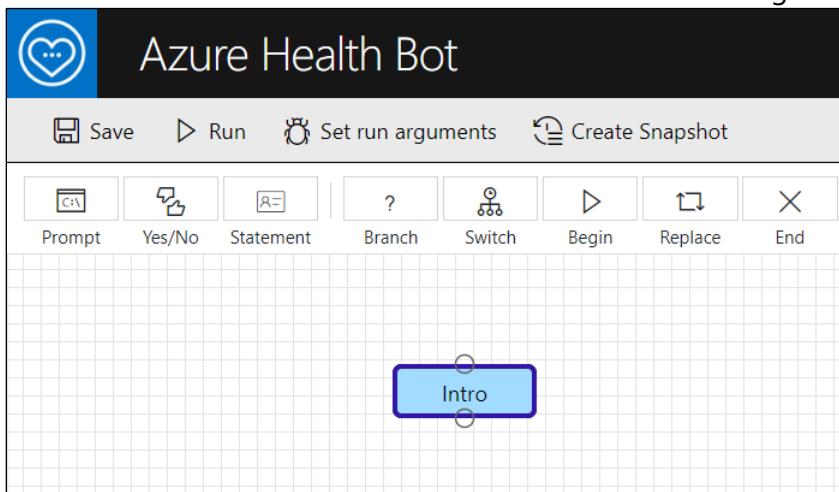
2. Enter the **Display Text**:

Hi there, I'm your Healthcare Assistant. How can I help you today?

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 **Medication Request**.

3. Click **Cards** button.

The screenshot shows the 'Medication Request' card configuration dialog. It includes fields for Display text, Variable name, Variable Data Type, Is Active, Suggestions, and buttons for Cards, Message Metadata, OK, and Cancel.

**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):** (empty field)

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

**Buttons:** Cards (highlighted in blue), Message Metadata, OK, Cancel

4. Select **Add Card**.

The screenshot shows the 'Cards' configuration dialog. It includes an 'Add Card' button and a 'Layout' dropdown menu set to 'vertical'. There are also OK and Cancel buttons at the bottom.

**Add Card**

**Layout:** vertical

**Buttons:** OK, Cancel

5. Select Card Type as **HeroCard** and fill in the title as "**Medication Refill or Live Agent**".
6. 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**

<b>Card Type:</b>	HeroCard	
<b>Image Url:</b>	<input type="text"/> 	
<b>Title:</b>	Medication Refill or Live Agent	
<b>Sub Title:</b>	<input type="text"/>	
<b>Actions</b>		
Action type	Action value	Action title
imBack	MedicationRefill	"Medication Refill"
Action type	Action value	Action title
imBack	LiveAgent	"Live Agent"
<input type="button" value="Add Action"/>		
<input type="button" value="OK"/> <input type="button" value="Cancel"/>		

7. Click Ok three times to get back to designer

**Cards**

<input type="button" value="Add Card"/>	
<b>Layout:</b>	vertical
<input type="button" value="OK"/> <input type="button" value="Cancel"/>	

## Medication Request

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):

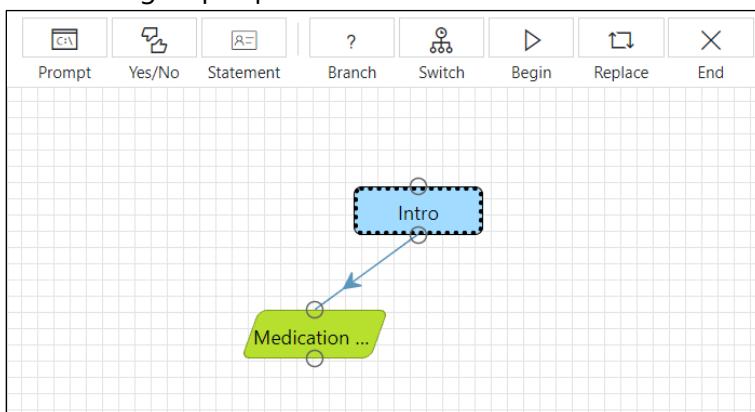
Edit Cards

Message Metadata

OK

Cancel

8. 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 Appointment boxes using ellipse pointer.



9. Select **Save**.



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

The screenshot shows the Azure Health Bot interface. At the top, there's a toolbar with icons for Save, Run, Set run arguments, and Create Snapshot. The main area is titled "Web Chat v4" and shows the following conversation:

- Bot: "Hi there, I'm your Healthcare Assistant,"
- Bot: "How can I help you today?"
- Bot: "Would you like to book an appointment with the doctor or chat with a live agent?"
- Bot: **Medication Refill or Live Agent**
- Bot: **Medication Refill**
- Bot: **Live Agent**

At the bottom left, it says "Just now".

### 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.



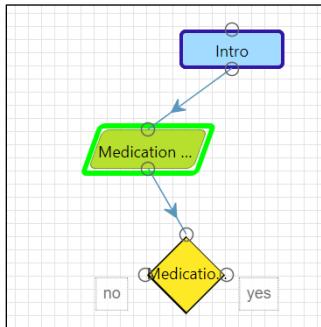
2. Enter the following in the **javascript Boolean expression**:

scenario.MedicationOrAgent === "MedicationRefill"

3. Rename to **MedicationOrAgent**. Select **OK**.

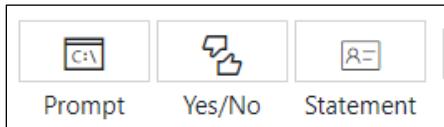


4. Drag the medication prompt to the MedicationOrAgent branch decision to connect the boxes.



#### 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:

- Display Text:** Submit
- Variable name:** formData
- Variable Data Type:** Object
- Change Title to **Submit**

**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**.

- 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:**

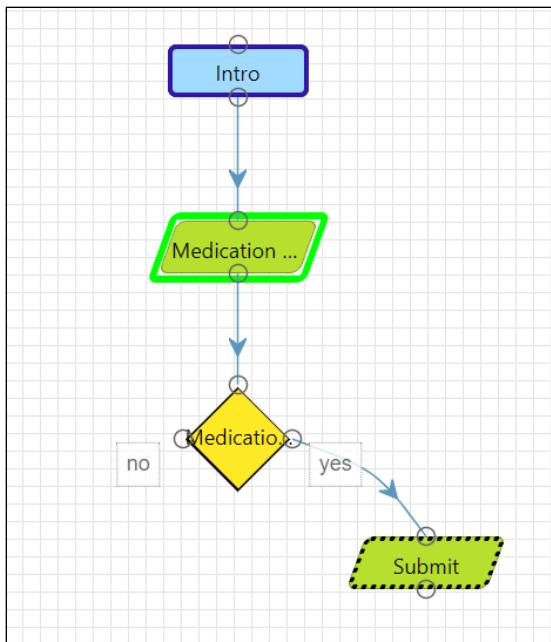
AdaptiveCard

**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": "bold",
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",
27            }
28          ]
29        }
30      ]
31    }
32  ]
33]
```

**OK** **Cancel**

- Select **OK** three times to get back to the designer.
- Connect** the **Yes** condition of the **MedicationOrAgent** 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.

Web Chat v4      en-us

**Bot**      **Submit**

**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

Your email

Medication Requested

Medication Name

**Submit**

Just now

Type your message



## Step 5: Add Confirmation Statement

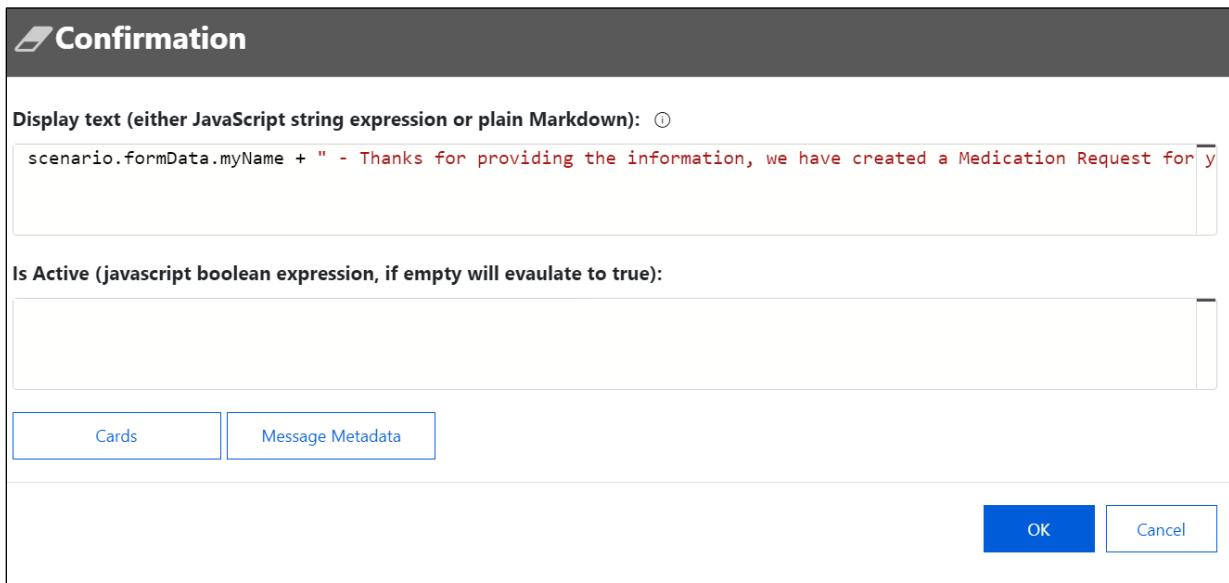
1. Add a **Statement** element.



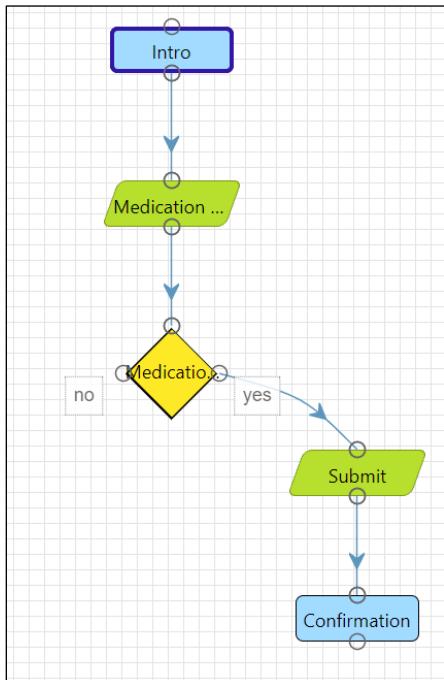
2. Add **Display text** as the following:

```
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 icon to the Confirmation icon in the designer canvas.



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

Web Chat v4 en-us

Bot Submit

**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

Your email

Medication Requested

**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

Your name

Your email

Medication Requested

**Submit**

IAD User 99 - Thanks for providing the information, we have created a Medication Request for you regarding the following medication:  
Albuterol

Just now

Type your message 



## Step 6: Invoke Live Agent Action

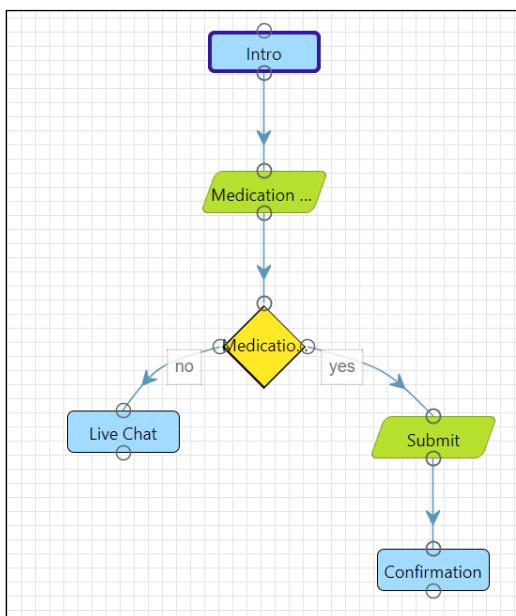
1. Add a **Statement** element to the canvas.



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**.

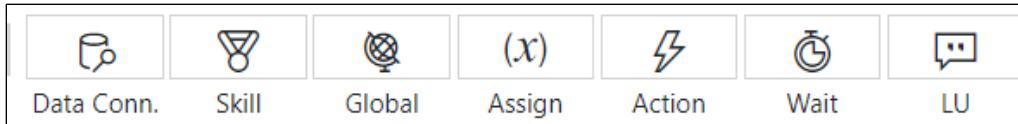


4. Click **OK** to return to the designer page.
5. **Connect** the **No** decision of **MedicationOrAgent** to the **Live Chat** statement.



## Step 7: Add Action to Invoke Escalation

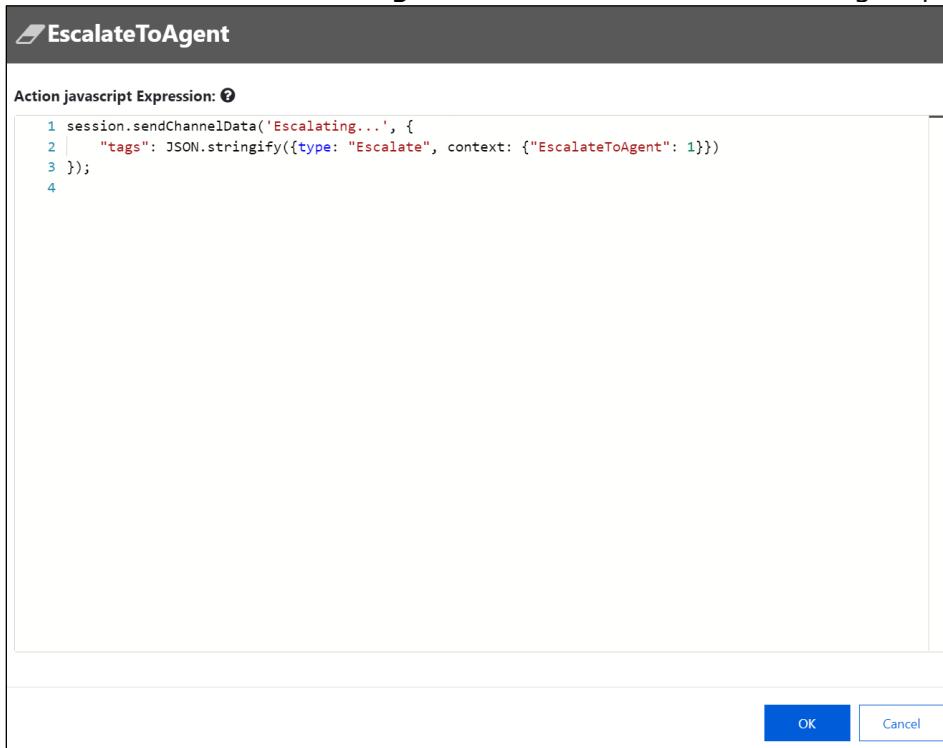
6. Add an **Action** element to the canvas, used to trigger an escalation to Omnichannel Live Agent



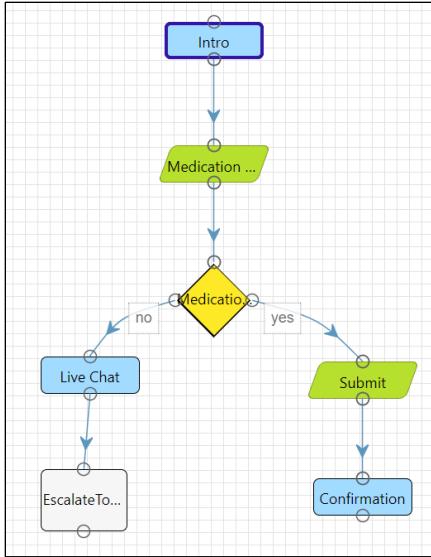
7. Add the following code in the action, which will trigger the Live agent chat:

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

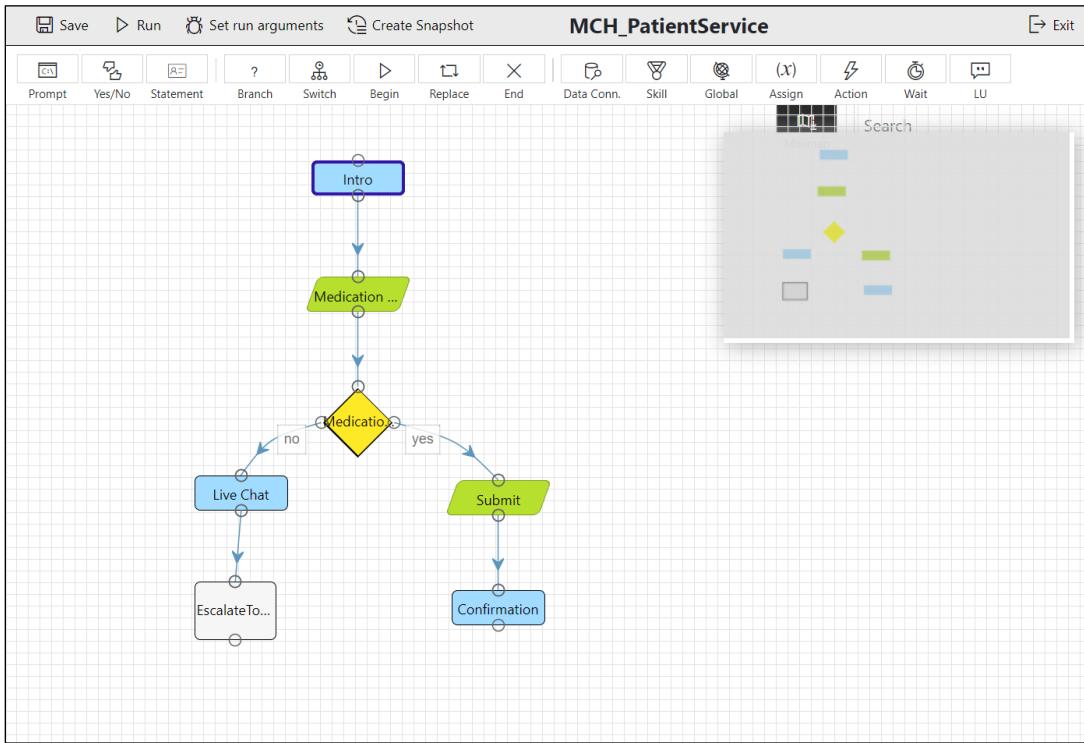
8. Name the action **EscalateToAgent**. Click **OK** to return to the designer page.



9. Connect the **Live Chat** to the new **EscalateToAgent** action.

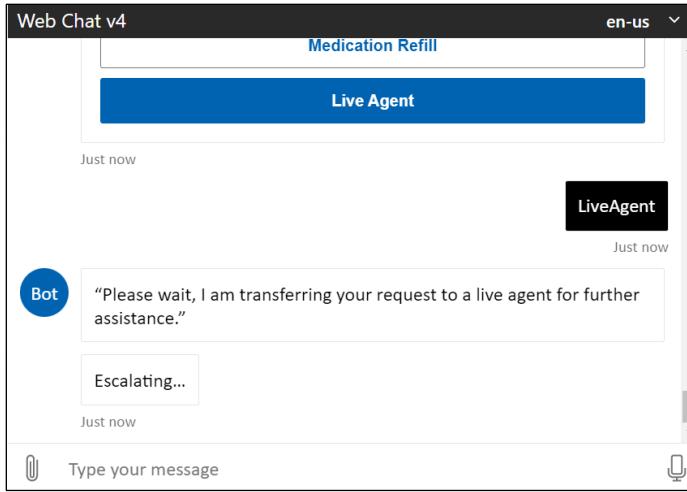


10. You have completed the final connection! Here is your full scenario:



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

12. Selecting Live Agent in the authored card should show the escalation action.



13. **Exit** the MCH\_PatientService scenario editor. Make sure you save before confirming to exit.



## Task 2: Create MCH\_PatientServiceWelcome Scenario

In this task, we will create another bot scenario called **MCH\_PatientServiceWelcome** scenario 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**: (radio button selected)
- Breaking scenario**: (radio button unselected)

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

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



4. Enter **Display text**: "Welcome to Lamna Healthcare Patient Service Portal!"

5. Rename the statement **Intro**.

The screenshot shows the 'Intro' configuration dialog. At the top, there is a title bar with the word 'Intro'. Below it, there is a section labeled 'Display text (either JavaScript string expression or plain Markdown):' with a help icon. A text input field contains the text: "Welcome to Lamna Healthcare Patient Service Portal!". Below this, there is a section labeled 'Is Active (javascript boolean expression, if empty will evaluate to true):' with a large empty text input field. At the bottom of the dialog, there are two buttons: 'Cards' and 'Message Metadata' on the left, and 'OK' and 'Cancel' on the right.

6. Select **Cards**.

The screenshot shows the 'Cards' configuration dialog. It has a similar structure to the 'Intro' dialog. The 'Display text' field contains the same welcome message. The 'Is Active' field is empty. At the bottom, the 'Cards' button is highlighted in blue, while 'Message Metadata' is in a standard light blue box. The 'OK' and 'Cancel' buttons are at the bottom right.

7. Select **Add Card**.

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

8. Choose **HeroCard**

9. 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**

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

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

**Card**

**Card Type:**  
HeroCard

**Image Url:**  
 

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

**Sub Title:**

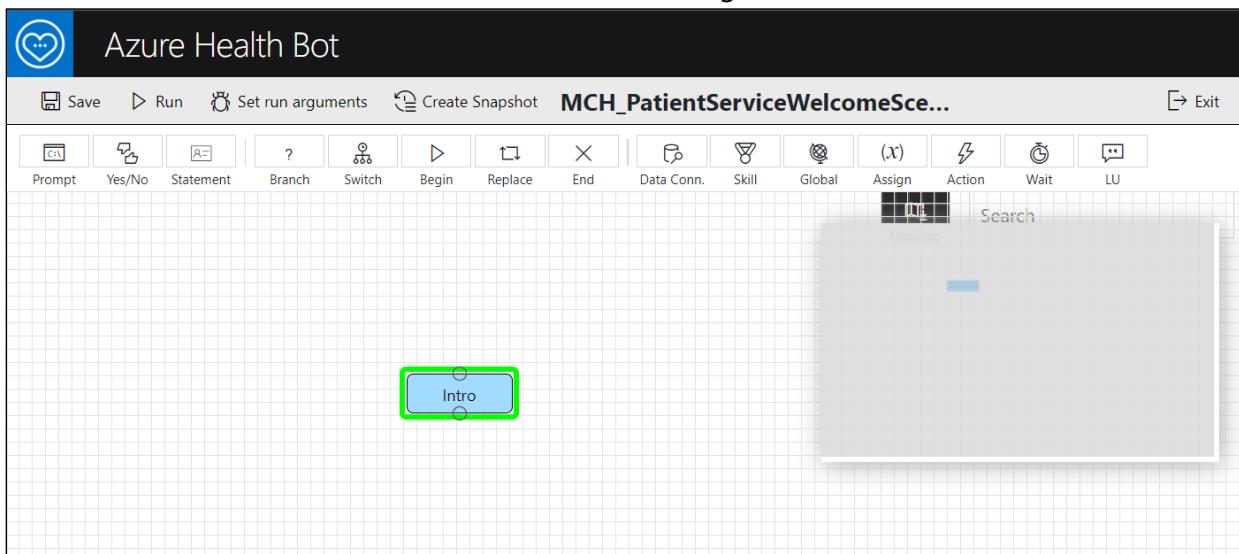
**Actions**

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

Add Action

**OK** **Cancel**

11. 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.

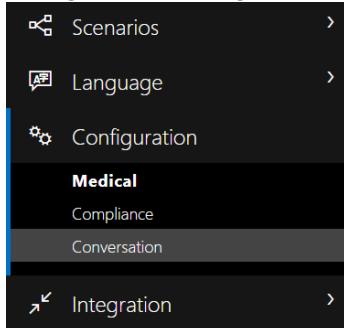


6. **Exit** the scenario designer.

## 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 scenario when a user starts conversion with **Health Bot** from Portal.

1. Navigate to Configuration > Conversation



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

The screenshot shows the 'Interactions' tab of the Azure Health Bot interface. Under the 'Automatic welcome' section, there is a note about multiple ways to display messages and a link to learn more. Below this is a text input field for the 'Automatic welcome message'. At the bottom, there is a dropdown for 'Automatic welcome scenario' with the value '\*\* scenario not selected \*\*' and a note that it works only with scenarios that display a single statement step.

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

This is a detailed view of the 'Automatic welcome scenario' dropdown menu. It shows a list of available scenarios, with 'MCH\_PatientServiceWelcome' selected. The dropdown has a placeholder 'Select scenario' and a note that it works only with scenarios that display a single statement step.

**Congratulations!** You have successfully set MCH\_PatientServiceWelcome scenario to be the default scenario that kicks off when a user interacts with the Health Bot.

## 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**.

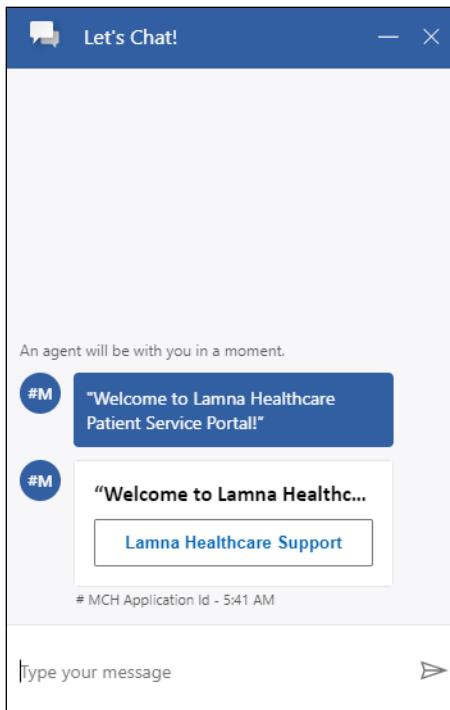
The screenshot shows the Microsoft Power Apps portal interface. On the left is a navigation sidebar with options like Home, Learn, Apps, Create, Data, Flows, Chatbots, AI Builder, and Solutions. The 'Apps' section is selected. The main area displays a list of apps under the heading 'Apps Component libraries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

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 of the page, there is a 'Forums' section. In the bottom right corner, there is a blue rectangular button with a white speech bubble icon and the text 'Let's Chat! 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**.

Apps

Apps Component libraries (preview)

⚠ 10 environment variables need to be updated. See environment variables

- ⊕ Name
- ✓ ❤️ Customer Service workspace
- ❤️ 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)

Dynamics 365 | Customer Service workspace

SANDBOX

Customer Service Agent Dash... Omnichannel Ong... Omnichannel Intra... +

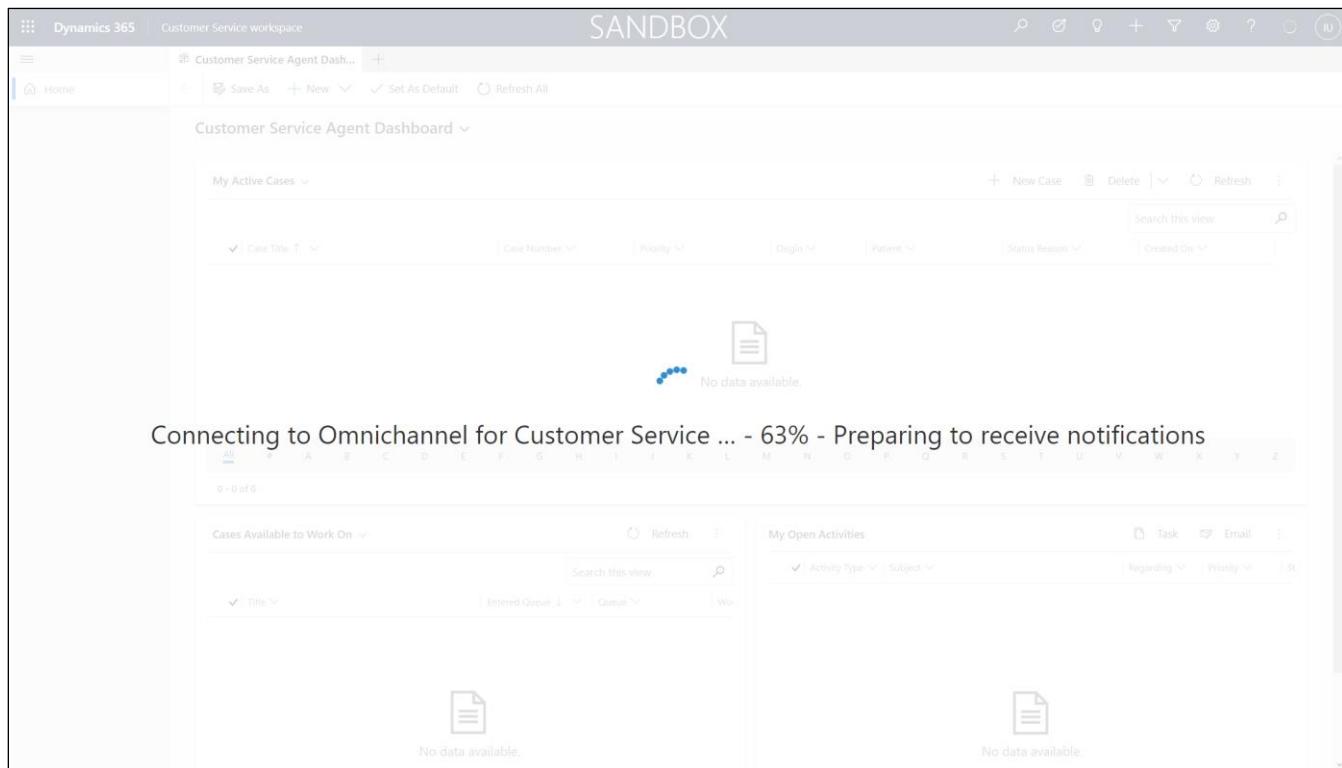
Home Save As New Set As Default Refresh All

Customer Service Agent Dashboard

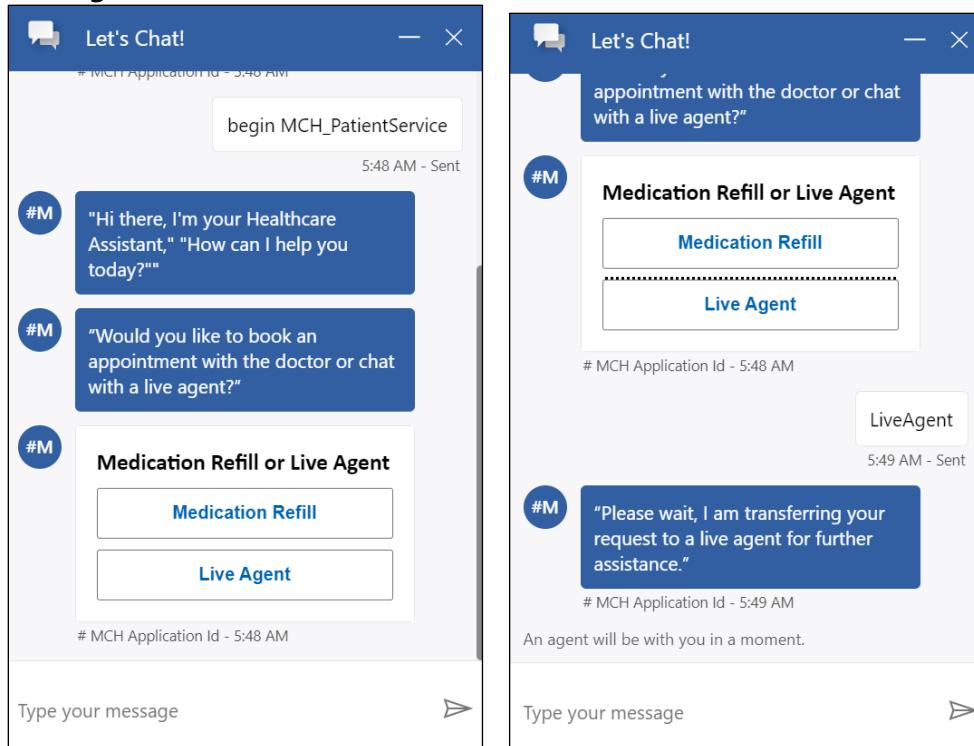
My Active Cases

Presence status

Splash screen:

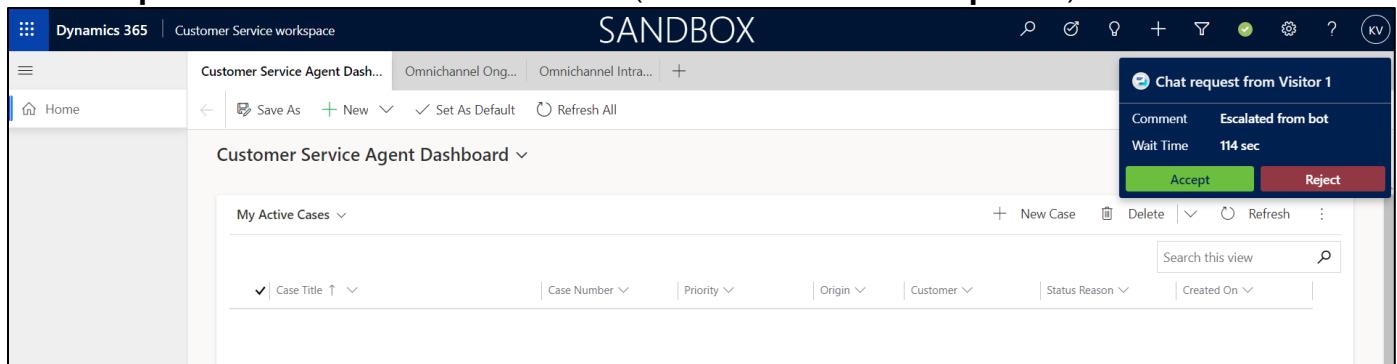


5. In the Health Bot in Lamna Healthcare Patient Portal, click **Lamna Healthcare Support** button, then the **Live Agent** button to witness the escalation into Omnichannel to chat with a live agent (your user!)

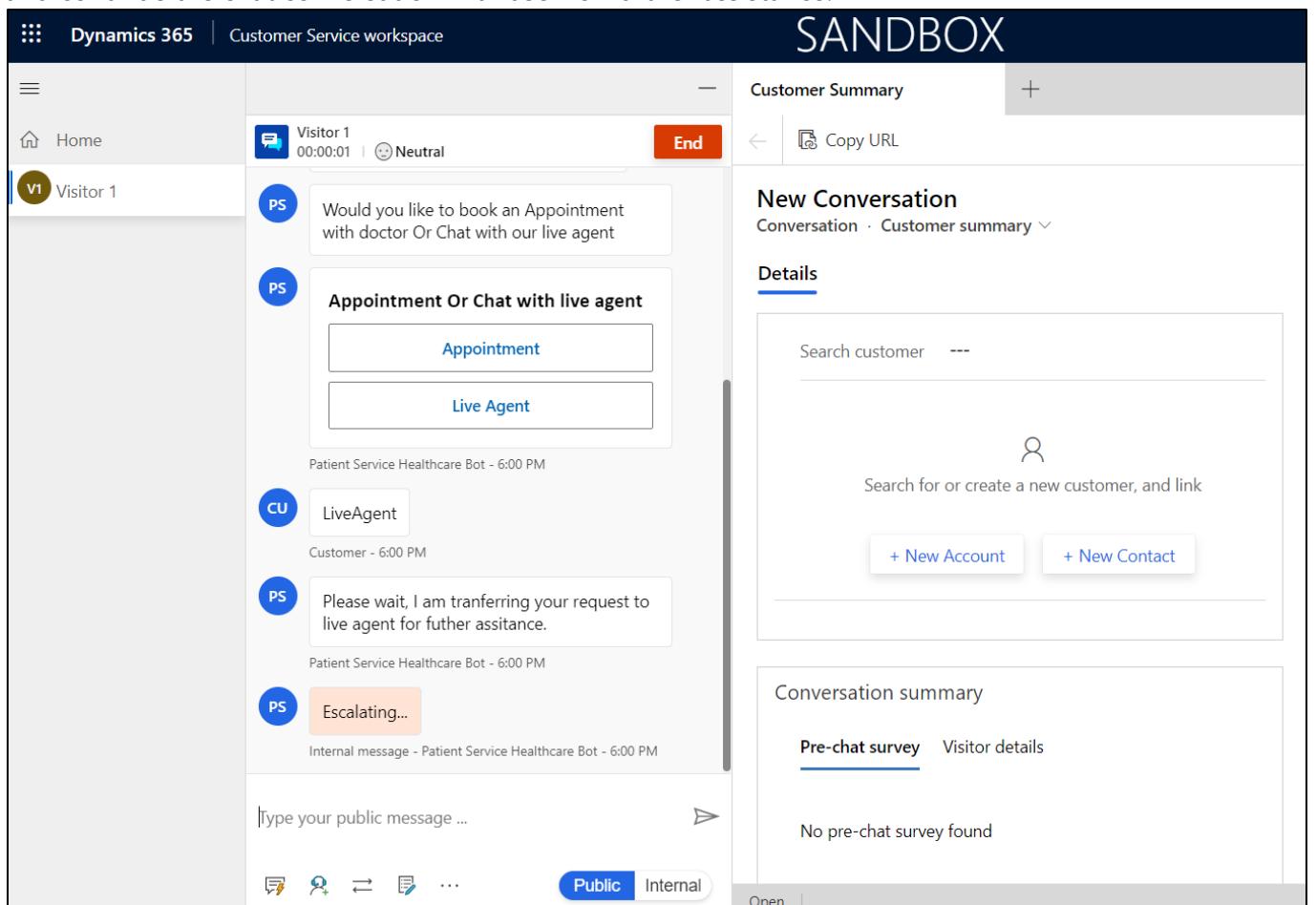


6. Navigating back to Omnichannel for Customer Service, your user as the **Live Agent** should receive an incoming notification with **Accept/Reject** options for that chat.

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 tested the end-to-end escalation scenario from the patient using a Health Bot in the Power Apps Portal to a Live Agent in Omnichannel for Customer Service.

# Summary

**Nice work!** You have completed **Lab 04 – Aure 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