Microsoft 365

3-1. Conversational Bots

Create interactive conversational bots for Microsoft Teams

Kanghee(Joe) Cho, Consultant
Modern Workplace

Oct. 12th, 2021

Agenda

01 Overview of bots in Microsoft Teams

1. Classification by Conversation Type
2. How Do bots Work?

02 Developing bots for Microsoft Teams

1. Creating conversational bots for Microsoft Teams
2. Web Service
3. Basics of conversation bots
4. Register the Web Service as a bot using Azure Bot
5. Microsoft Teams app manifest and app package with Developer Portal

DEMO 1 : Creating Conversational bots for Microsoft Teams

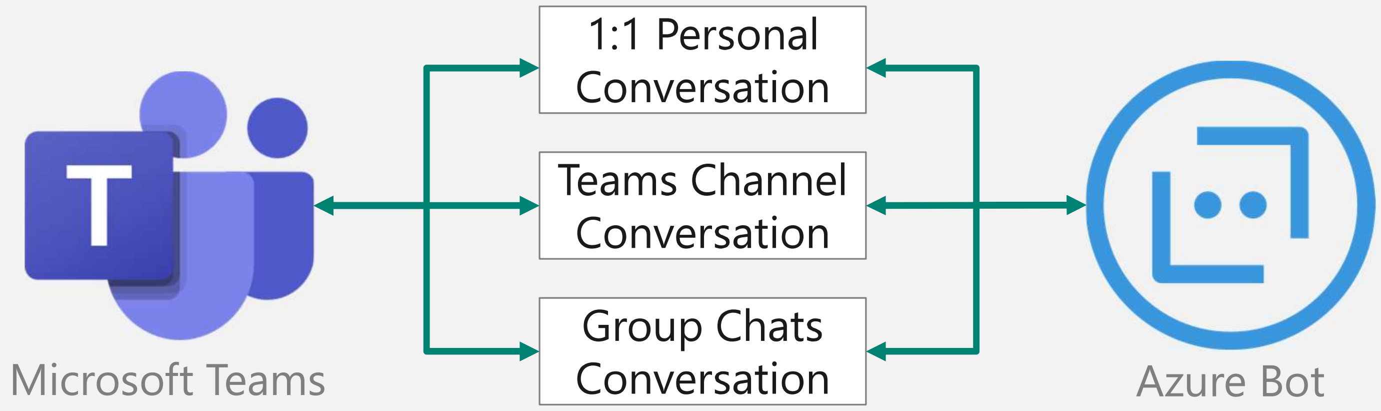
DEMO 2 : Bots in Microsoft Teams channels and group chats

DEMO 3 : Proactive Messages from bots

01. Overview of bots in Microsoft Teams

Classification by Conversation Type 1/2

- Bots can in Microsoft Teams can interact with users in the following ways:
  - Personal Chats – This conversation type includes conversations between bots and a single user.
  - Channel Chats – This conversation type is visible to all members of the channel.
  - Group Chats – This conversation type includes chant between a bot and two or more users. It also enables your bot in meeting chats.



[Conversation basics - Teams | Microsoft Docs]

Classification by Conversation Type 2/2

1:1 Personal Conversation

- Personal, 1:1 Chats between users and bots is the traditional use case for bots
- Always consider if a bot is the best way to present functionality
- Enable diverse workloads & can initiate workflows
- **Example:** Wizard process approach (Booking system)
  Task Modules (Approval system)
  Giving Information (ARS, FAQ or Q&A System)

Team Channel Conversation

- Channels contain threaded conversations between multiple people
- Bots have potential to massive reach to these users
- Bots have access to messages where they are directly @mentioned
- **Example:** Notifications
  Feedback

Group Chats Conversation

- Group chats are non-threaded conversations between three or more people
- Tend to have fewer members, but similar to a channel
- Bots have access to messages where they are directly @mentioned
- All scenarios where a bot works well in a channel will usually work well in a group chat

How do bots work?

- Conversation bots consist of the following components:
  - Publicly accessible web service
  - Bot Registration that identifies your web service with Microsoft Bot Framework
  - Teams App registration that identifies the bot and links it with the Bot Framework registration

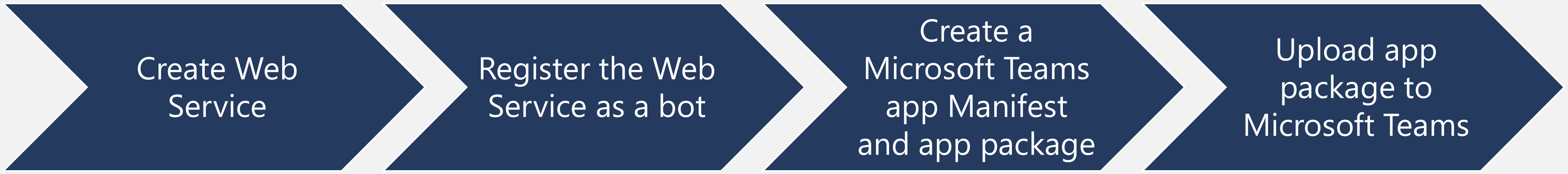
- How to make bots in Microsoft Teams unique
  - Bots created with the Microsoft Bot Framework are diverse & can be use in multiple channels
  - Bots developed for Microsoft Teams include some differences from the other platforms
    - Primary difference: how activities are handled
  - Microsoft Teams activity handler derives from the Bot Framework
  - Route all Teams activities before allowing any non-Teams-specific activity to be handled

- Microsoft Teams activity handlers
  - When a Microsoft Teams bot receives an activity, it's passed to *Activity Handlers*
  - These are derived on one base handler – the **Turn Handler**
  - The turn handler calls the required activity handler to handle the specific type of received activity
  - When creating bots for Microsoft Teams, use the **TeamsActivityHandler** class from the SDK that's derived from the Microsoft Bot Framework **ActiviryHandler** class

02. Developing bots for Microsoft Teams

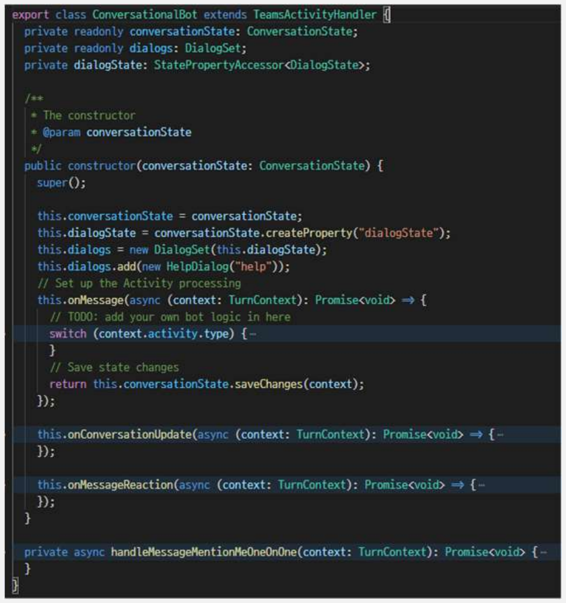
Creating Conversational bots for Microsoft Teams

- Creating a Conversational bot for Microsoft Teams requires the following things:


Create Web Service
Register the Web Service as a bot
Create a Microsoft Teams app Manifest and app package
Upload app package to Microsoft Teams

Web Service

- Web Service
  - The Web Service is the heart of your bot
  - Defines a single HTTPS route where it receives all requires
  - Microsoft Bot Framework will send different types of messages to your web service
  - **Recommendation:** Use the available SDKs to implement your web service
    - Without the SDK: receive, inspect and process messages of type `composeExtension/fetchTask`
    - With the SDK: implement the `handleTeamsMessagingExtentionFetchTask()` method



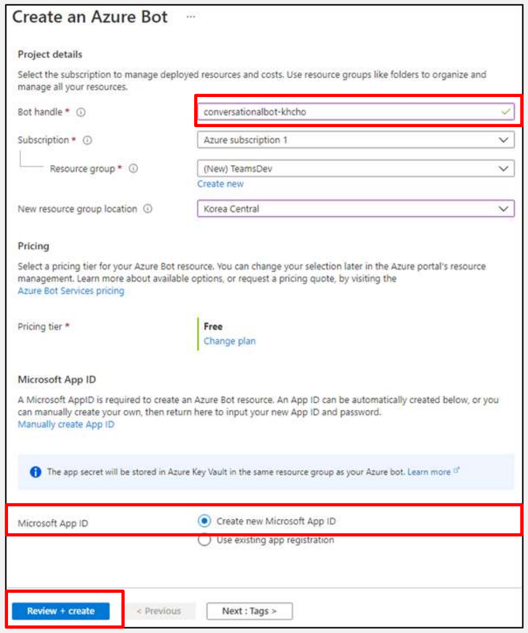
```
export class ConversationalBot extends TeamsActivityHandler {
  private readonly conversationState: ConversationState;
  private readonly dialogs: DialogSet;
  private dialogState: StatePropertyAccessor<DialogState>;

  /**
   * The constructor
   * @param conversationState
   */
  public constructor(conversationState: ConversationState) {
    super();

    this.conversationState = conversationState;
    this.dialogState = conversationState.createProperty("dialogState");
    this.dialogs = new DialogSet(this.dialogState);
    this.dialogs.add(new HelpDialog("help"));
    // Set up the Activity processing
    this.onMessage(async (context: TurnContext): Promise<void> => {
      // TODO: add your own bot logic in here
      switch (context.activity.type) {…
      }
      // Save state changes
      return this.conversationState.saveChanges(context);
    });

    this.onConversationUpdate(async (context: TurnContext): Promise<void> => {…
    });

    this.onMessageReaction(async (context: TurnContext): Promise<void> => {…
    });
  }

  private async handleMessageMentionMeOneOnOne(context: TurnContext): Promise<void> {…
  }
}
```

Basics of conversation bots

- Conversations are series of messages between one or more users & a bot in an available scope
  - Team (`teams`)
  - Group Chat (`groupChat`)
  - Personal (`personal`)
- Bots behave differently depending on the scope
  - Must be @mentioned to activate the bot in a team conversation & group chat
  - Can access messages in a personal, 1:1 chat with a user
- **Activities**
  - All messages are sent as *activities* and contain a `messageType` property
- **Receive messages**
  - Use the `Activity.text` property to inspect the message
- **Send Messages**
  - Send an `Activity` to the Microsoft Bot Framework using the turn context's `SendActivity()` method

Register the Web Service as a bot using Azure Bot

- Register the Web Service as a Bot
  - The Web Service must be registered as a bot with the Microsoft Bot Framework
  - Provides a secure communication channel between Microsoft Teams clients and your web service
    - Microsoft Teams & your web service never communicate directly
- Azure Portal: https://portal.azure.com
  - Click [Create a resource] → Search "Azure Bot" → Create [Azure Bot]



Create an Azure Bot ···

Project details

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

Bot handle *	conversationalbot-khcho
Subscription *	Azure subscription 1
Resource group *	(New) TeamsDev Create new
New resource group location	Korea Central

Pricing

Select a pricing tier for your Azure Bot resource. You can change your selection later in the Azure portal's resource management. Learn more about available options, or request a pricing quote, by visiting the Azure Bot Services pricing

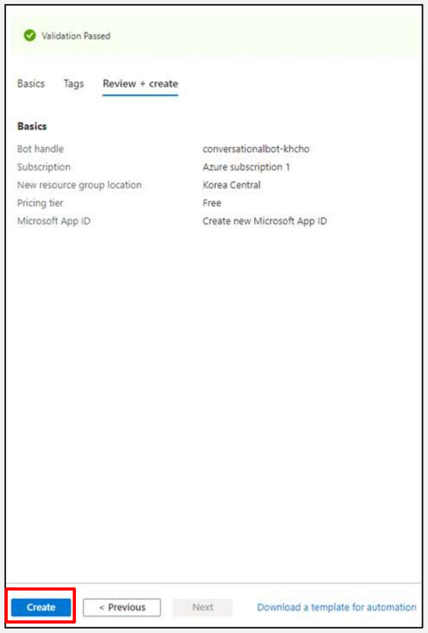
Pricing tier *	**Free** Change plan

Microsoft App ID

A Microsoft AppID is required to create an Azure Bot resource. An App ID can be automatically created below, or you can manually create your own, then return here to input your new App ID and password.
Manually create App ID

The app secret will be stored in Azure Key Vault in the same resource group as your Azure bot. Learn more

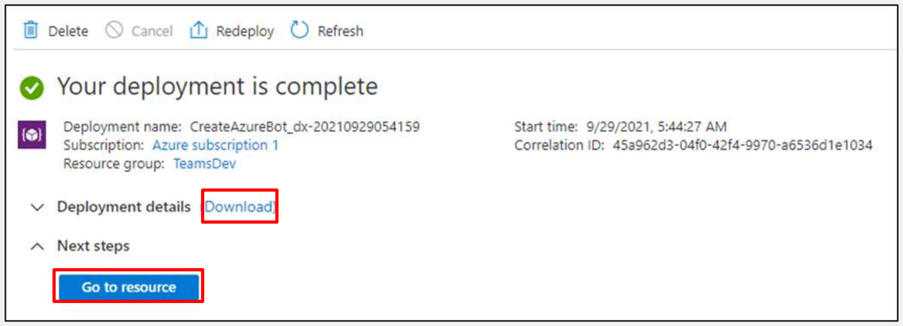
Microsoft App ID: [✓ Create new Microsoft App ID] Use existing app registration



Validation Passed

Basics

Bot handle	conversationalbot-khcho
Subscription	Azure subscription 1
New resource group location	Korea Central
Pricing tier	Free
Microsoft App ID	Create new Microsoft App ID

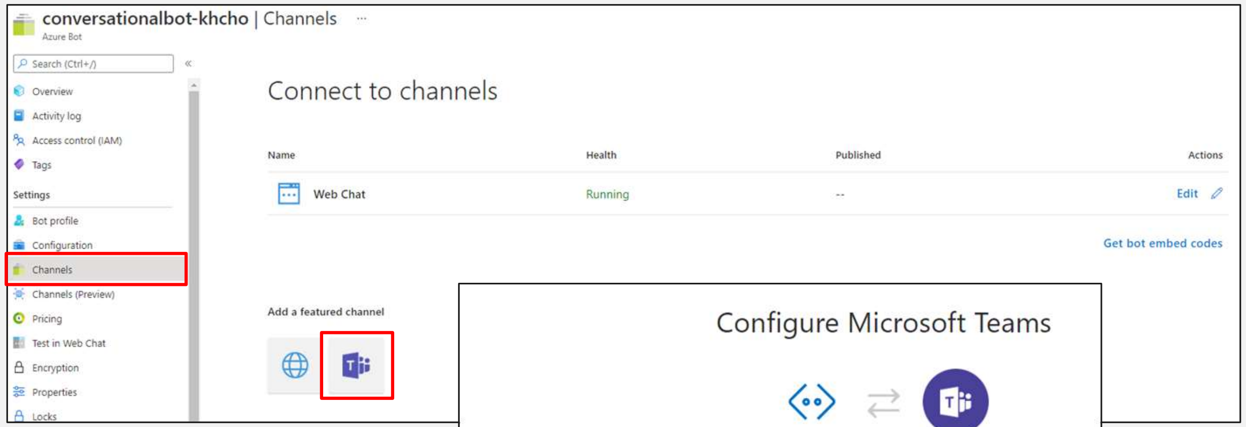


Your deployment is complete

Deployment name: CreateAzureBot_dx-20210929054159
Subscription: Azure subscription 1
Resource group: TeamsDev
Start time: 9/29/2021, 5:44:27 AM
Correlation ID: 45a962d3-04f0-42f4-9970-a6536d1e1034

Azure Bot Service [Setting Channel]

- Azure Portal: https://portal.azure.com
  - Click [Go to Resource] → Click [Channels] → Click [Teams Icon]



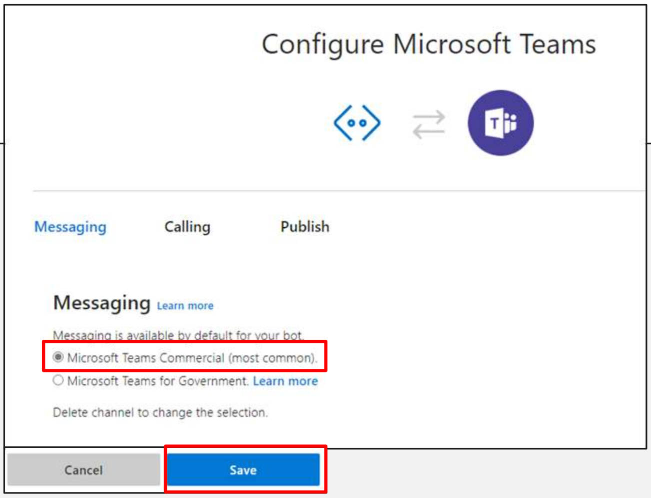
conversationalbot-khcho | Channels
Azure Bot

Connect to channels

Name	Health	Published	Actions
Web Chat	Running	--	Edit

Get bot embed codes

Add a featured channel



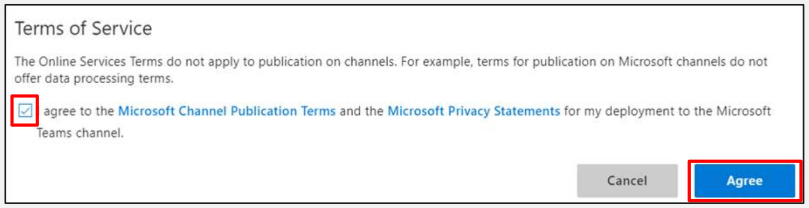
Configure Microsoft Teams

Messaging Learn more

Messaging is available by default for your bot

[✓ Microsoft Teams Commercial (most common).]
Microsoft Teams for Government. Learn more

Delete channel to change the selection.



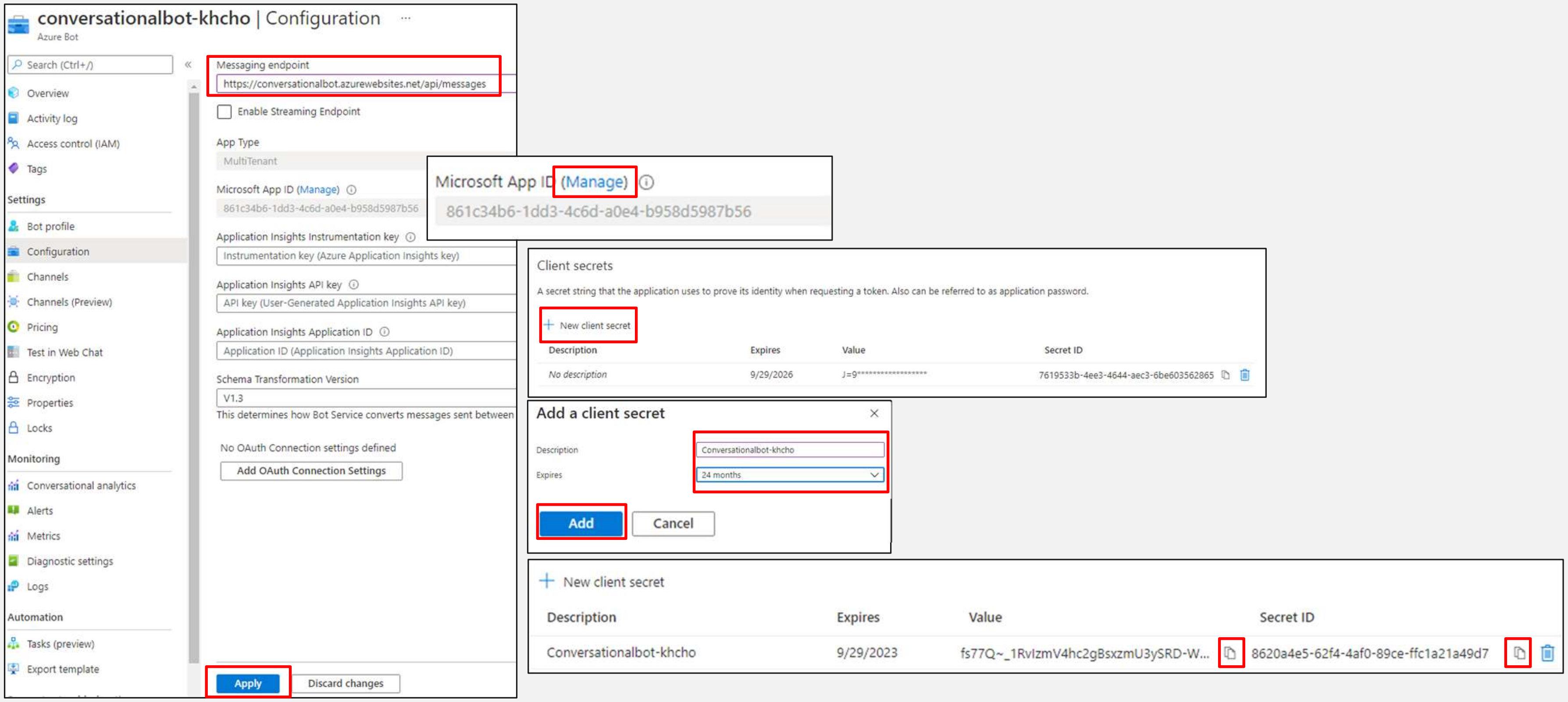
Terms of Service

The Online Services Terms do not apply to publication on channels. For example, terms for publication on Microsoft channels do not offer data processing terms.

[✓] I agree to the Microsoft Channel Publication Terms and the Microsoft Privacy Statements for my deployment to the Microsoft Teams channel.

Azure Bot Service [Setting Configuration]

- Azure Portal: https://portal.azure.com
  - Click [Configuration] → Messaging Endpoint `Your Web Service URL` → Click [Apply] → Click on [Manage] next to [Microsoft App ID]



Messaging endpoint: https://conversationalbot.azurewebsites.net/api/messages

Enable Streaming Endpoint

App Type: MultiTenant

Microsoft App ID (Manage): 861c34b6-1dd3-4c6d-a0e4-b958d5987b56

Application Insights Instrumentation key: Instrumentation key (Azure Application Insights key)

Application Insights API key: API key (User-Generated Application Insights API key)

Application Insights Application ID: Application ID (Application Insights Application ID)

Schema Transformation Version: V1.3

This determines how Bot Service converts messages between

No OAuth Connection settings defined

Microsoft App ID (Manage)

861c34b6-1dd3-4c6d-a0e4-b958d5987b56

Client secrets

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

Description	Expires	Value	Secret ID
No description	9/29/2026	J=9******************	7619533b-4ee3-4644-aec3-6be603562865

Add a client secret

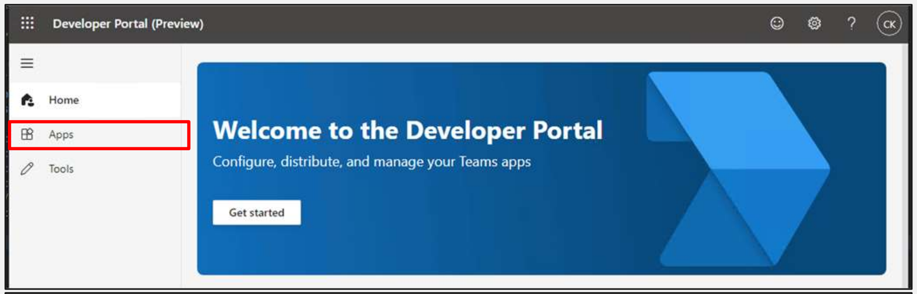
Description: Conversationalbot-khcho

Expires: 24 months

Description	Expires	Value	Secret ID
Conversationalbot-khcho	9/29/2023	fs77Q~_1RvIzmV4hc2gBsxzmU3ySRD-W...	8620a4e5-62f4-4af0-89ce-ffc1a21a49d7

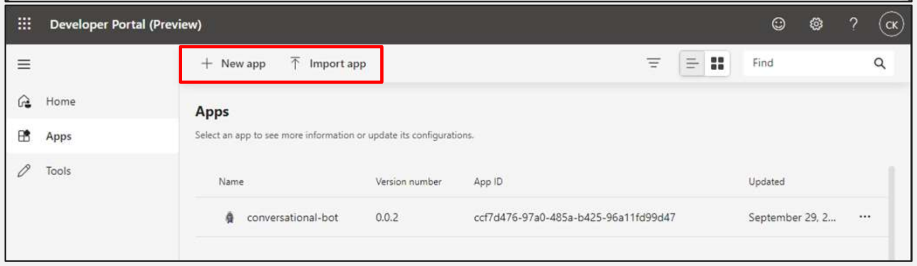
Microsoft Teams app manifest and app package with Developer Portal

- Developer Portal: https://dev.teams.Microsoft.com
    - Bot must be registered with the Microsoft Teams app manifest, then uploaded to Microsoft Teams



Welcome to the Developer Portal

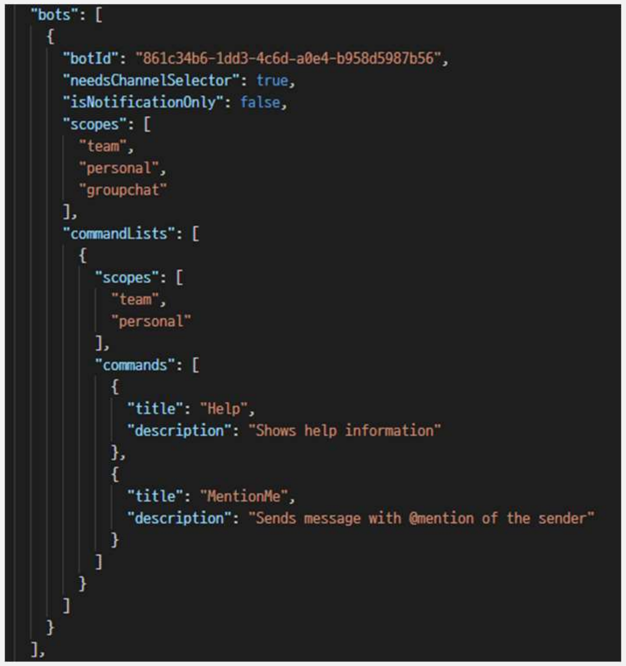
Configure, distribute, and manage your Teams apps



Apps

Select an app to see more information or update its configurations.

Name	Version number	App ID	Updated
conversational-bot	0.0.2	ccf7d476-97a0-485a-b425-96a11fd99d47	September 29, 2...



```
"bots": [
  {
    "botId": "861c34b6-1dd3-4c6d-a0e4-b958d5987b56",
    "needsChannelSelector": true,
    "isNotificationOnly": false,
    "scopes": [
      "team",
      "personal",
      "groupchat"
    ],
    "commandLists": [
      {
        "scopes": [
          "team",
          "personal"
        ],
        "commands": [
          {
            "title": "Help",
            "description": "Shows help information"
          },
          {
            "title": "MentionMe",
            "description": "Sends message with @mention of the sender"
          }
        ]
      }
    ]
  }
],
```

DEMO 1 :
Creating Conversational bots for Microsoft Teams

In this exercise, you’ll learn how to create and add a new bot to a Microsoft Teams app and interact with it from the Microsoft Teams Client

Conversation bots in Microsoft Teams

- Microsoft Teams sends notifications to your bot for events that happen in scopes where your bot is active

- Capture events in your code & take action on them:
  - Trigger welcome message when your bot is added to a team
  - Trigger welcome message when a new team member is added or removed
  - Trigger notifications when channels are created/renamed/deleted
  - When one of the bot's messages are liked by users

- Conversation Update events
- Bot receive `ConversationUpdate` Events when:
  - It's been added to a conversation
  - Other members are added/removed from a conversation
  - Conversation metadata changes
- The event is sent to your bot when it receives information on membership updates to teams where it's been added
- Also receives updates when it's been added for the first time for personal conversations

DEMO 2 :
Bots in Microsoft Teams channels and group chats

Conversation bots can do many things within the Microsoft Teams client. They can proactively send a message to a channel or group chat, listen for and act on Microsoft Teams specific events and even update their own message

Proactive messages

- Proactive messages are when the bot creates a new message in a channel

- Possible scenarios
  - Welcome message for personal bot conversation
  - Poll responses
  - Notification of external events

- Consider when to use proactive messages
  - Proactive messages can be an effective way to communicate with users
  - However, consider from a user's perspective, message appears to come to them unprompted
  - Welcome messages will be the first time they interact with your app
  - **Recommendation**: Consider using proactive messages sparingly

DEMO 3 :
Proactive Messages from bots

In this exercise, you’ll update the existing Teams app to send a proactive message from your bot

Microsoft 365

© Copyright Microsoft Corporation. All rights reserved.