



Microsoft Cloud for Healthcare in a Day

Lab 04: Azure Health Bot

Step-by-Step Lab

90 minutes

October 2022

Contents

Overview	3
Learning Objectives.....	3
Prerequisites	3
Azure Health Bot.....	3
Industry Prioritized Scenarios	4
Healthcare Story	4
Exercise 1: Set Up Azure Health Bot	5
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.....	14
Exercise 2: Configure Omnichannel Live Chat.....	16
Task 1: Assign Omnichannel Agent Security Role	16
Task 2: Create Health Bot User in Dynamics 365 Customer Service.....	20
Task 3: Create and Configure Human Agent Queue	26
Task 4: Create Workstream with Context Variables and Routing Rules.....	29
Exercise 3: Embed Health Bot in Power Apps Portal	41
Exercise 4: Extend Azure Health Bot with Custom Scenarios	44
Task 1: Create MCH_PatientService Scenario.....	45
Task 2: Create MCH_PatientServiceWelcome Scenario	65
Task 3: Configure Welcome Scenario as Automatic	69
Task 4: Test Health Bot Escalation from Power Apps Portal to Dynamics 365 Omnichannel	70
Summary	74

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.

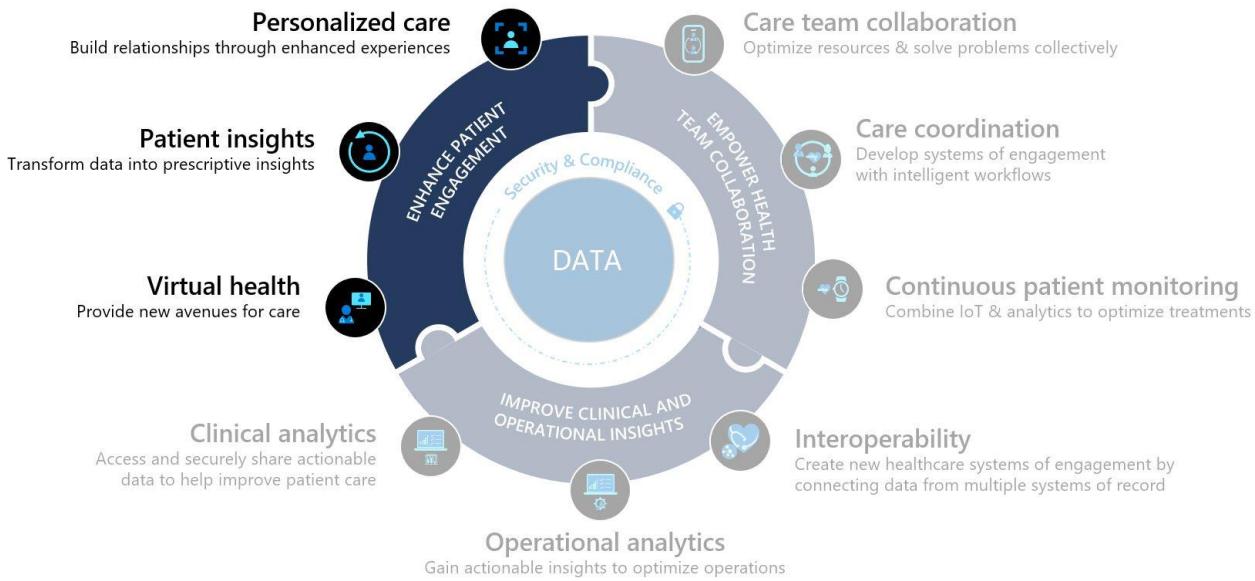
Using the service, healthcare organizations can build a "health bot instance" and integrate it with their systems that patients, nurses, doctors, and other representatives interact with. Building an instance allows you to:

- 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](#).

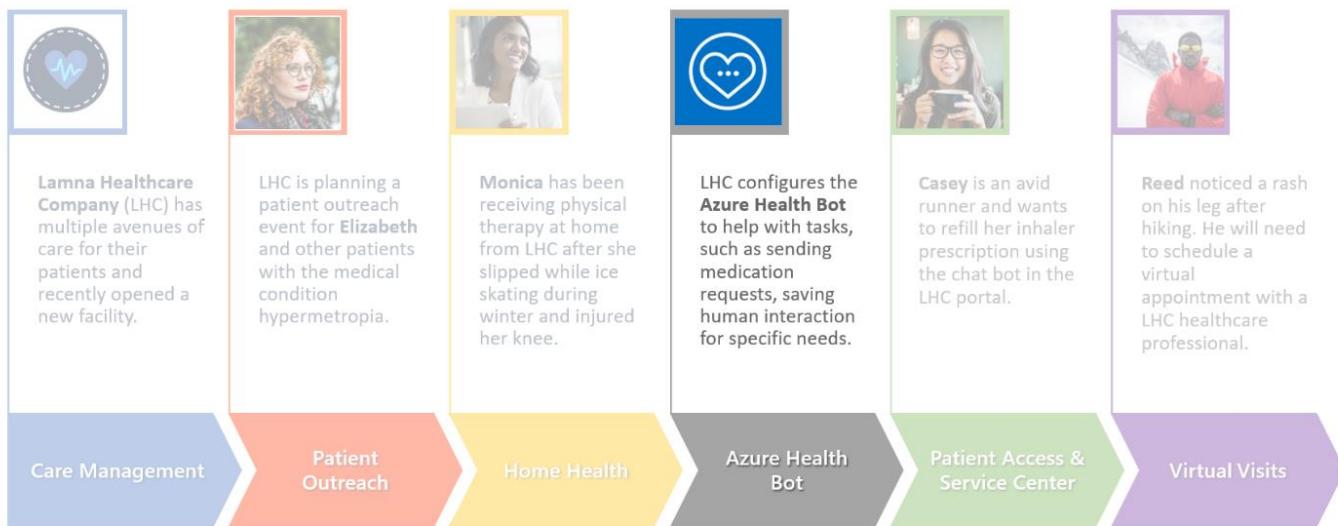
Industry Prioritized Scenarios

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

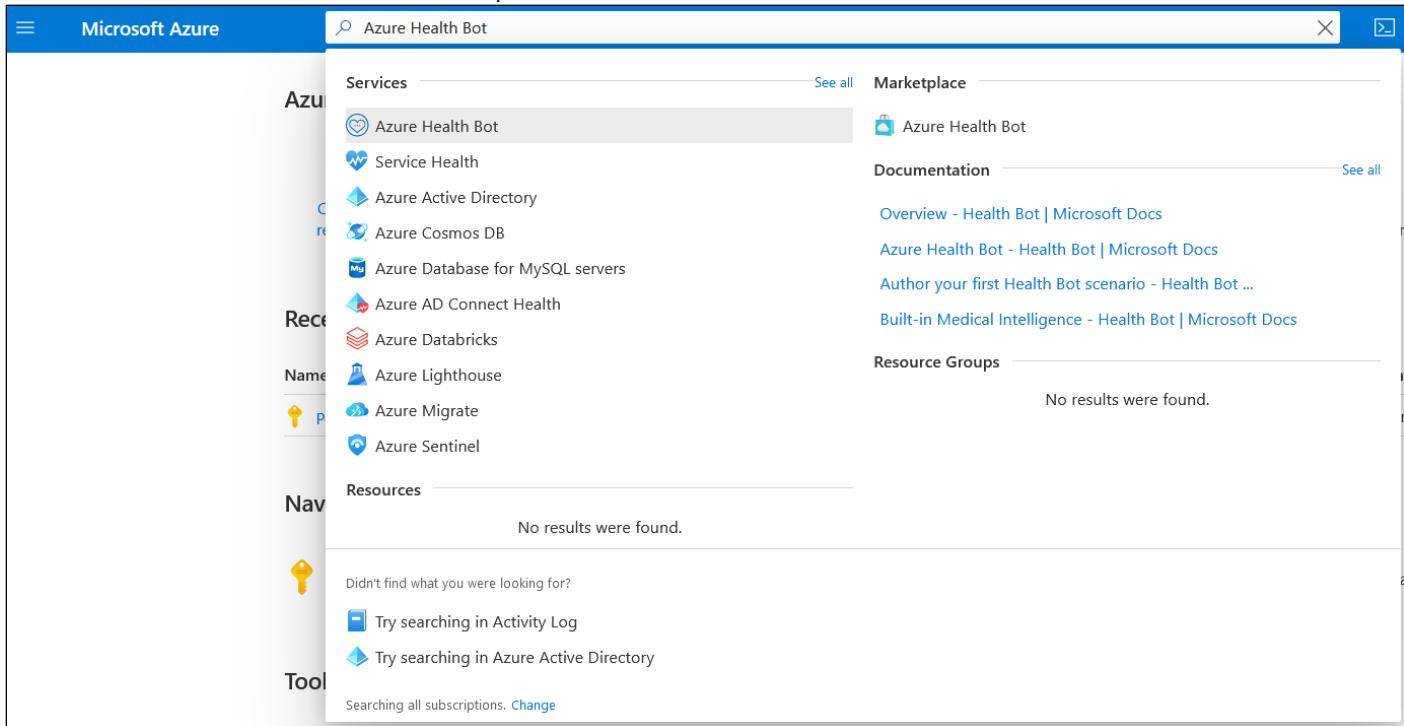
In this exercise, you will do the following:

- 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.

The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with 'Microsoft Azure' and a search bar. Below it, the page title is 'Azure Health Bot'. A red box highlights the '+ Create' button in the top-left corner of the main content area. Other buttons like 'Manage view', 'Refresh', 'Export to CSV', 'Open query', 'Assign tags', and 'Feedback' are also visible. At the bottom, there are filter options for 'Subscription', 'Resource group', and 'Location'.

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)

The screenshot shows the 'Basics' step of the Azure Health Bot creation wizard. 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 'Instance details' section, the 'Name' field contains 'iaduser99-healthbot', which has a green checkmark. The 'Region' dropdown is set to 'East US' and the 'Plan' dropdown is set to 'Free (F0)'. At the bottom, there are buttons for 'Review + create', '< Previous', and '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

Overview Inputs Outputs Template

We'd love your feedback! →

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	Operation details

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" resource group. In the "Essentials" section, there is a "Management portal" link highlighted with a red box, which points to 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 menu icon in the top-left corner of the sidebar. The sidebar also includes icons for 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, it says "To get started select one of the following:" and lists three options: "Select a template scenario", "Create a new scenario", and "Configure built-in capabilities". Each option has a brief description and a corresponding icon.

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

This screenshot is identical to the previous one, but the sidebar is now fully expanded, revealing the sitemap labels next to each icon. The labels are: "Scenarios", "Manage", "Template catalog", "Search", "Language", "Configuration", "Integration", "Analytics", "Users", and "Resources". The rest of the page, including the 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. The "Conversation" item in the list is highlighted with a red box. The other items in the list are: Scenarios, Language, Configuration, Medical, Compliance, Integration, Analytics, Users, and Resources. The rest of the page is blank, showing the standard navigation bar header and footer.

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: Displays the placeholder text '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** and **Omnichannel** channels.

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 'Channels' list interface. At the top, there's a header with the title 'Channels' and a sub-instruction: 'Make your Health bot available on multiple channels. [Learn more](#)'. Below the header is a toolbar with 'Channel list' (selected), 'Bot icon', a 'Refresh' button, and a '...' button. The main area is a table with columns: 'Active', 'Channel', and 'Actions'. There are five rows: 1. Web Chat (Active, ...), 2. DirectLine (Active, ...), 3. Microsoft Teams (Active, ...), 4. Twilio (Active, ...). The Microsoft Teams row is highlighted with a grey background.

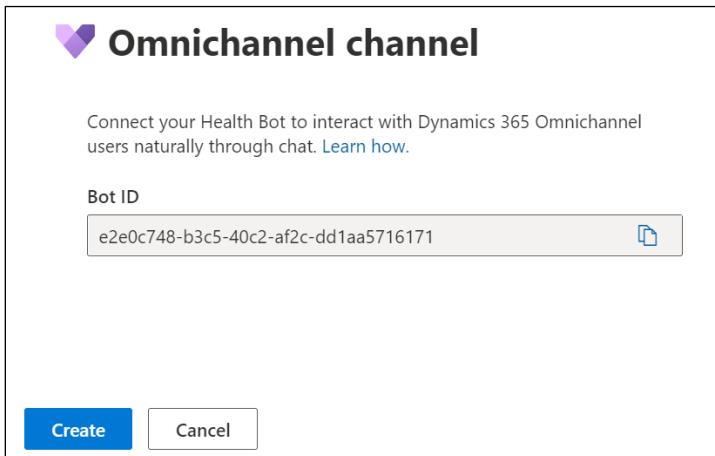
10. This will open a side window with your **Bot ID** information. **Copy and store** the Bot ID for later to use when creating the Dynamics 365 Application User. Select **Create**.

The screenshot shows a 'Microsoft Teams channel' configuration dialog. The title bar says 'Microsoft Teams channel'. The main content area has the sub-instruction: 'Connect your Health Bot to interact with Microsoft Teams users naturally through chat. [Learn how](#)'. Below this is a 'Bot ID' field containing 'e2e0c748-b3c5-40c2-af2c-dd1aa5716171' with a copy icon to its right. At the bottom are two buttons: 'Create' (highlighted in blue) and 'Cancel'.

11. Also in the channels list, select the toggle to **enable Omnichannel**.

The screenshot shows a 'Channels' list interface similar to the previous one. At the top, there's a header with the title 'Channels' and a sub-instruction: 'Make your Health bot available on multiple channels. [Learn more](#)'. Below the header is a toolbar with 'Channel list' (selected), 'Bot icon', a 'Refresh' button, and a '...' button. The main area is a table with columns: 'Active', 'Channel', and 'Actions'. There are ten rows: 1. Web Chat (Active, ...), 2. DirectLine (Active, ...), 3. Microsoft Teams (Active, ...), 4. Twilio (Active, ...), 5. Facebook (Active, ...), 6. Telegram (Active, ...), 7. Alexa (preview) (Active, ...), 8. WhatsApp - via Twilio (preview) (Active, ...), 9. Omnidchannel (Active, ...). The Omnidchannel row is highlighted with a grey background.

12. Select **Create** in the side window.



13. Both the Microsoft Teams and Omnichannel channels should now be enabled and active.

The screenshot shows the "Channels" configuration page. The title is "Channels" with the subtitle "Make your Health bot available on multiple channels. [Learn more](#)". There are two tabs: "Channel list" (selected) and "Bot icon". A "Refresh" button is present. The main area is a table with columns "Active", "Channel", and "Actions". The rows show the following channels:

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

At the bottom, a note says "Other Azure Bot Framework channels can be enabled by submitting a support request. [Contact support](#)".

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 with icons: App registrations, Event Grid Partner Registrations, App Configuration, App proxy, App Services, Function App, Application gateways, Application groups, Application Insights, and Application Services. To the right of this list, it says "No results were found." Below this list, there are links to "Documentation" such as "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 ...". There is also a "Resource Groups" section with a link to "Documentation" and a note "No results were found.". At the bottom of the search results, there are suggestions: "Didn't find what you were looking for?", "Try searching in Activity Log", and "Try searching in Azure Active Directory". A note at the very bottom says "Searching all subscriptions. Change".

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, there is a toolbar with buttons for "New registration", "Endpoints", "Troubleshooting", "Refresh", "Download", "Preview features", and "Got feedback?". Underneath the toolbar, there are 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 reads "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 'Essentials' section. The 'Application (client) ID' field is highlighted with a red box. Other fields shown include 'Display name : MCH Application Id', '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:

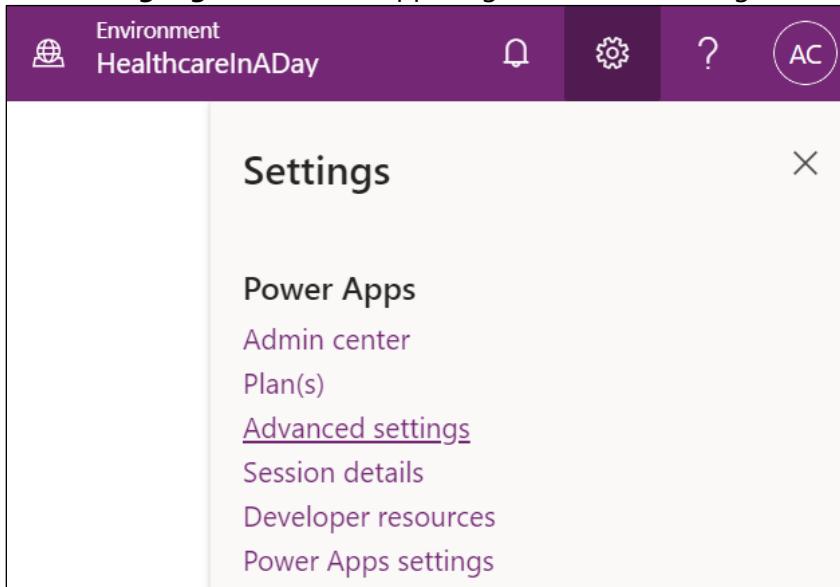
1. Assign Omnichannel Agent security role to your user
2. Create an Application User using the **MCH Application Id** and your **Bot Id**
3. Configure Queues and Workstreams for bot and agent users
4. Configure a Context Variable and Routing rule to route the message either to a Bot or Agent.

Task 1: Assign Omnichannel Agent Security Role

1. While in the In-Private or Incognito window, navigate to [Power Apps](#).
2. Ensure the **YOUR numbered** 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?

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

Business	Customization	System
 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?

 Users 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.

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

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]

Search Results ▾

<input type="checkbox"/>	Full Name ↑	Position	Main Phone	Business Unit	Site	Title	Primary Email	⋮	Charts
	IAD User 01						IADUser01@PowerPlatformOp...		

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

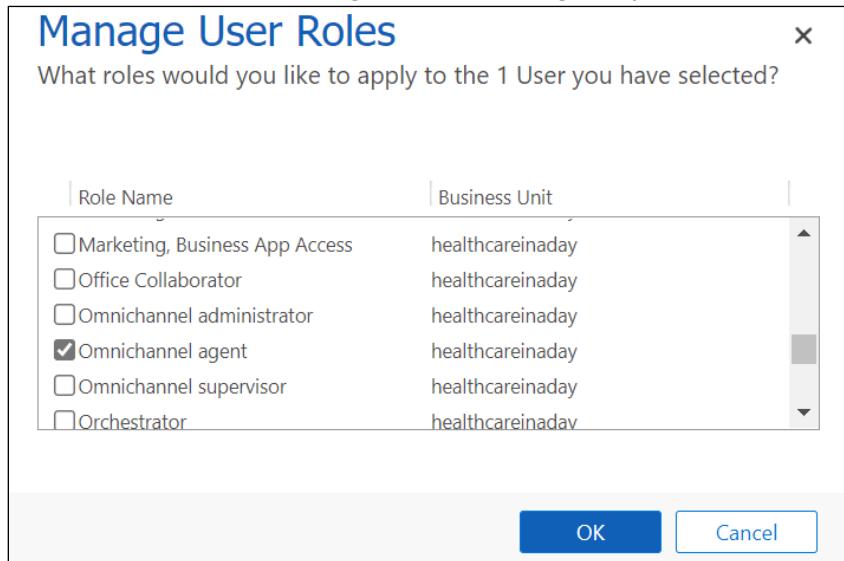
Dynamics 365 Settings ▾ Security SANDBOX

+ NEW EDIT APPROVE EMAIL REJECT EMAIL PROMOTE TO ADMIN **MANAGE ROLES** CHANGE BUSINESS UNIT

Search Results ▾

<input checked="" type="checkbox"/>	Full Name ↑	Position	Main Phone	Business Unit	Site
<input checked="" type="checkbox"/>	IAD User 01				unq0ed694338a62465...

10. Ensure the Omnichannel Agent role 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** (highlighted with a red box and a cursor icon)
 - 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
✓ MCH Application Id	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

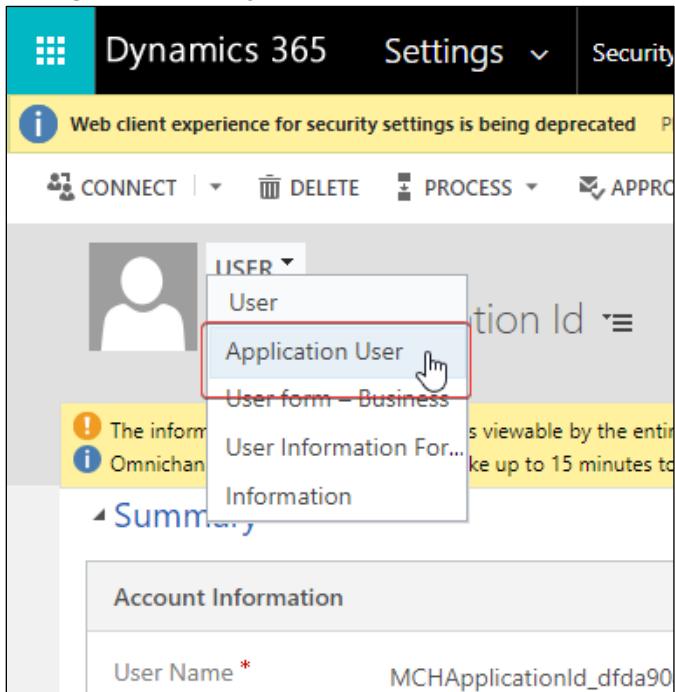
- @Me
- Access Mode Interactive Users
- Administrative Access Users
- Administrators
- Agents
- All
- Application Users
- Associated Record Team Members
- Bot agents
- Bot Users
- By Me
- Disabled Users
- Disabled users consuming licenses
- Enabled Users**
- Full Access Users

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						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.

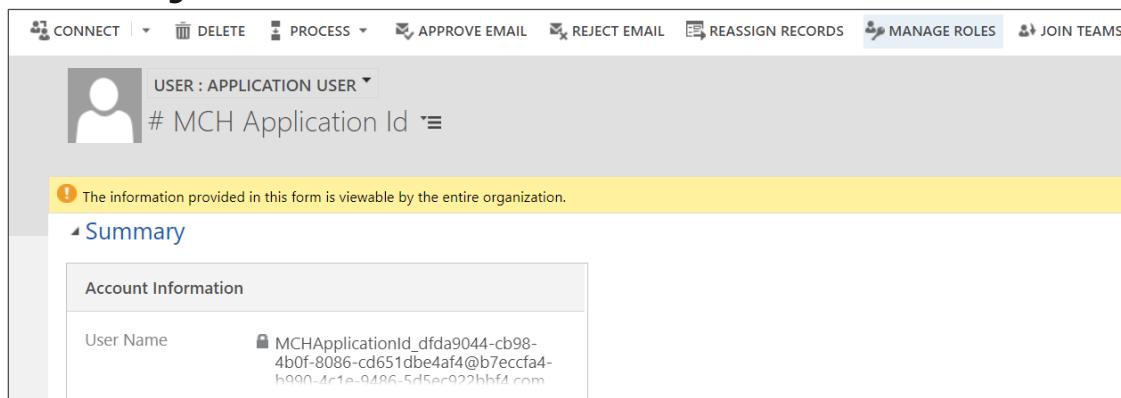
The screenshot shows the 'USER : APPLICATION USER' form. At the top, it displays '# MCH Application Id'. A yellow status bar at the top left says 'The information provided in this form is viewable by the entire organization.' The form is divided into sections: 'Summary' (which is collapsed), 'Account Information', and 'User Information'. The 'Account Information' section contains fields for 'User Name' (locked, value 'MCHApplicationId_dfda...'), 'Application ID' (locked, value blurred), 'Application ID URI' (locked, value blurred), and 'Azure AD Object ID' (locked, value blurred). The 'User Information' section contains fields for 'Full Name' (locked, value '# MCH Application Id'), 'Primary Email' (locked, value 'MCHApplicationId_dfda...'), and 'User type' (value '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.

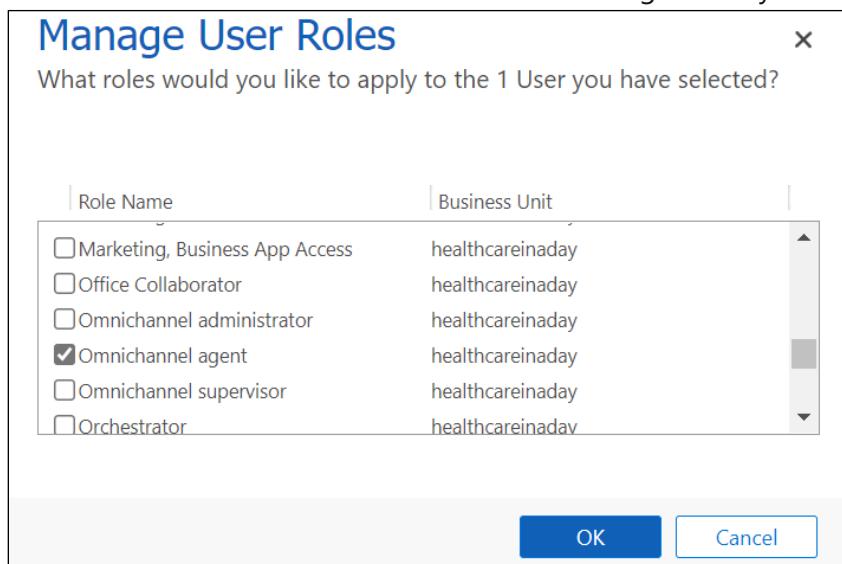
The screenshot shows the 'User Information' form. The 'User type' dropdown is open, and 'Bot application user' is selected. Below it, the 'Bot application ID' field contains the value 'e6e854a0-07bb-4329-8504-bf41912d519d'. Other fields like 'Full Name' and 'Primary Email' also have values entered.

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



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.



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 Microsoft Power Apps portal. 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 of apps. One app, 'Omnichannel admin center', is listed with a purple heart icon. A red banner at the top of the list says '7 environment variables need to be updated. See environment variables'. The table columns are 'Name' and 'Modified'.

Name	Modified
Omnichannel admin center	2 wk ago

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 sections for 'Live demos', 'Workstreams', and 'Queues'. In the 'Queues' section, there's a blue circular icon with three people and a speech bubble. To the right of the icon, there's a 'Create a queue' button, which is highlighted with a red box.

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](#)

Users		Search	See more	Add 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.

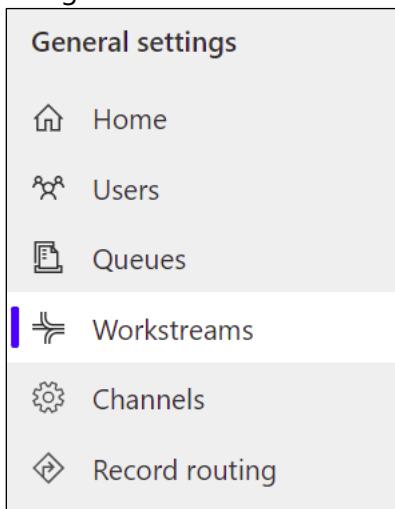
Task 4: Create Workstream with Context Variables and Routing Rules

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](#).

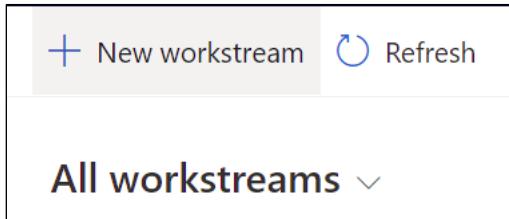
In this task, we will set up basic chat routing with a new workstream. This will allow for users to chat with a bot user initially and route to a live human agent in the proper situation. We will complete the following:

- **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>Live chat <small>Required</small></p> <p>Set up your chat channel</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. Learn more</p> <p style="text-align: center;">Set up chat</p>													
<p>Routing rules Learn more</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. Learn more + Create ruleset</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. Learn more + Create ruleset</p>													
<p>Work distribution</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;"><small>Work distribution mode</small> Push</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> Push	<small>See more</small>	Capacity	Unit based 30	Block capacity for wrap up	Always block	Allowed presences	Available Busy	Default skill matching	None	<p>Bot <small>Optional</small></p> <p>Add a bot</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>Learn more + Add bot</p>	
<small>Work distribution mode</small> Push	<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
Block capacity for wrap up: Always block
Allowed presences: Available
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](#)

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 contains a 'Context variables' section. On the right side of this section is a blue '+ Add' button. Below this section are two columns: 'Name' and 'Type'.

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

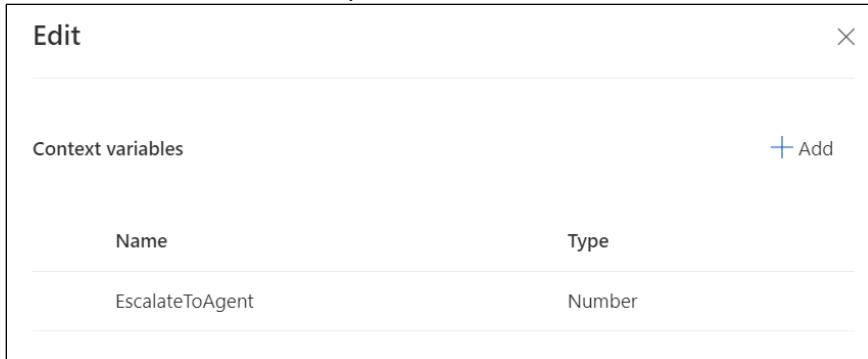
- Name:** EscalateToAgent
- Type:** Number
- Click **Create****

The 'Add context variable' dialog box has the following fields:

- Name ***: EscalateToAgent
- Type ***: Number

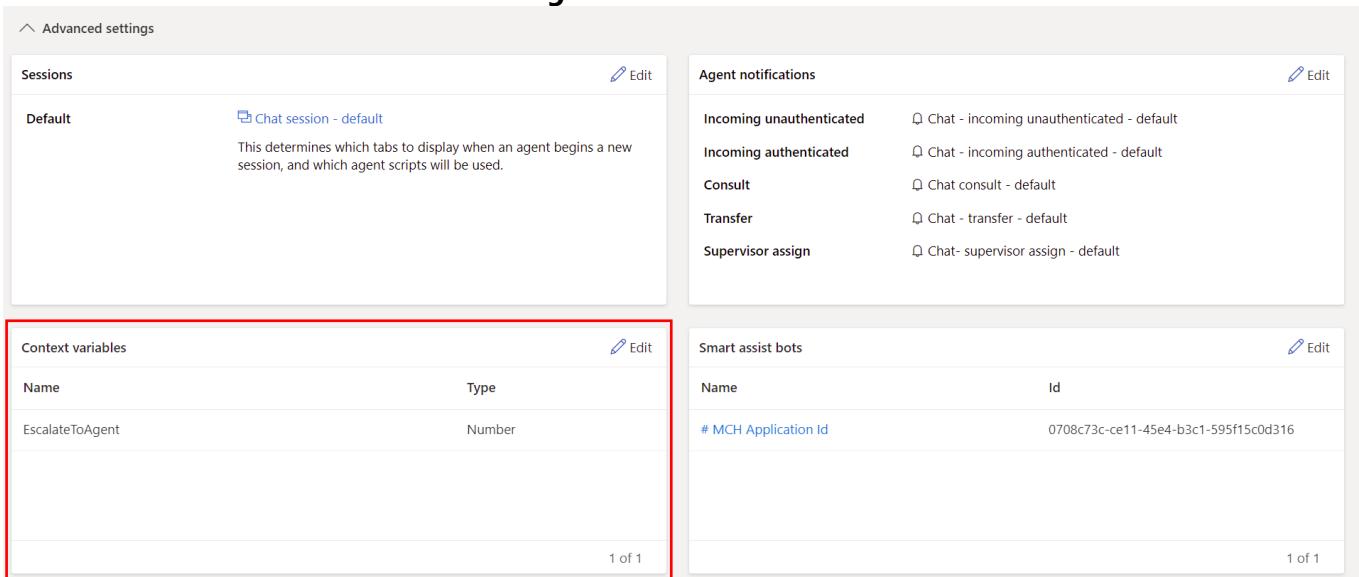
At the bottom are two buttons: a blue 'Create' button and a white 'Cancel' button.

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.



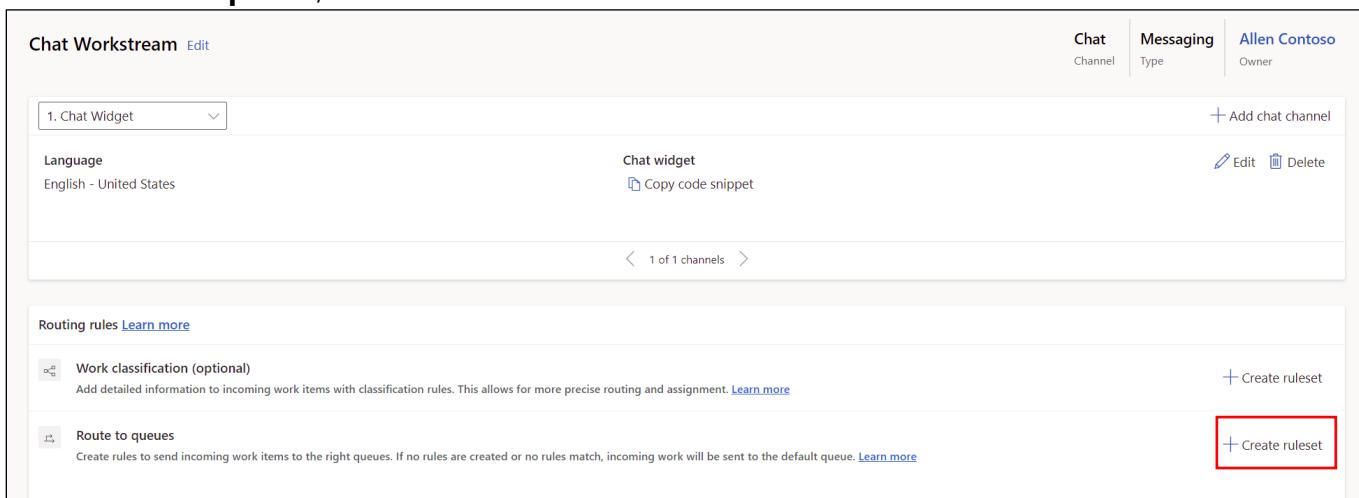
The screenshot shows the 'Advanced settings' page for a 'Chat Workstream'. It includes sections for 'Sessions' (with a 'Default' tab selected), 'Agent notifications', 'Context variables' (which contains the 'EscalateToAgent' entry), and 'Smart assist bots'. A red box highlights the 'Context variables'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

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



A 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 X

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

Name *

Description

[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 *

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 features a complex logical structure with multiple levels of nesting. The top level is an 'And' condition. Inside it is another 'And' condition with two dropdowns: 'Context item value' and 'Contains data'. Within this, there is a third 'And' condition containing a single dropdown for 'EscalateToAgent' set to 'Equals 1'. There are also '+ Add' buttons for further expansion.

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 contains a dropdown menu with 'Escalate to Human' selected, indicated by a red asterisk.

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
← Human Agent Edit			+ Create rule Refresh Search	
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. Learn more				
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 checkbox icon and a 'Name' column containing 'Lamna Healthcare Patient Portal'. The second row has a checked checkbox icon and a 'Name' column containing 'Portal Management'.

	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 rows of data are visible:

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 row, which contains the "Chat Widget Code" record for "Customer Self-Service", is highlighted with a blue background. The other row is visible 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 Chat Widget Code 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 ▾	
✓ 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	* Healthcare Patient Portal
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**.

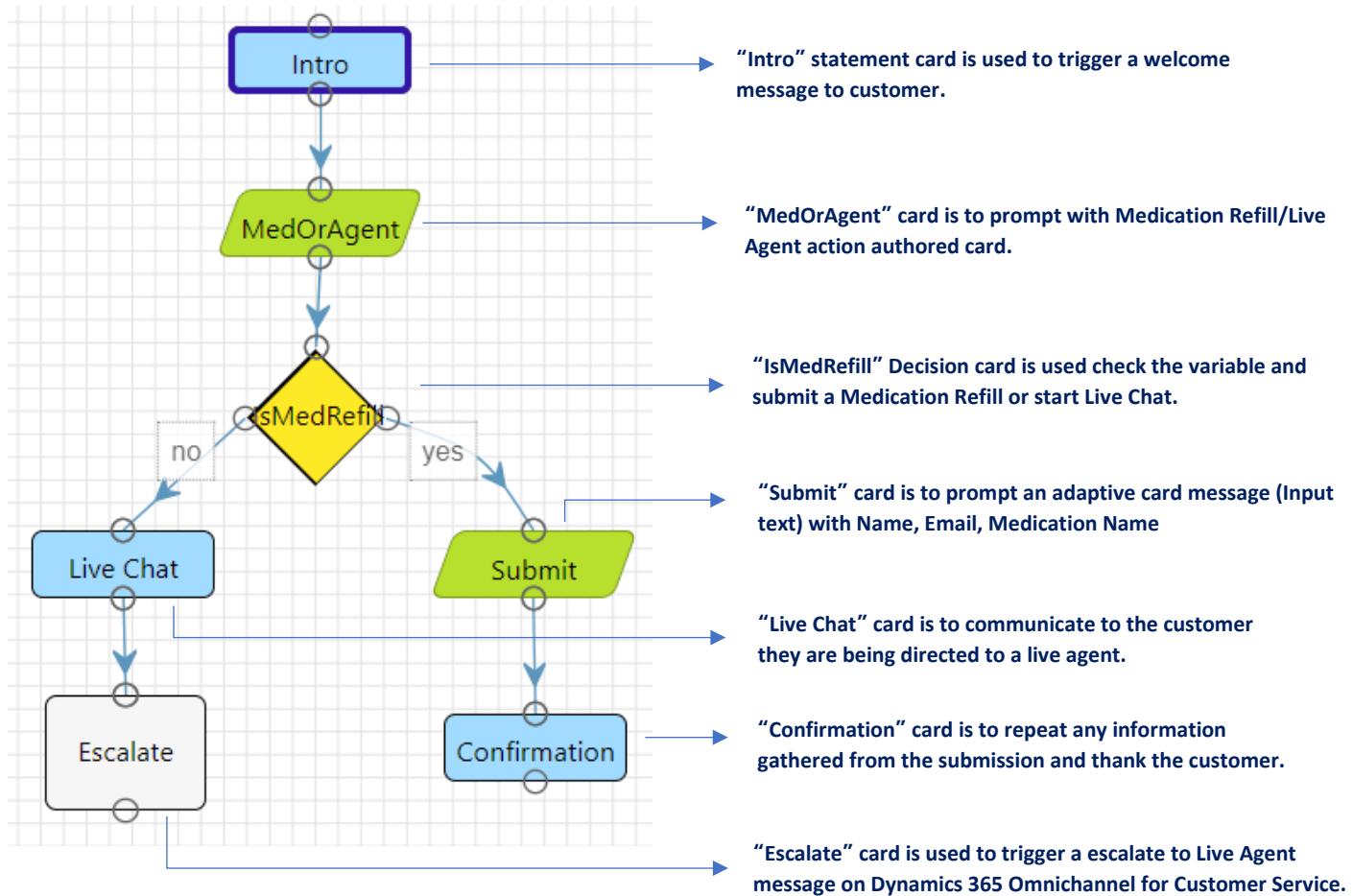
Congratulations! In this exercise you have successfully updated the chat widget in the Power App Portal Content Snippets. With this configuration, the Health Bot will be visible on the Power Apps portal for both the customer self-service template and the healthcare patient portal template.

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**

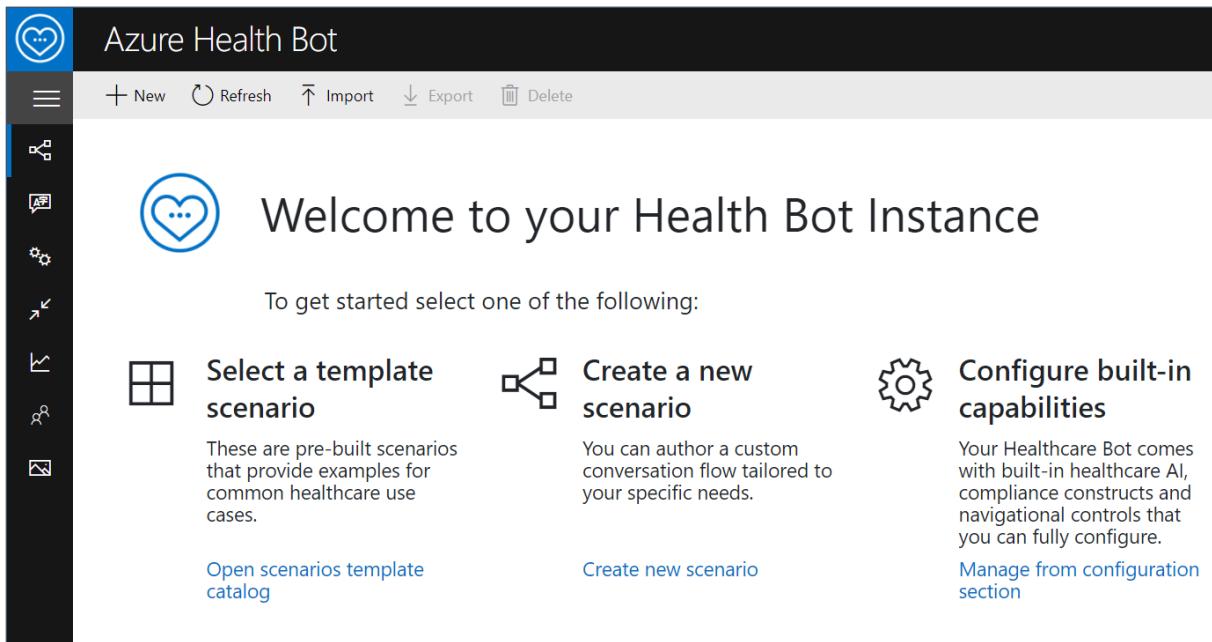


2. Design another Health Bot Scenario called **MCH_PatientServiceWelcome**. This scenario holds the starting statement which will allow the user to invoke the **MCH_PatientService** scenario.
3. Set the **Automatic Welcome Scenario** to be the **MCH_PatientServiceWelcome** custom scenario you create.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

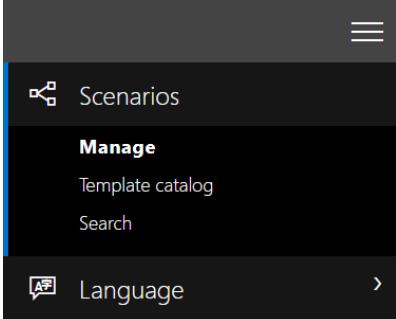
Task 1: Create MCH_PatientService Scenario

In this task, you will create the **MCH_PatientService** bot scenario using the designer canvas.

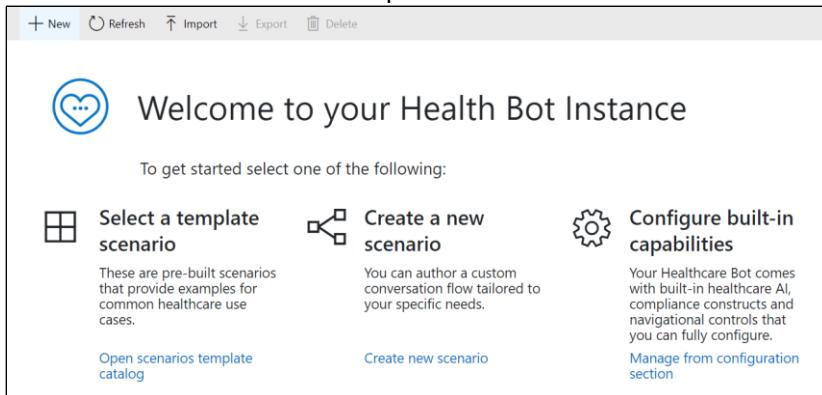
Navigate back to the Azure Health Bot instance homepage where you set the bot settings. This is the portal management link you copied from the Azure portal.



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 and click **Create**:

- 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.

Azure Health Bot - iaduser73-healthbot

Scenarios **MCH_PatientService** Active + Add element Save Discard Run Set run arguments History Web Chat English (United States)

Start authoring this scenario
Add your first step now. Read more about [Authoring custom scenarios](#)

Scenarios Language Configuration Integration Analytics Users Resources

Step 1: Add Bot Introduction Statement

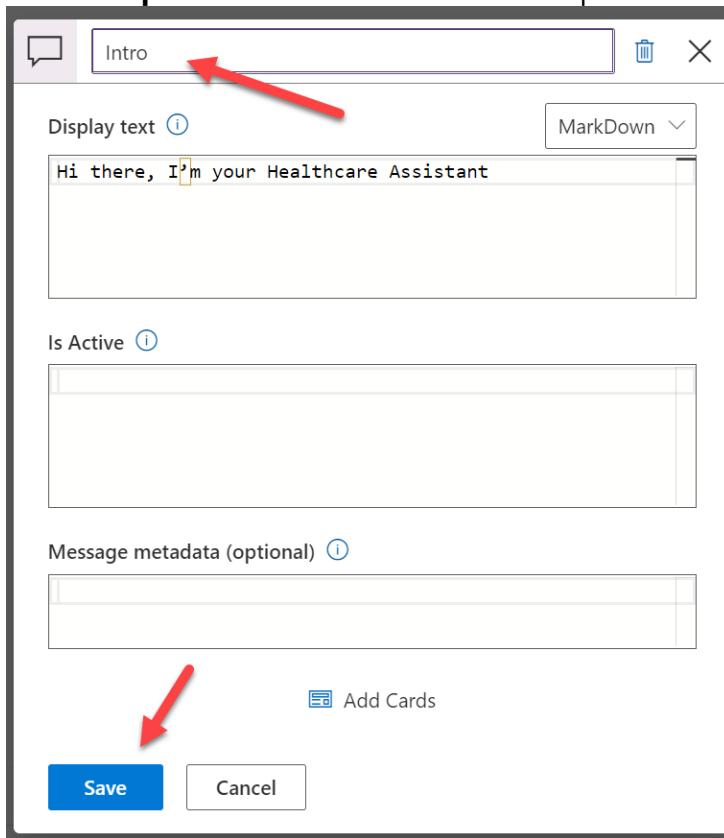
1. Add a beginning **Statement** to the scenario by **selecting** the **+Add element** menu item and select **Statement** onto the editor.

The screenshot shows the Microsoft Power Automate designer interface for the 'MCH_PatientService'. At the top, there's a header with the service name, an 'Active' toggle switch, and several action buttons: 'Add element', 'Save', 'Discard', 'Run', 'Set run arguments', and 'History'. Below the header is a search bar with a magnifying glass icon. Underneath the search bar is a dropdown menu with three main categories: 'Conversational elements', 'Flow control elements', and 'Advanced control elements'. The 'Statement' option under 'Conversational elements' is highlighted with a red arrow.

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**".



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

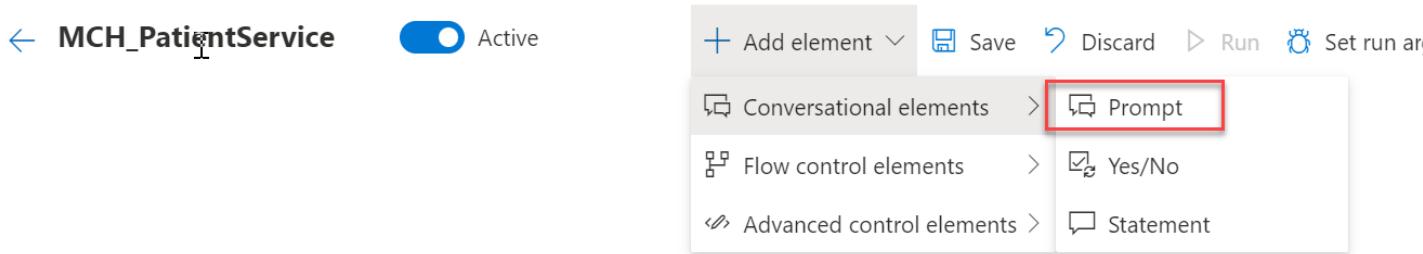
The screenshot shows the designer canvas with a single element card labeled 'Intro'. A red arrow points to this card.



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 to add an element and choose **Prompt**



2. Enter the following details:

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

The screenshot shows the configuration dialog for the 'MedOrAgent' card. The card title is 'MedOrAgent'. The 'Display text (optional)' field contains 'request a medication refill or chat with a live agent?'. The 'Input variable *' field is set to 'MedicationOrAgent'. The 'Input type' dropdown is set to 'String'. The 'Suggestions (javascript string array expression)' field is empty. The 'Is active' field is empty. The 'Message metadata (optional)' field is empty. At the bottom, there are 'Save' and 'Cancel' buttons, and a 'Add Cards' button.

MedOrAgent

Display text (optional) MarkDown

Input variable *

MedicationOrAgent

Input type

String

Suggestions (javascript string array expression)

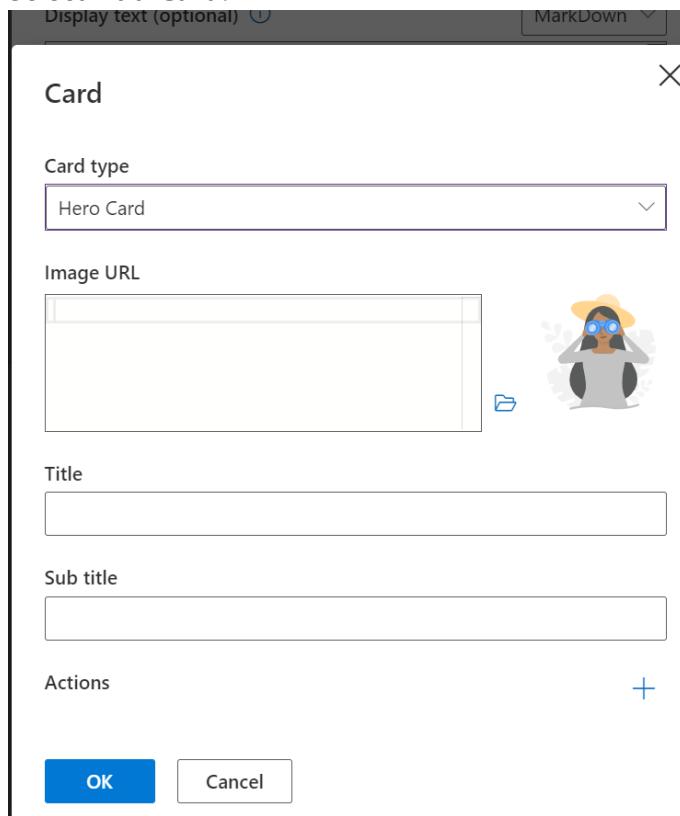
Is active

Message metadata (optional)

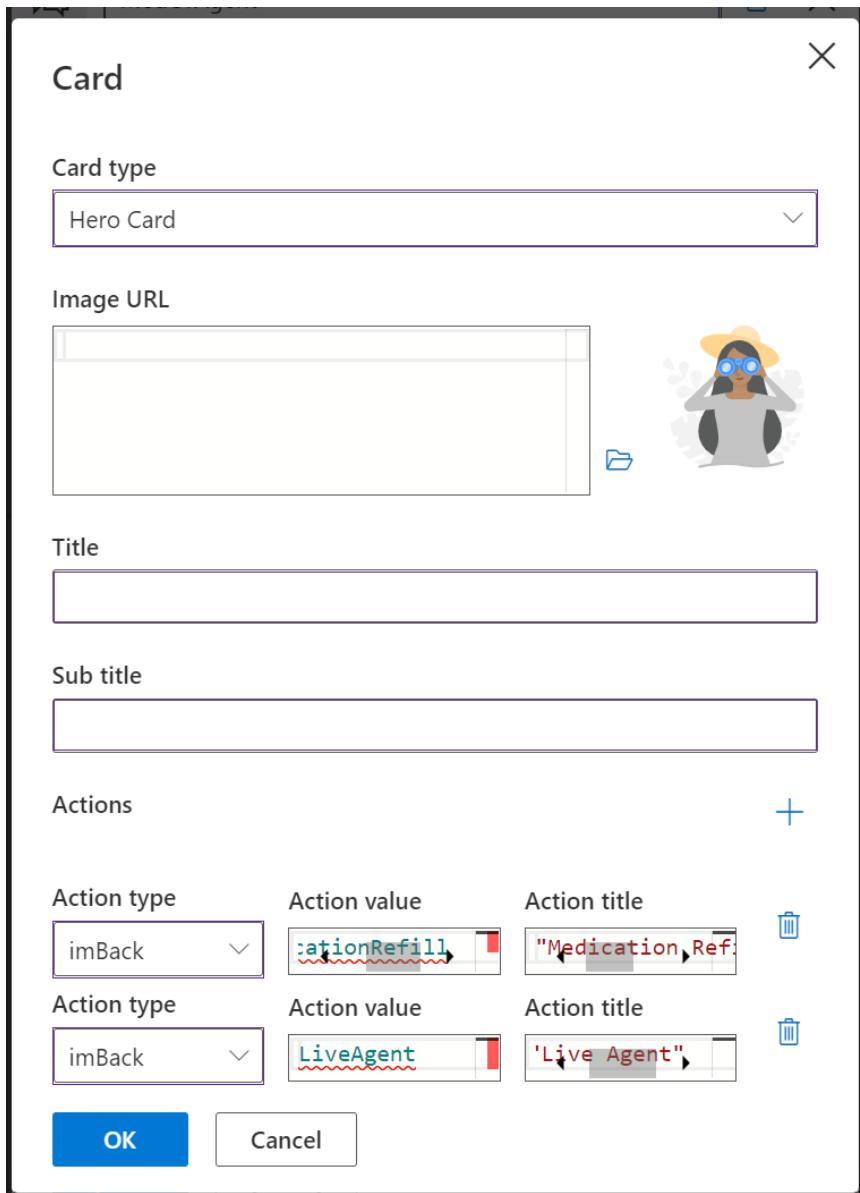
Add Cards

Save Cancel

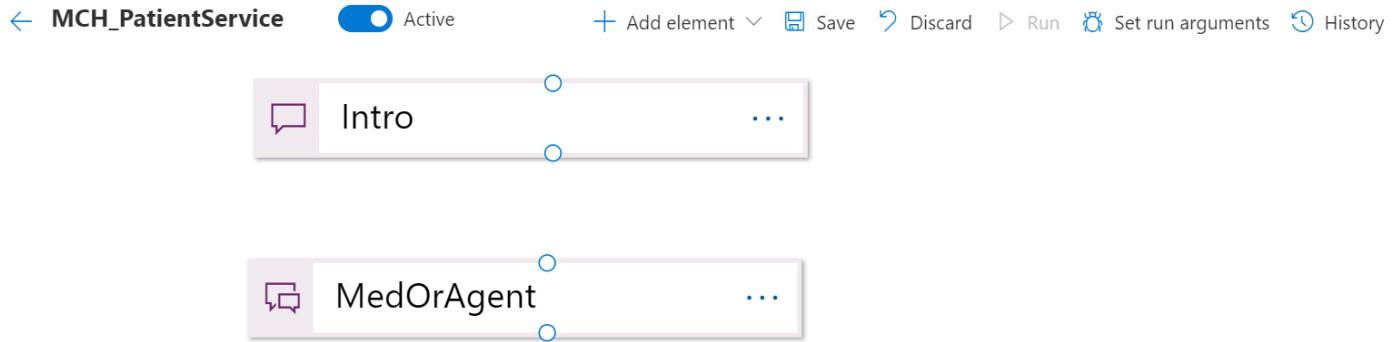
3. Select **Add Card**.



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:
- For the first action, select the following:
 - Action type: imBack
 - Action value: MedicationRefill
 - Action title: "Medication Refill"
 - For the second action, fill in the following:
 - Action type: imBack
 - Action value: LiveAgent
 - Action title: "Live Agent"



6. Click **Ok**, then click **Save** to get back to designer (Rearrange boxes if necessary)



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.



Bot

Hi there, I'm your Healthcare Assistant

Would you like to request a medication refill or chat with a live agent?

Medication Refill

Live Agent

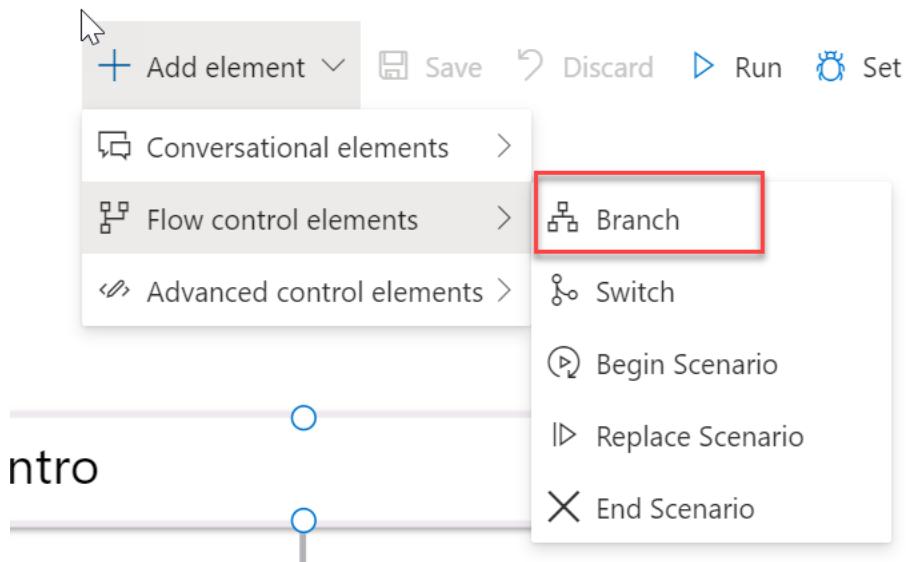
Just now

Type your message

Step 3: Add MedicationOrAgent Decision Branch

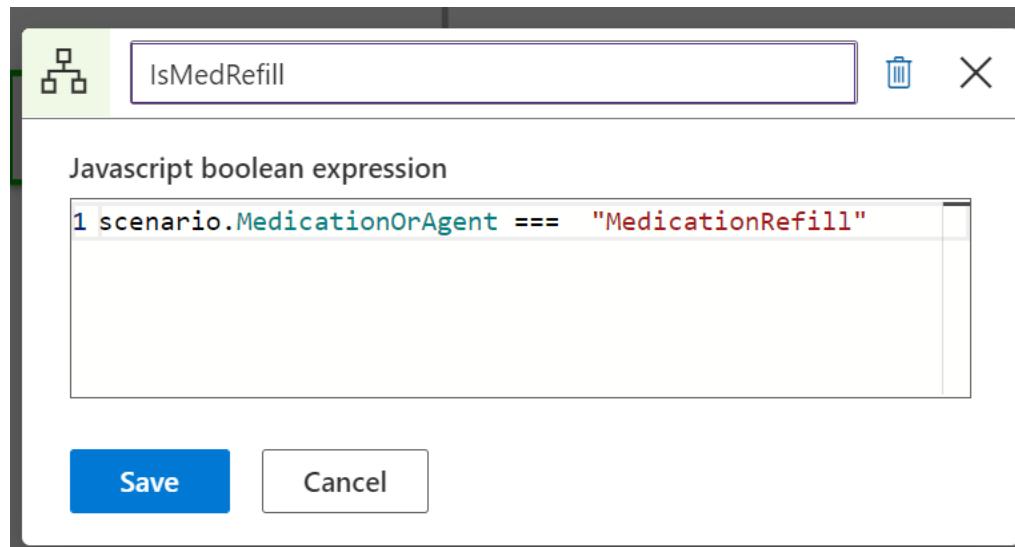
This section checks whether the user has clicked Medication Refill or Live Agent with the help of the variable MedicationOrAgent. It will redirect the message accordingly.

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

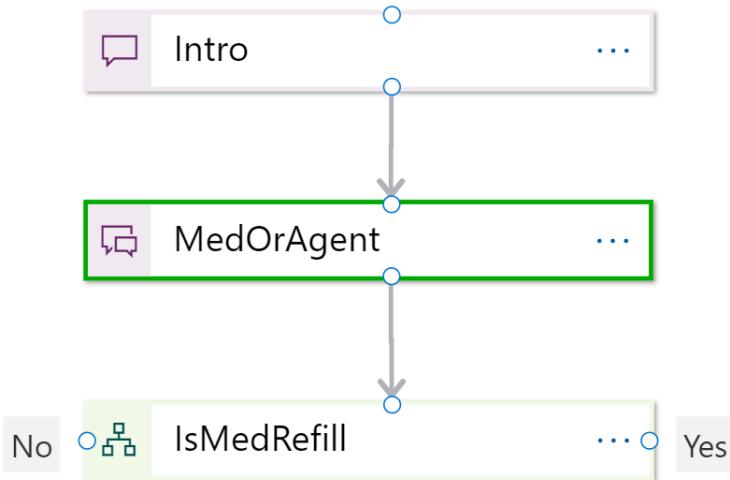


2. Enter the following in the **javascript Boolean expression**:
scenario.MedicationOrAgent === "MedicationRefill"

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

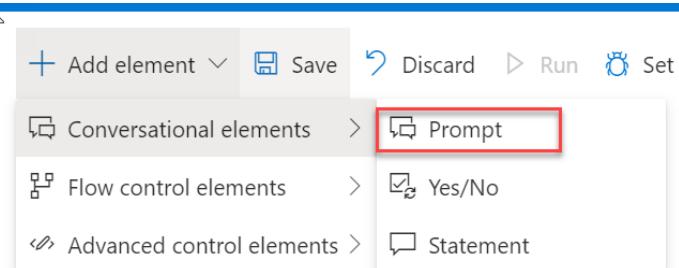


4. Select and drag the bottom circle of the **MedOrAgent** prompt to the top circle of the **IsMedRefill** branch decision to connect them.



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:
 - a. **Input Variable:** formData
 - b. **Input Type:** Object
 - c. Change Title to **Submit**
 - d. Do not add any display text since the adaptive card will display instead

The screenshot shows a configuration dialog for an Adaptive Card. At the top right are icons for 'Submit' (blue speech bubble), 'Delete' (trash can), and 'Close' (X). Below these are sections for 'Display text (optional)' (with a 'MarkDown' dropdown menu) and 'Input variable *' (set to 'formData'). A dropdown for 'Input type' is set to 'Object'. There are fields for 'Is active' (checkbox) and 'Maximum number of retries' (text input). A section for 'Message metadata (optional)' is also present. At the bottom are 'Save' and 'Cancel' buttons.

Display text (optional) [i](#)

MarkDown [v](#)

Input variable *

formData

Input type

Object [v](#)

Is active [i](#)

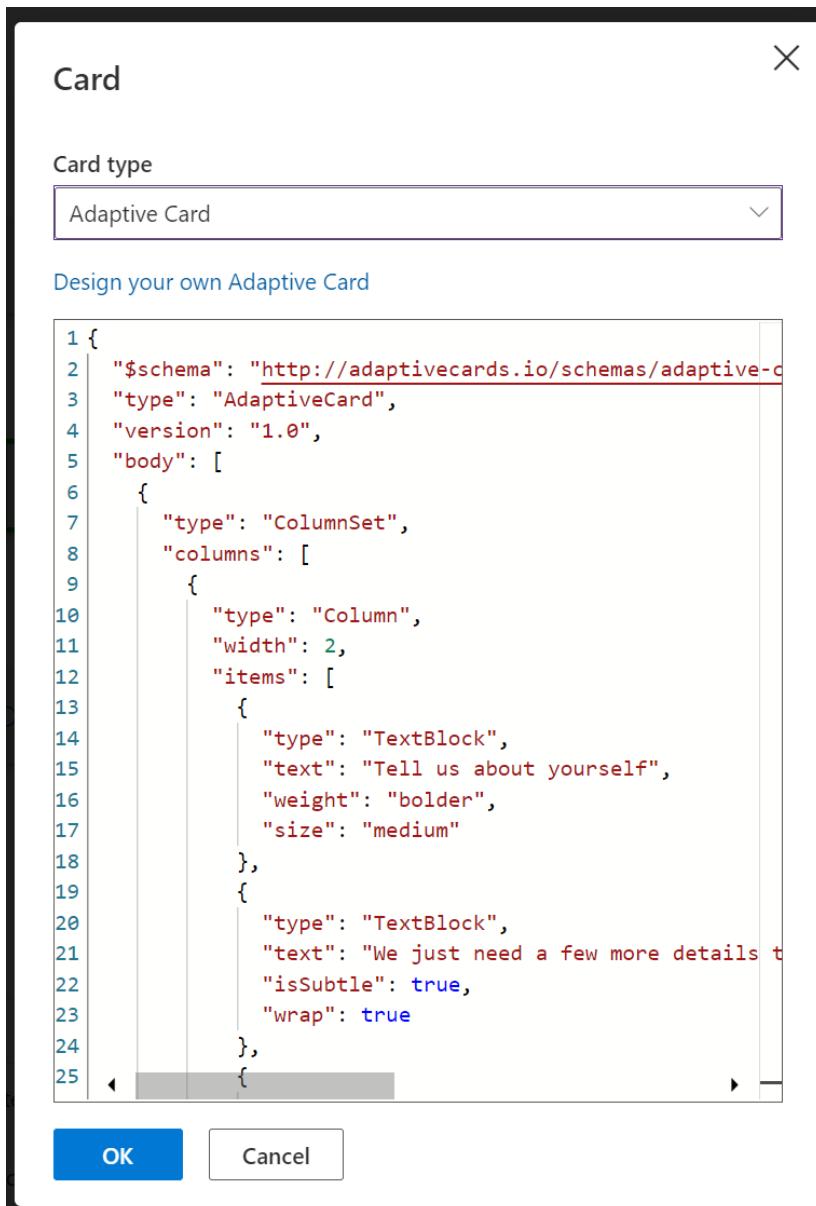
Maximum number of retries

Message metadata (optional) [i](#)

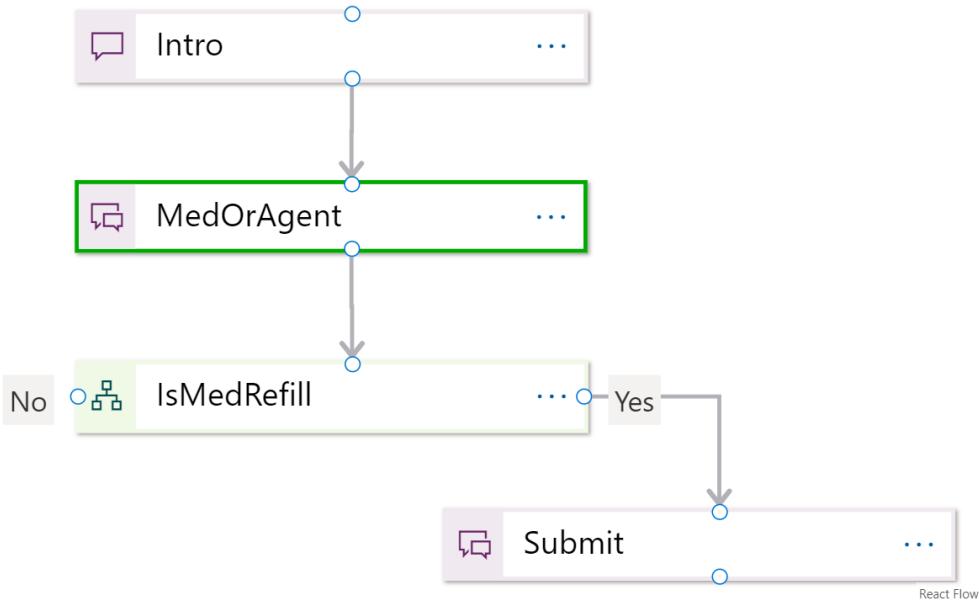
Add Cards

Save Cancel

3. Click **Add Cards** button → Set card type to: **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.



5. Select **OK** and **Save** 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.

[MCH_PatientService](#) Active [Add element](#) [Save](#) [Discard](#) [Run](#) [Set run arguments](#) [History](#)

[MCH_PatientService](#) Active [Add element](#) [Save](#) [Discard](#) [Run](#) [Set run arguments](#)

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

Hi there, I'm your Healthcare Assistant

Would you like to request a medication refill or chat with a live agent?

Medication Refill

Live Agent

A minute ago

MedicationRefill

Just now

Bot

Tell us about your...



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

Last, First

Your email

youremail@example.co

Medication Requested

Medication Name

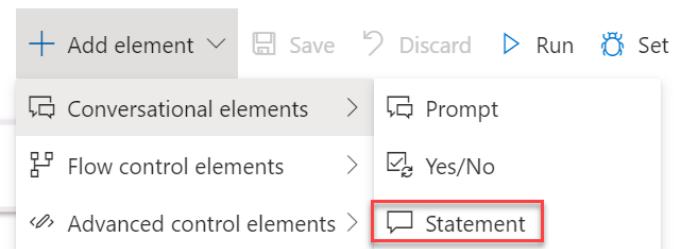
Submit

Flow

...

Step 5: Add Confirmation Statement

1. Add a **Statement** element.



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

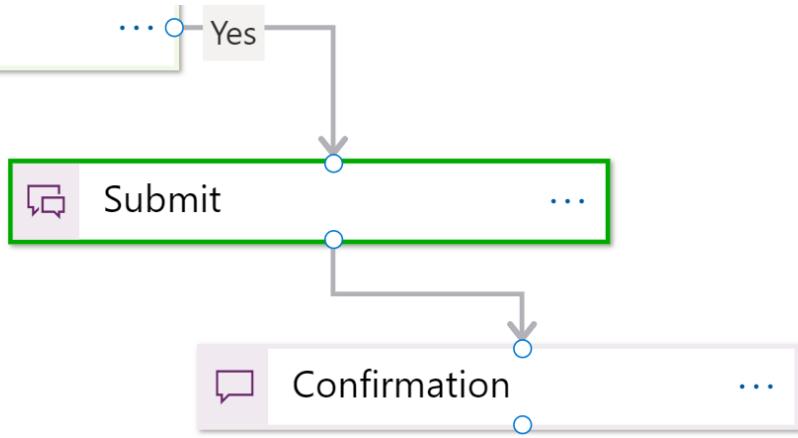
scenario.formData.myName + " - Thanks for providing the information, we have created a Medication Request for you regarding the following medication: " + scenario.formData.myMedReq

3. Rename the statement to **Confirmation** and click **Save**.

The screenshot shows the configuration dialog for the 'Confirmation' step. At the top, the name 'Confirmation' is displayed in a text input field with a red arrow pointing to it. Below it is a 'Display text' section containing the placeholder text 'a following medication: " + scenario.formData.myMedReq', which is also highlighted with a red box. Further down are sections for 'Is Active' and 'Message metadata (optional)', both of which are currently empty. At the bottom, there are 'Save' and 'Cancel' buttons.

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

MedRefill



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.

7 minutes ago

Bot

Tell us about your...



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.c

Medication Requested

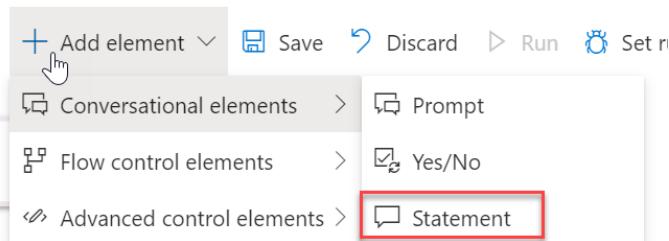
Albuterol inhaler

Submit

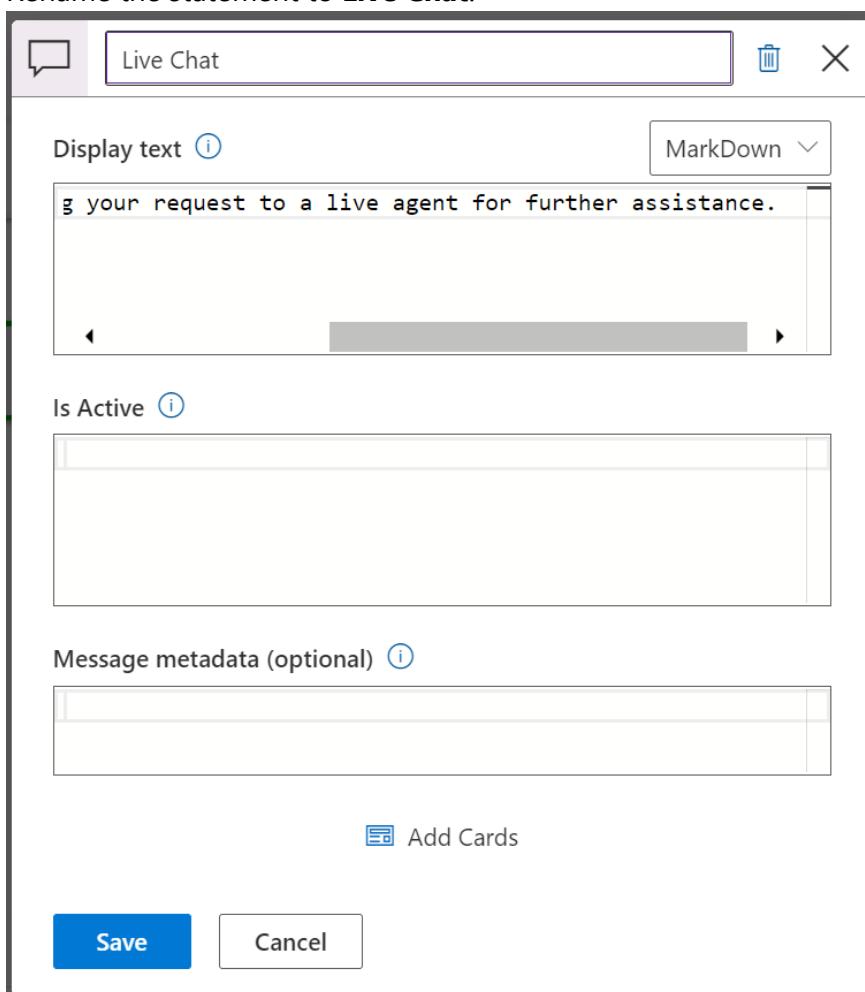
7 minutes ago

Step 6: Invoke Live Agent Action

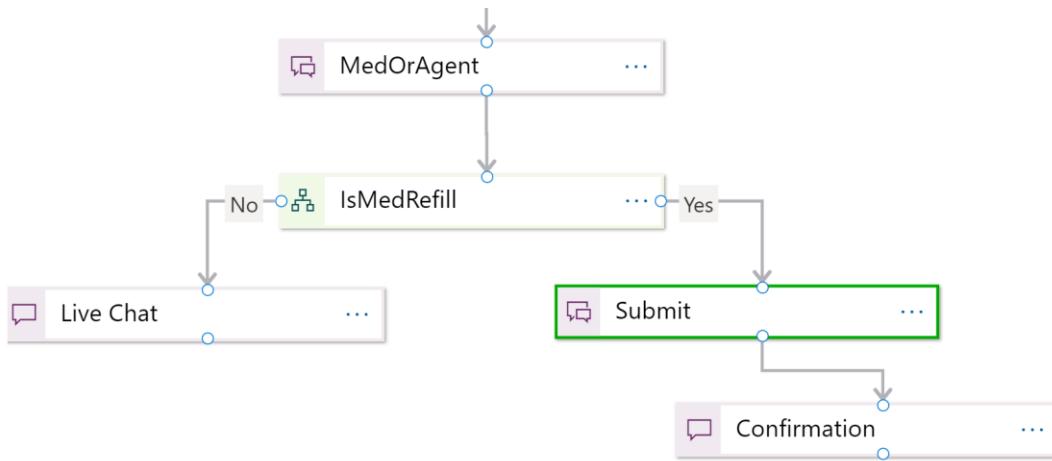
1. Add a **Statement** element to the canvas.



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**.

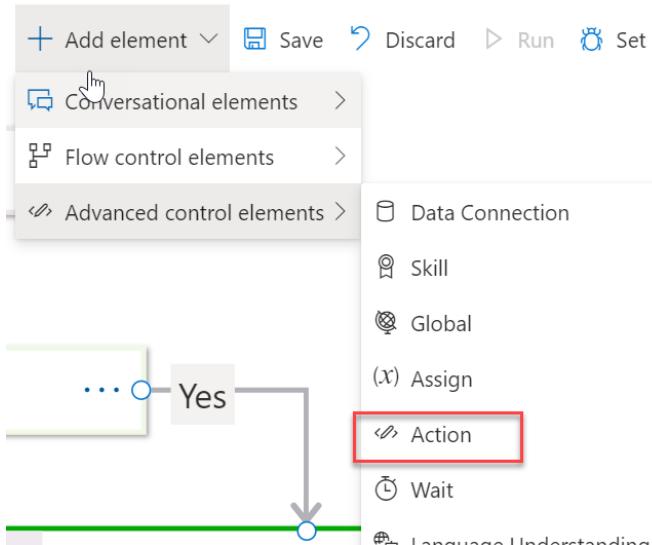


4. Click **Save** to return to the designer page.
5. **Connect** the **No** decision of the **IsMedRefill** branch to the **Live Chat** statement.



Step 7: Add Action to Invoke Escalation

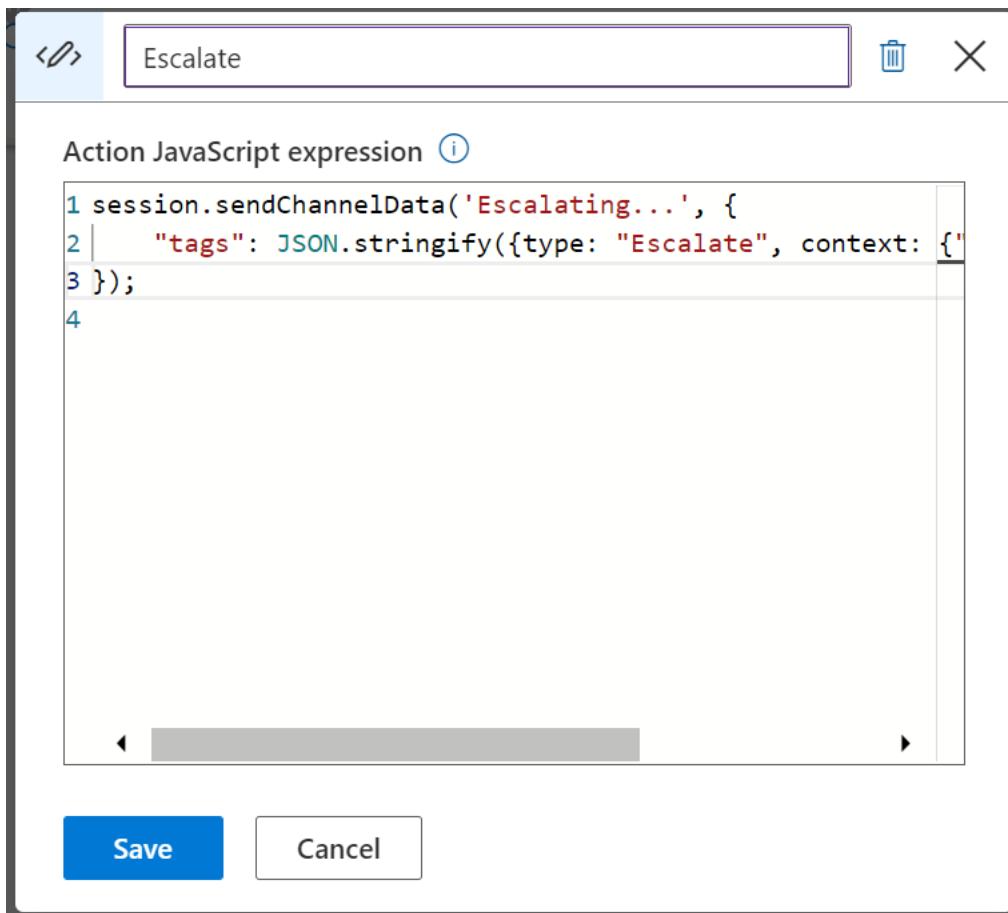
- Add an **Action** element to the canvas, used to trigger an escalation to Omnichannel Live Agent



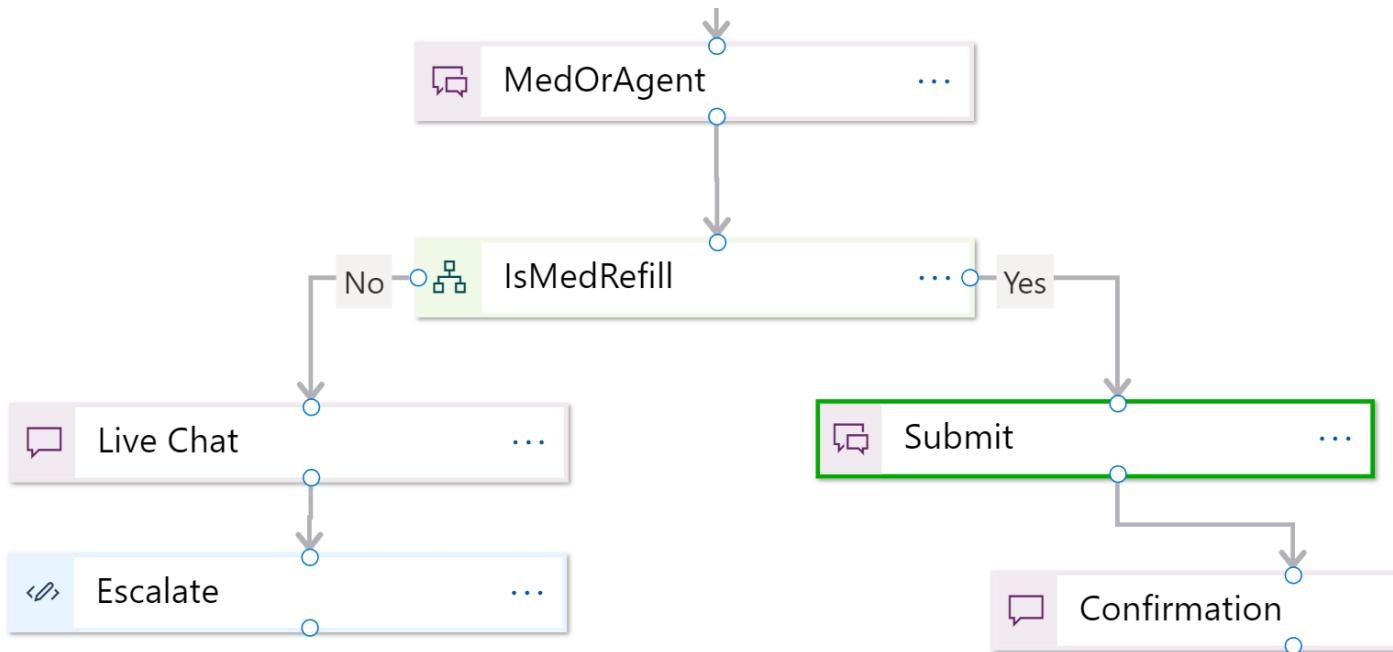
- Copy and paste the following code in the action, which will trigger the Live agent chat:

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

- Name the action **Escalate**. Click **Save** to return to the designer page.

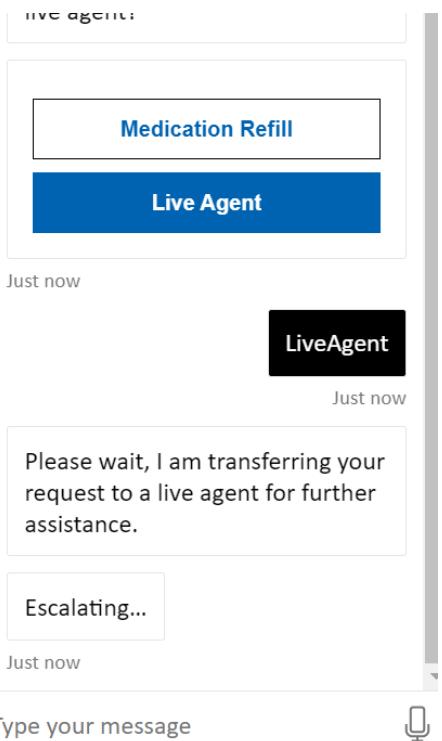


9. **Connect** the **Live Chat** to the new **EscalateToAgent** action. You completed the final connection!

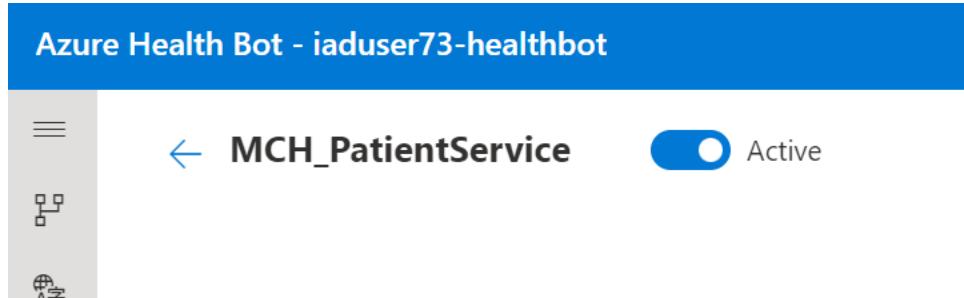


10. **Save** and **run** your scenario to see the full scenario output.

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 'Active' checked, 'Name' set to 'MCH_PatientSer...', 'Scenario ID' set to 'MCH_PatientServ...', and 'Description' partially visible.

2. Provide the following details for the new scenario:

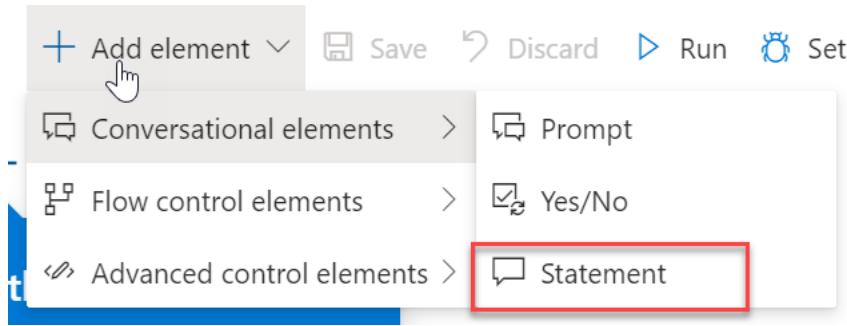
- a. **Name:** MCH_PatientServiceWelcome
- b. **Scenario ID:** MCH_PatientServiceWelcome
- c. Select **Create**.

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

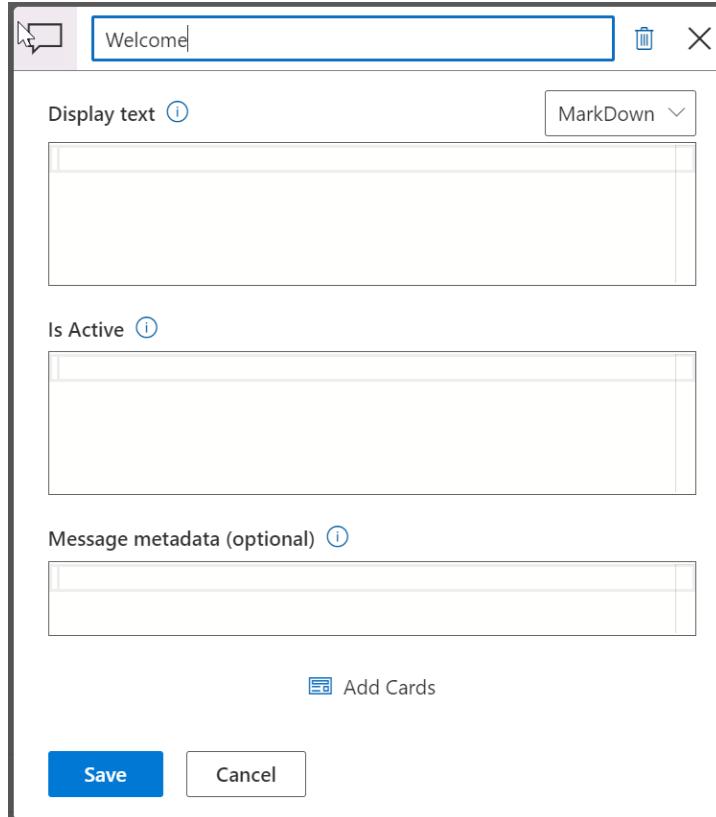
- Name***: MCH_PatientServiceWelcome
- Description**: (empty text area)
- Scenario ID***: MCH_PatientServiceWelcome
- Returning Message**: (empty text area)
- Interrupting scenario**: (radio button selected)
- Breaking scenario**: (radio button unselected)

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

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



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



5. Select **Add Cards**.
6. Choose **HeroCard**. Add **Title**: Welcome to Lamna Healthcare Patient Service Portal

Card

X

Card type

Hero Card

Image URL



Title

Welcome to Lamna Healthcare Patient Service Portal

Sub title

Actions

+

OK

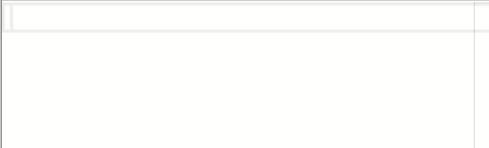
Cancel

7. 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
Hero Card

Image URL
 

Title
Welcome to Lamna Healthcare Patient Service Portal

Sub title

Actions +

Action type	Action value	Action title
imBack	leptService"	are Support"

OK **Cancel**

- Click **OK** and **Save** to 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.

[MCH_PatientService...](#) Active [Add element](#) [Save](#) [Discard](#) [Run](#) [Set run arguments](#) [History](#) [Web Chat](#)



- Save and run to test your bot scenario **MCH_PatientServiceWelcome** scenario in the Web Chat.

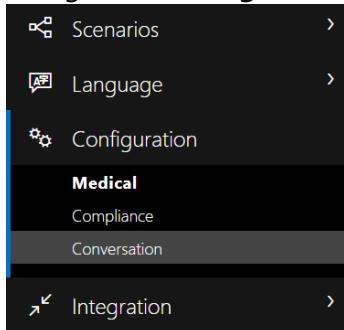


- Exit the scenario designer.

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 interface. The top navigation bar shows 'Azure Health Bot' and a user 'iaduser99-healthbot'. The left sidebar is identical to the one in the previous screenshot. The main area is titled 'Interactions' and contains sections for 'Default retry message (date prompt)', 'Default retry message (attachment prompt)', and 'Automatic welcome'. The 'Automatic welcome' section includes a note about multiple ways to display messages, a placeholder for the 'Automatic welcome message', and a dropdown for 'Automatic welcome scenario' which is currently set to '** scenario not selected **'. 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' section from the previous interface. The 'Automatic welcome message' field is empty. Below it, the 'Automatic welcome scenario' dropdown is open, showing a list with 'MCH_PatientServiceWelcome' selected.

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**.

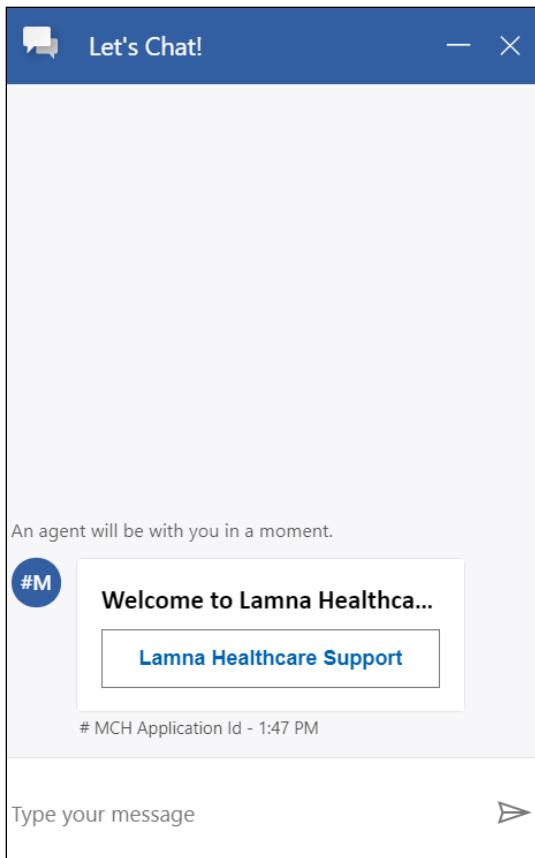
The screenshot shows the Microsoft Power Apps portal interface. On the left is a navigation sidebar with options like Home, Learn, Apps, Create, Data, Flows, Chatbots, AI Builder, and Solutions. The 'Apps' section is selected. The main area displays a list of apps under the heading 'Apps Component libraries (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 the banner 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 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**.

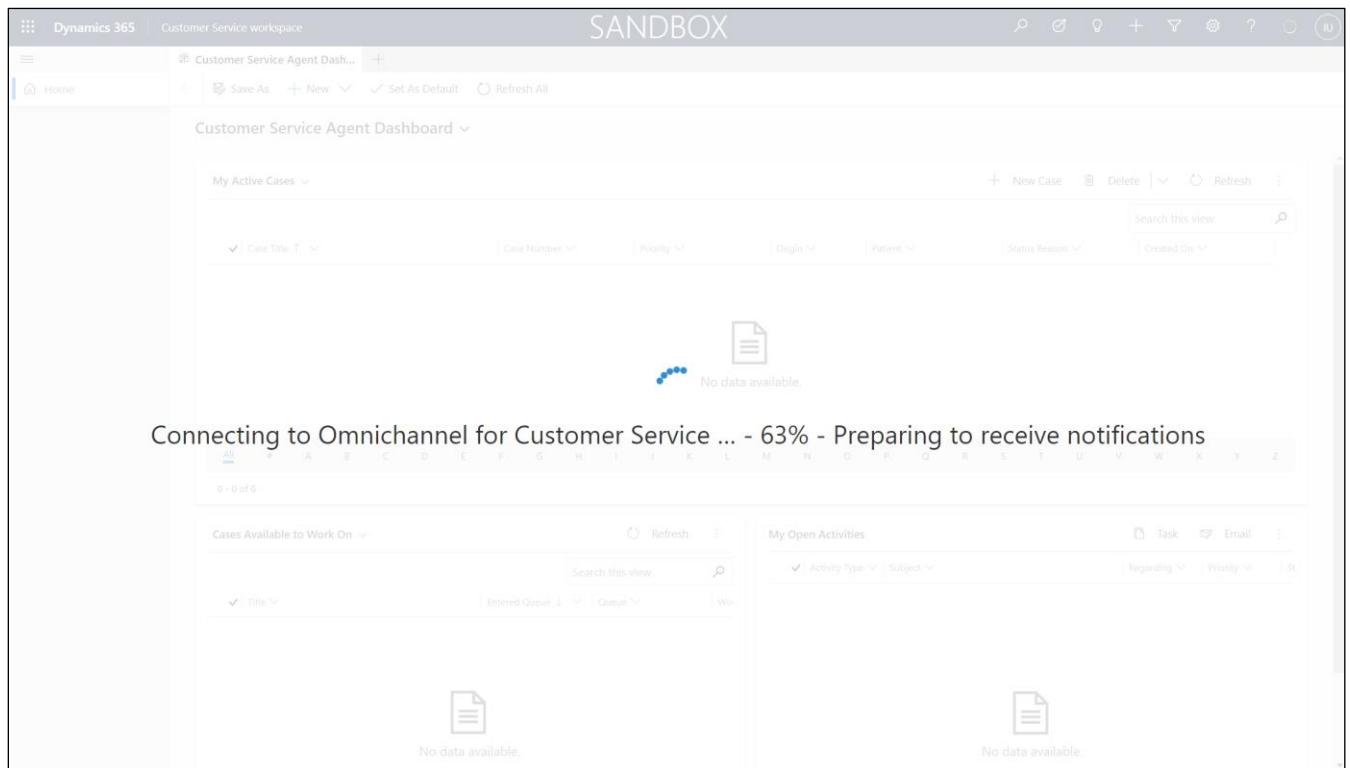
The screenshot shows the Power Apps portal. Under the 'Apps' section, the 'Customer Service workspace' app is selected, indicated by a checkmark icon and highlighted in grey. Other apps like 'Customer Service Hub' are also listed.

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

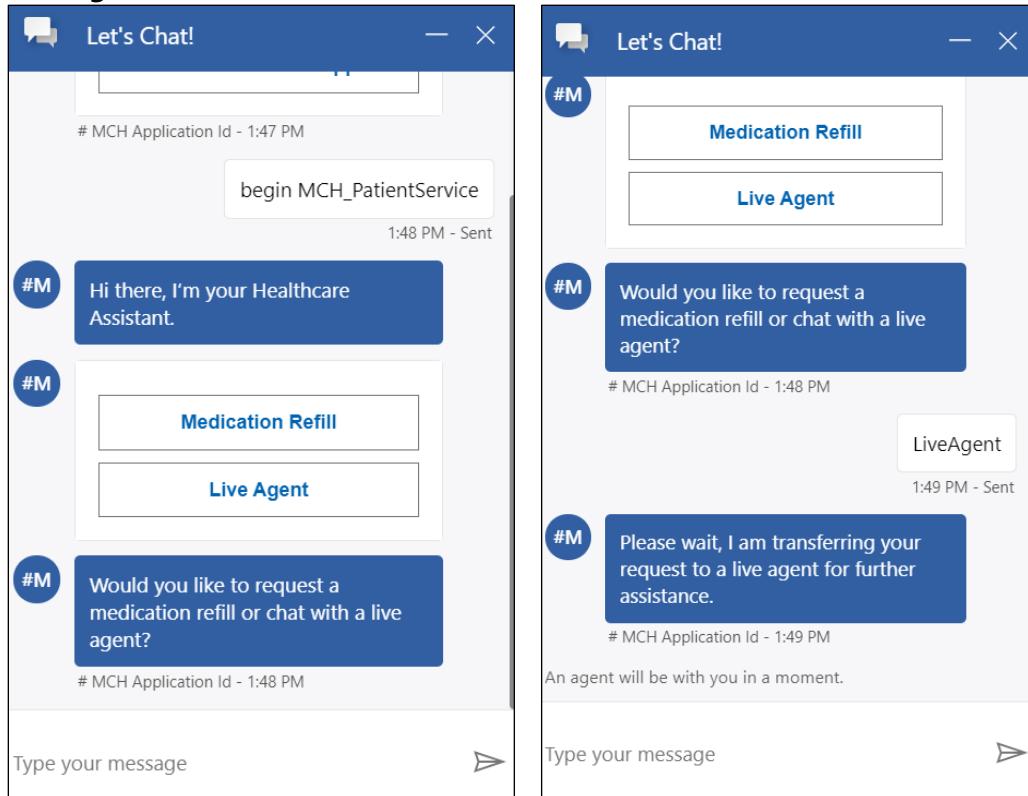
There should be a splash loading screen that goes through multiple steps and then displays the status indicator as available once loaded. (Status is enabled when green with checkmark in circle)

The screenshot shows the Dynamics 365 Customer Service workspace. The top navigation bar has a 'Sandbox' tab. In the top right, there is a presence status indicator (green circle with a checkmark) which is highlighted with a red arrow. The main dashboard shows a 'Customer Service Agent Dashboard' with a 'My Active Cases' grid and various filters at the bottom.

Splash screen:

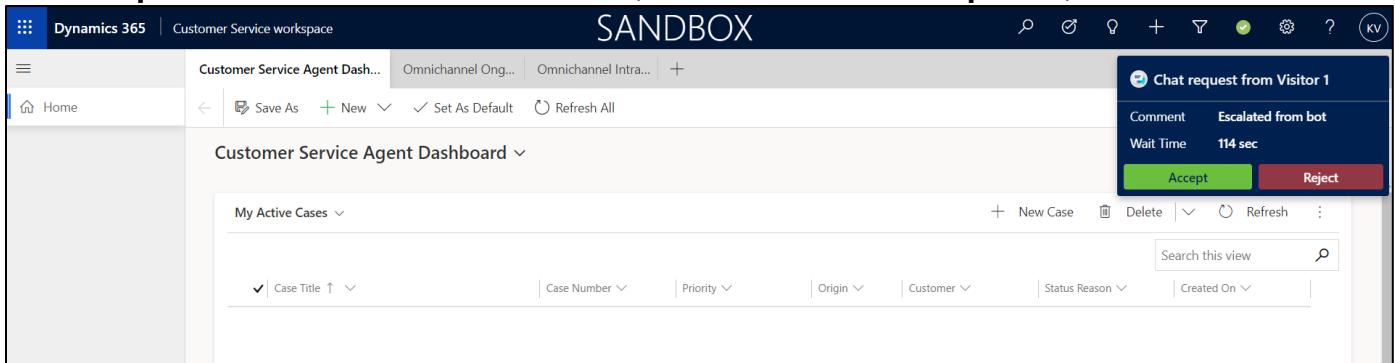


- In the Health Bot in Lamna Healthcare Patient Portal, click **Lamna Healthcare Support** button, then the **Live Agent** button to witness the escalation into Omnichannel to chat with a live agent (your user!)

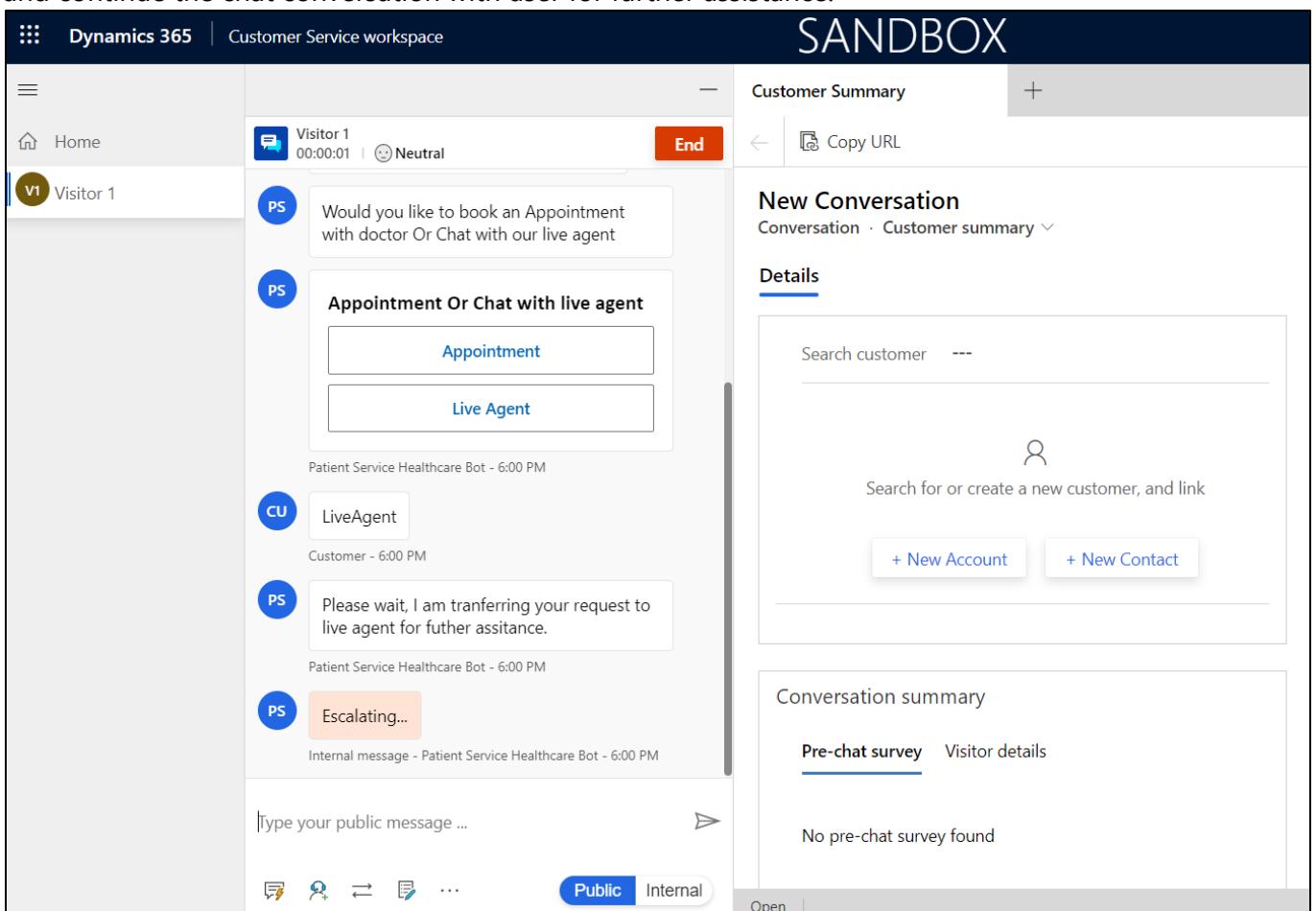


- Navigating back to Omnichannel for Customer Service, your user as the **Live Agent** should receive an incoming notification with **Accept/Reject** options for that chat.

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