inbenta

# Microsoft Team Integration

# Index

# Introduction

The purpose of this documentation is to define the integration of Inbenta's Chatbot and Hyperchat solution with Microsoft Teams.

# Features

These are the supported answer types and features:

- Text answers
- Multiple Options buttons (3 buttons max)
- Polar Questions
- Chained answers
- Forms
- Custom FAQ title in buttons when displaying multiple options

Other important features:

- Content ratings (yes/no + comment)
- HyperChat escalation after X no-results answers (triesBeforeEscalation)
- Escalate to HyperChat when the FAQ has 'ESCALATE' setting set to 'TRUE'

# Prepare your Inbenta instances

## Create translations object in ExtraInfo (optional)

You can manage the translation labels from Extra Info. Here are the steps to create the translations object:

1. In your Inbenta App instance, go to *Knowledge* > *Extra Info*. Click on '*Manage groups and types* > *teams* > *Add type*'. Name it '**translations**', add a new property with 'Multiple named with your chatbot's language label (en, es, it…)
2. Inside the language object, add all the labels that you want to override. Each label should be a 'text' type entry (you can find the labels list below).
3. Save your translations object.

You can now create the ExtraInfo object by clicking the **New entry** button, selecting the 'translations' type and naming it as 'translations'. Then, fill each label with your desired translation and remember to publish ExtraInfo by clicking the **Post** button in *Knowledge* > *Contents*.

Here is a list of all the current labels with their English value:

- **agent_joined** => 'Agent $agentName has joined the conversation.'
- **api_timeout** => 'Please, reformulate your question.'
- **ask_rating_comment** => 'Please tell us why'
- **ask-to-escalate** => 'Do you want to start a chat with a human agent?'
- **chat_closed** => 'Chat closed'
- **creating_chat** => 'I will try to connect you with an agent. Please wait.'
- **error_creating_chat** => 'There was an error joining the chat'
- **escalation_rejected** => 'What else can I do for you?'
- **no** => 'No'
- **no_agents** => 'No agents available'
- **rate-content-intro** => 'Was this answer helpful?'
- **read_answer** => 'Read this response'
- **thanks** => 'Thanks !'
- **yes** => 'Yes'
- **validate** => 'Confirm'

**Information**

Remember: Always publish your changes to *Extra Info* using the 'Post' button in *Knowledge > Contents*.

## HyperChat integration (optional)

If you use HyperChat, you must subscribe your UI to the HyperChat events. Open your Messenger instance and go to *Messenger* > *Settings* > *Chat* > *Webhook*. Find the 'Events' column and type "queues:update,invitations:new,invitations:accept,forever:alone,chats:close,messages:new,users:activity". In the 'Target' column paste the URL of your UI, then click on the '+' button to the right.

# Building the Microsoft Teams template

## Setup Chatbot Microsoft Teams Template

### Download Microsoft Teams Template

You can find the Chatbot Microsoft Teams template in GitHub. You can download or clone the template from https://github.com/inbenta-integrations/microsoft_teams_chatbot_template.

### Required Configuration

In your UI directory, go to **conf**. There is a readme file with examples of structures and explanations of use. If you want to build a chatbot, fill the **key** and **secret** value inside the **conf/custom/api.php** file with your Inbenta Chatbot API credentials. There is also the **microsoft_appid** and **microsoft_secret** to fill with credentials obtained from the Azure portal. If you want to modify other configuration parameters, copy the desired file(s) from **conf/default** into **conf/custom** and modify the values.

### Optional Configuration

You can enable several optional features from the configuration files. You must copy every optional configuration file from **conf/default** and store the custom version in **conf/custom**. The bot detects and loads the customized files automatically.

Here is a list of the optional configuration files, with a description of the configuration fields.

**HYPERCHAT** (chat.php)

- **chat**
    - **enabled**: Enable or disable HyperChat ("**true**" or "**false**").
    - **version**: HyperChat version. The default and latest one is 1.
    - **appId**: The ID of the HyperChat app. This defines the instances in which the chat opens. You can find it in your instance under *Messenger* > *Settings* > *Chat*.
    - **secret**: Your HyperChat instance application secret. You can find it in your instance under *Messenger* > *Settings* > *Chat*.
    - **roomId**: The room where the chat opens. This is mapped directly to a queue ID in the Inbenta Messenger App. This is a numeric value, not a string. You can find the list of your rooms in your instance under *Messenger* > *Settings* > *Queues*.
    - **lang**: Language code (in ISO 639-1 format) for the current chat. This is used when the engine checks if there are agents available for this language to assign the chat to one of them.
    - **source**: Source id from the sources in your instance. This is a numeric value, not a string. The default value is **3 - Chat**. You can find your sources list in your instance under *Messenger* > *Settings* > *Sources*.
    - **server**: The HyperChat server URL assigned to your instance. Ask your Inbenta representative for this configuration parameter.
    - **server_port**: The port from which you can communicate with the HyperChat server. This is defined in your instance under *Messenger* > *Settings* > *Chat* > *Port*.
    - **queue**:
        - **active**: Enable or disable the queue system ("**true**" or "**false**"). You **MUST** enable it in your instance too. You do this from *Messenger* > *Settings* > *Chat* > *Queue mode*.
- **triesBeforeEscalation**: Maximum number of no-result answers in a row before the bot should escalate to an agent (if available). This is a numeric value, not a string. Set to zero to leave it disabled.
- **negativeRatingsBeforeEscalation**: Maximum number of negative content ratings in a row before the bot should escalate to an agent (if available). This is a numeric value, not a string. Set it to zero to leave it disabled.

**CONVERSATION** (conversation.php)

- **default**: Contains the API conversation configuration. The values are described below:
    - **answers**:
        - **sideBubbleAttributes**: Dynamic settings to show side-bubble content. Because there is no side-bubble in Microsoft Teams the value is an empty array (“**array()**”).
        - **answersAttributes**: Dynamic settings to show as bot answer. The defualt is [“ANSWER_TEXT”]. Setting multiple dynamic settings generates a bot answer with concatenated values with a newline character (\n).
        - **maxOptions**: Maximum number of options returned in a multiple-choice answer. The maximum number allowed value is 6 because Microsoft Teams limits the buttons to 6.
    - **forms**:
        - **allowUserToAbandonForm**: Whether or not a user is allowed to abandon the form after a number of consecutive failed answers. The default value is **true**.
        - **errorRetries**: The number of times a user can fail a form field before being asked if they want to leave the form. The default value is 3.
    - **lang**: Language of the bot, represented by its ISO 639-1 code. Accepted values: ca, de, en, es, fr, it, ja, ko, nl, pt, zh, ru, ar, hu, eu, ro, gl, da, sv, no, tr, cs, fi, pl, el, th, id, uk
- **user_type**: Profile identifier from the Chatbot App Knowledge Base. Minimum: 0. Default: 0. You can find your profile list in your Chatbot Instance under *Settings* > *User Types*.
- **source**: Source identifier (e.g. “teams”). You can use it to filter the logs in the dashboards.
- **content_ratings**:
    - **enabled**: Enable or disable the rating feature (“**true**” or “**false**”).
    - **ratings**: Array of options to display in order to rate the content. Every option has the following parameters:
        - **id**: Id of your content rating. You can find your content ratings in your Chatbot instance under *Settings* > *Ratings*. Remember that your rating type must be “**content**”.
        - **label**: Key of the label translation to display within the rating option button. You can configure available labels from **/lang/**. You can also modify them from your Inbenta App, as described in section **Create translations object in ExtraInfo (optional)**.
        - **comment**: If **true**, asks for a comment for the rating. This is useful when a user rates a content negatively to ask why the negative rating.
        - **isNegative**: Id **true**, the bot increments the negative-comments counter in order to escalate with an agent (if HyperChat **negativeRatingsBeforeEscalation** is configured).

    - **digester**:
        - **button_title**: The value of this dynamic setting shows as the content title in multiple-options buttons. This is useful to set an alternative title that has less characters to avoid cropping text.

**ENVIRONMENTS** (environments.php)

This file allows you to configure a rule to detect the current environment for the connector. It can check the current **http_host** or the **script_name** in order to detect the environment.

- **development**:
    - **type**: Detection type: check the **http_host** (e.g. www.example.com) or the **script_name** (e.g. /path/to/the/connector/server.php).
    - **regex**: Regex to match with the detection type (e.g. "/^dev.mydomain.com$/m" will set the "development" environment when the detection type is dev.example.com).
- **preproduction**:
    - **type**: Detection type: check the **http_host** (e.g. www.example.com) or the **script_name** (e.g. /path/to/the/connector/server.php).
    - **regex**: Regex to match with the detection type (e.g. "/^.*/staging/.*$/m" will set the "preproduction" environment when the detection type contains "/staging/").

## Deployment

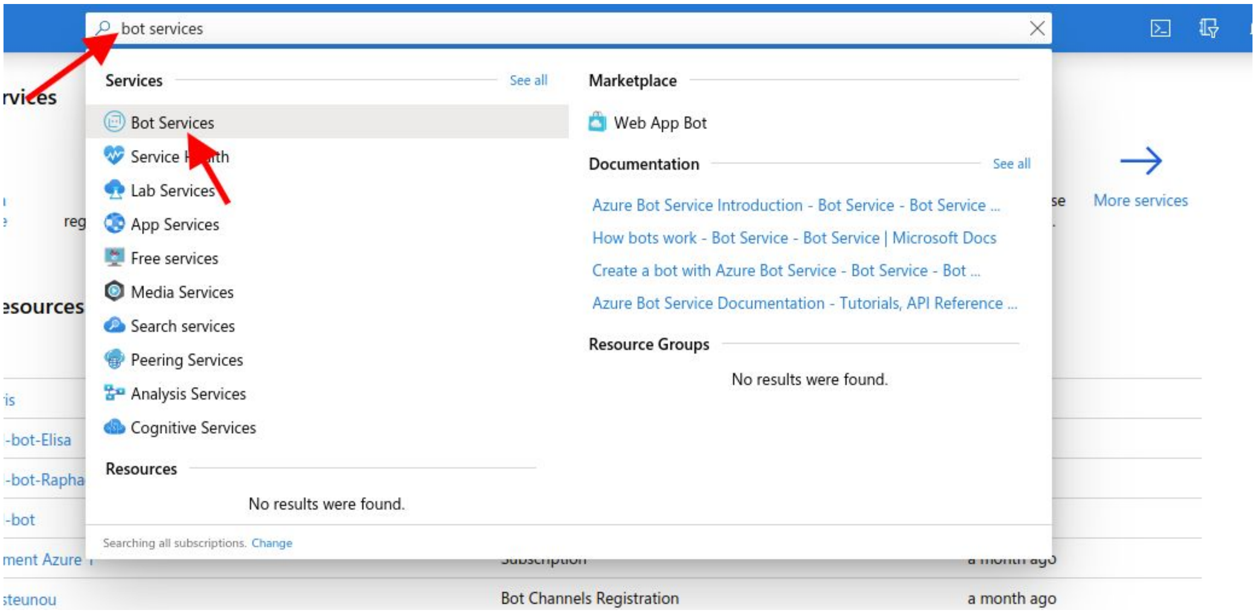
You must use a public webserver to serve the Microsoft Teams template. This makes it possible for Microsoft Teams to send events to it. The environment where the template was developed and tested has the following specifications

- Apache 2.4
- PHP 7.0+
- PHP Curl extension
- PHP DOM XML extension
- PHP JSON extension
- Non-CPU-bound
- The latest version of Composer (Dependency Manager for PHP), to install all dependencies that Inbenta requires for the integration.
- If the client has a **distributed infrastructure**, this means that multiple servers can manage the user session. They must adapt their SessionHandler so that the entire session is shared among all its servers.

# Prepare the Azure environment

## Create Bot Service and App Registration

Go to the Azure Portal and log in with an **Admin account**.
Search for “bot services” in the search bar and click on **Bot Services**.

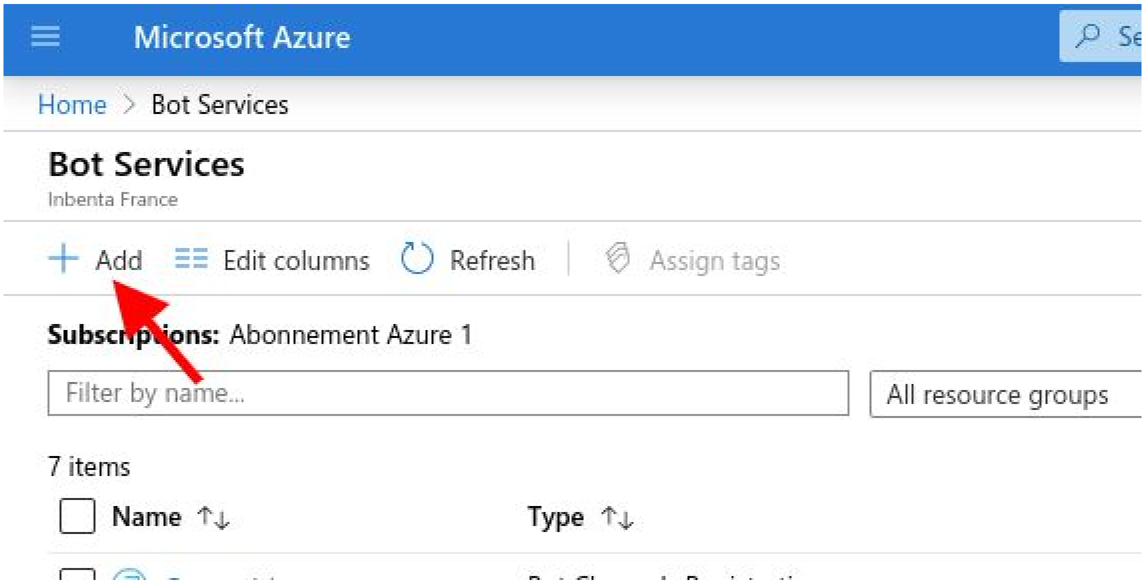


bot services

No results were found.

Searching all subscriptions. Change

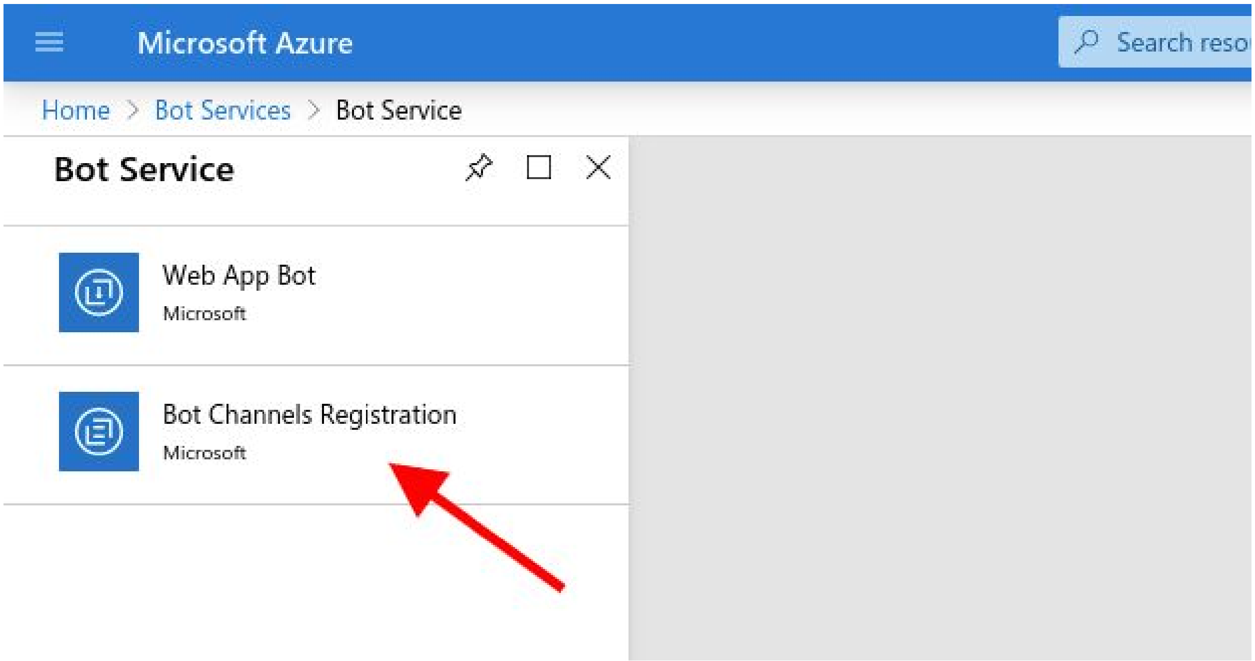
Click the **Add** button.



Subscriptions: Abonnement Azure 1

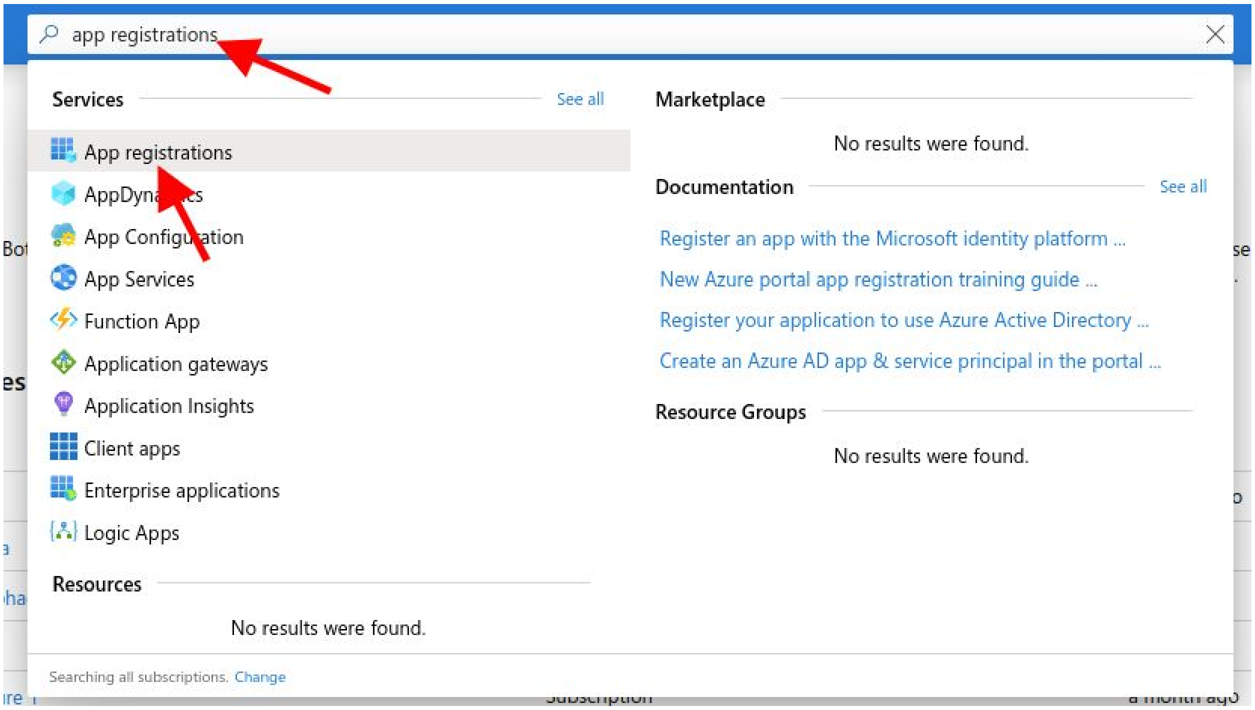
Choose **Bot Channels Registration** then the **Create** button, then fill all the necessary information, then submit the form by clicking on **create**.

- **Bot handle**: Name of the bot
- **Messaging endpoint**: Bot API endpoint

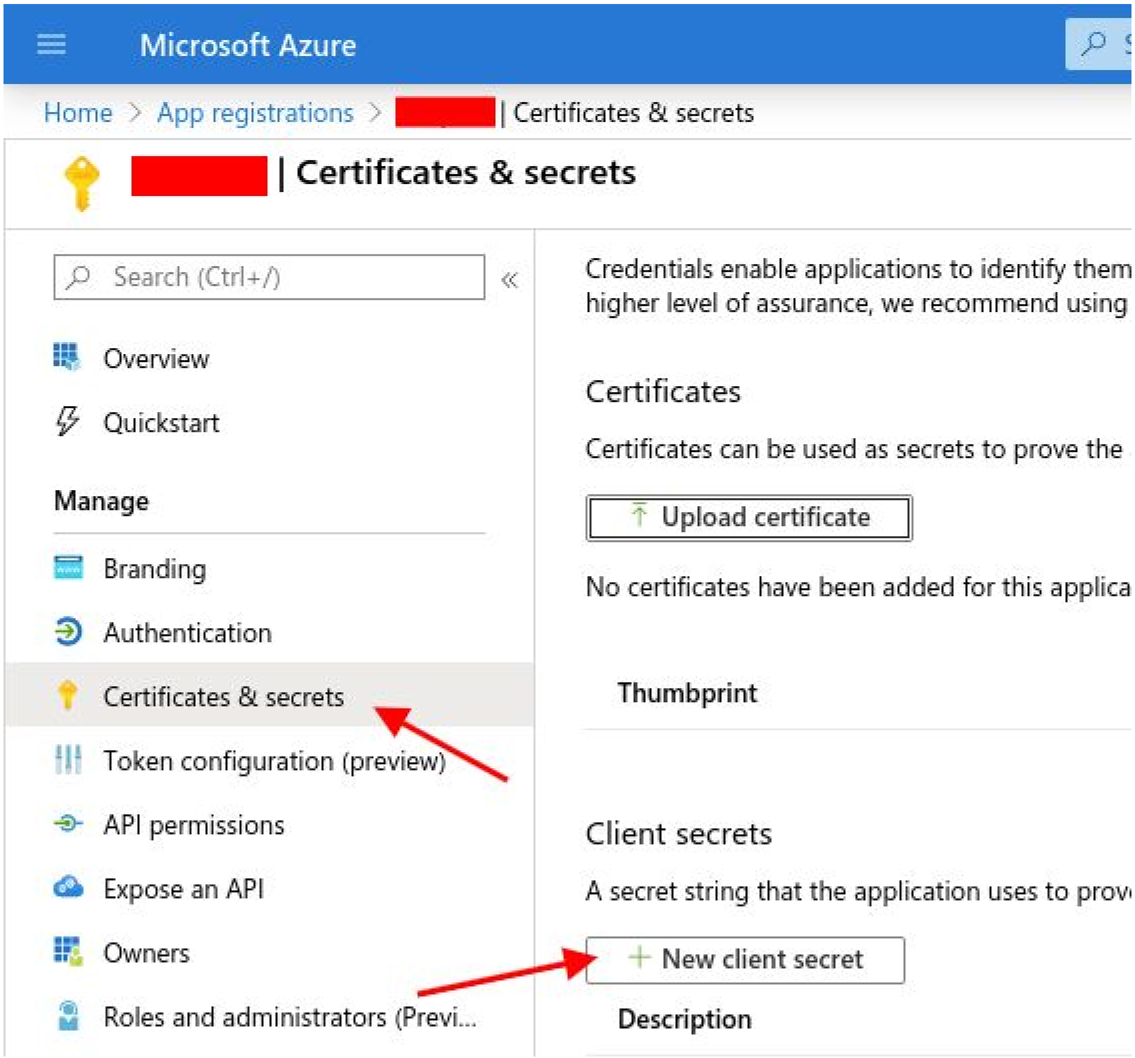


# Create an App Secret

Search for **App Registration** in the search input.



Select the previously created app then click on **Certificates & Secrets** on the left menu. Then click on **New client secret** and enter the **Description** the **Duration** and click on the **Add** button.



Credentials enable applications to identify them higher level of assurance, we recommend using

Certificates can be used as secrets to prove the

No certificates have been added for this applica

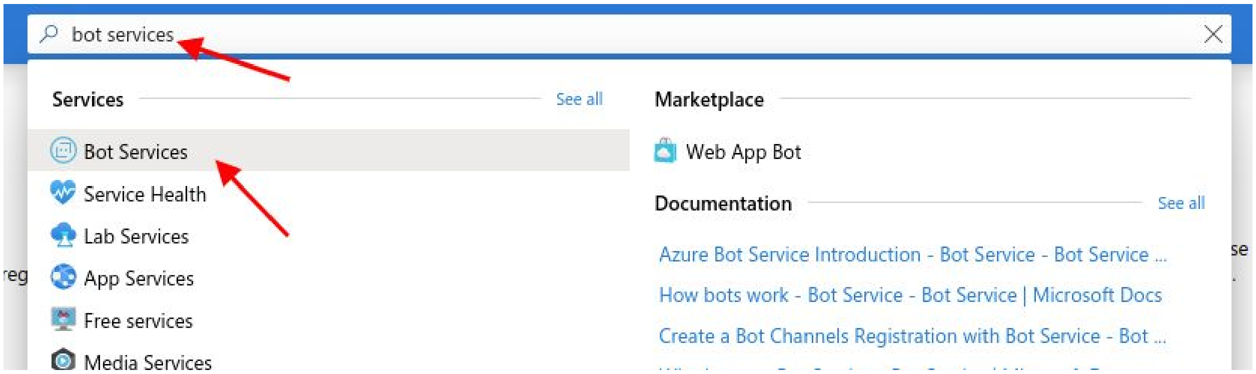
A secret string that the application uses to prov

Copy the **Secret** value generated and you can now add it to the **conf/custom/api.php** in the **microsoft_secret**.

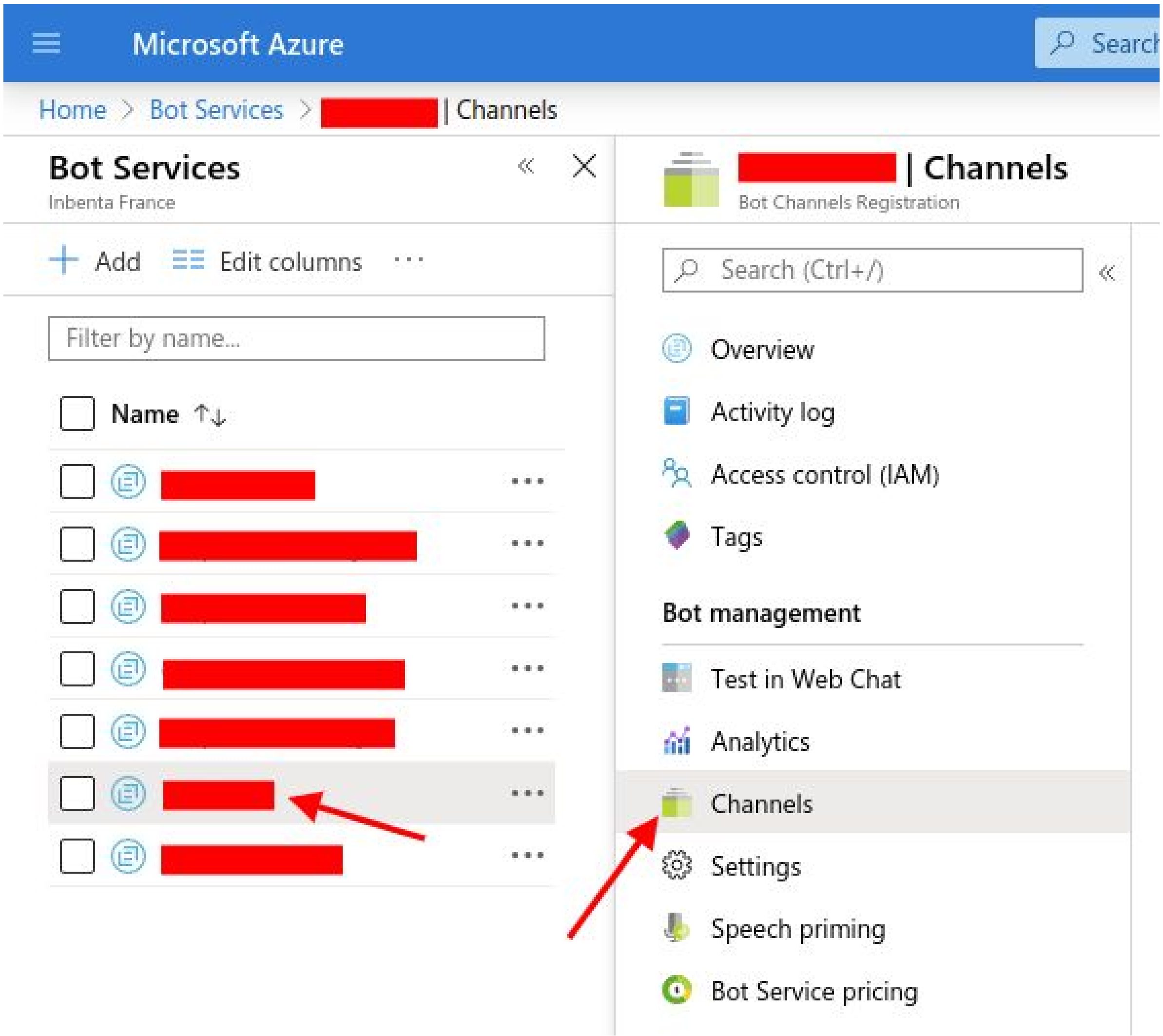
# Configure Azure Bot

## Configure Channels

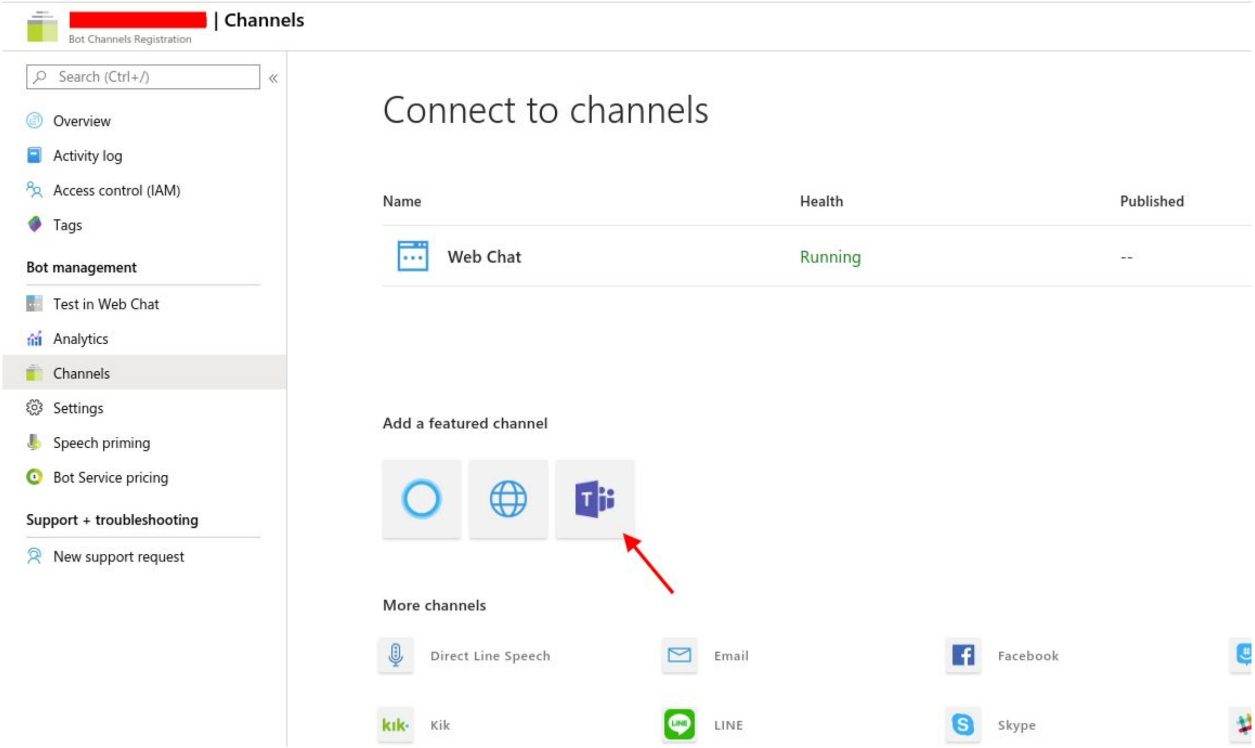
Search for **Bot Services** in the search input and select your bot.


bot services
Services
See all
Bot Services
Service Health
Lab Services
App Services
Free services
Media Services
Marketplace
Web App Bot
Documentation
See all
Azure Bot Service Introduction - Bot Service - Bot Service ...
How bots work - Bot Service - Bot Service | Microsoft Docs
Create a Bot Channels Registration with Bot Service - Bot ...

On the left menu after selecting your bot, click on **Channels**.



In the **Add a featured channel**, click on the **Teams** icon then on the **Save** button.



[redacted] | Channels

Bot Channels Registration

Connect to channels

| Name | Health | Published |
|---|---|---|
| Web Chat | Running | -- |

## Settings

In the **Bot Services** of your bot, click on **Settings**. Check the **Display name** along with the **Messaging endpoint** to be your server endpoint.

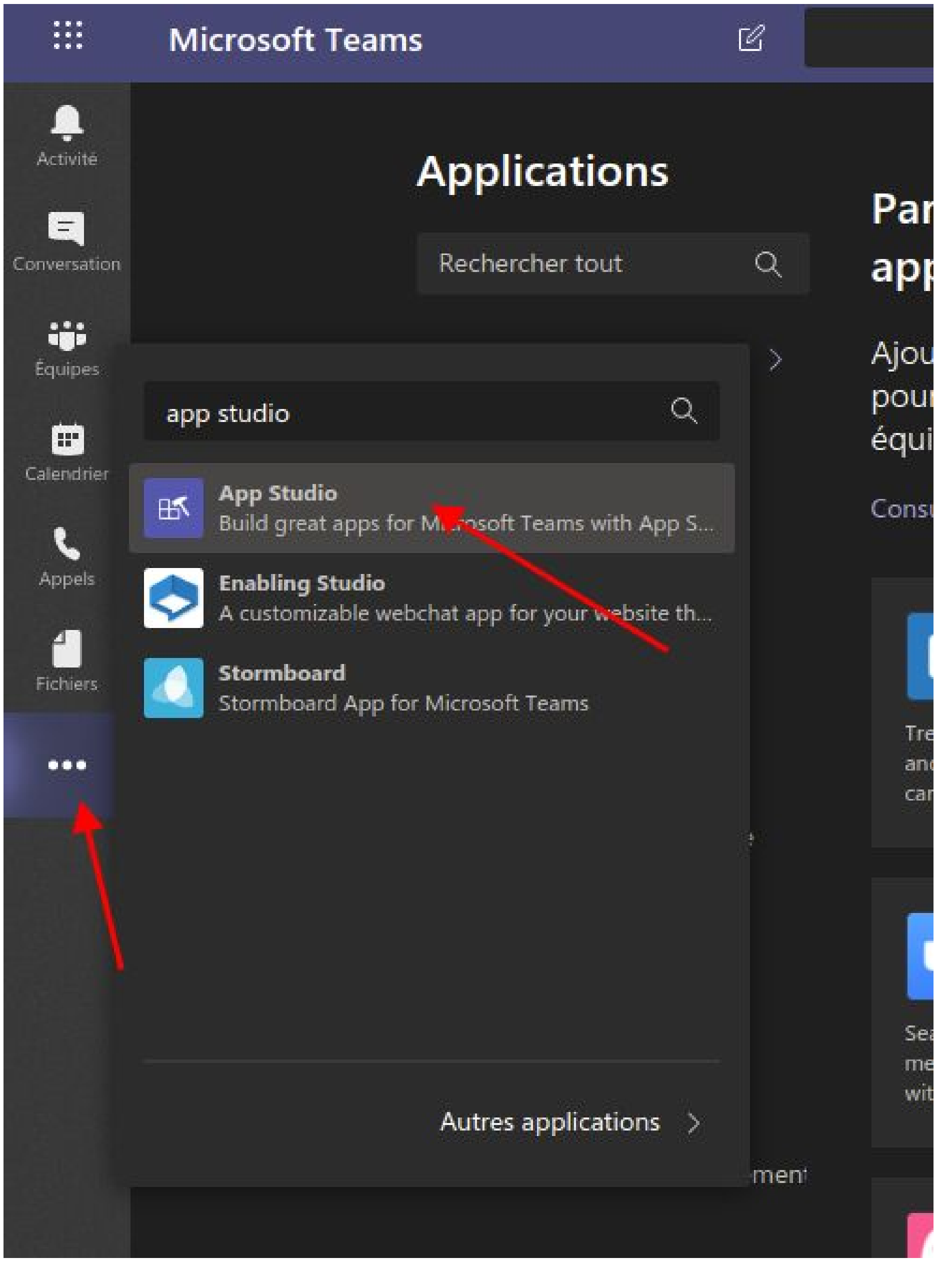
Copy the **Microsoft App ID** in the **conf/custom/api.php** in the **microsoft_appid**.

# Connect Azure App to Teams

First, download the **Teams** app from here or access the web app directly.

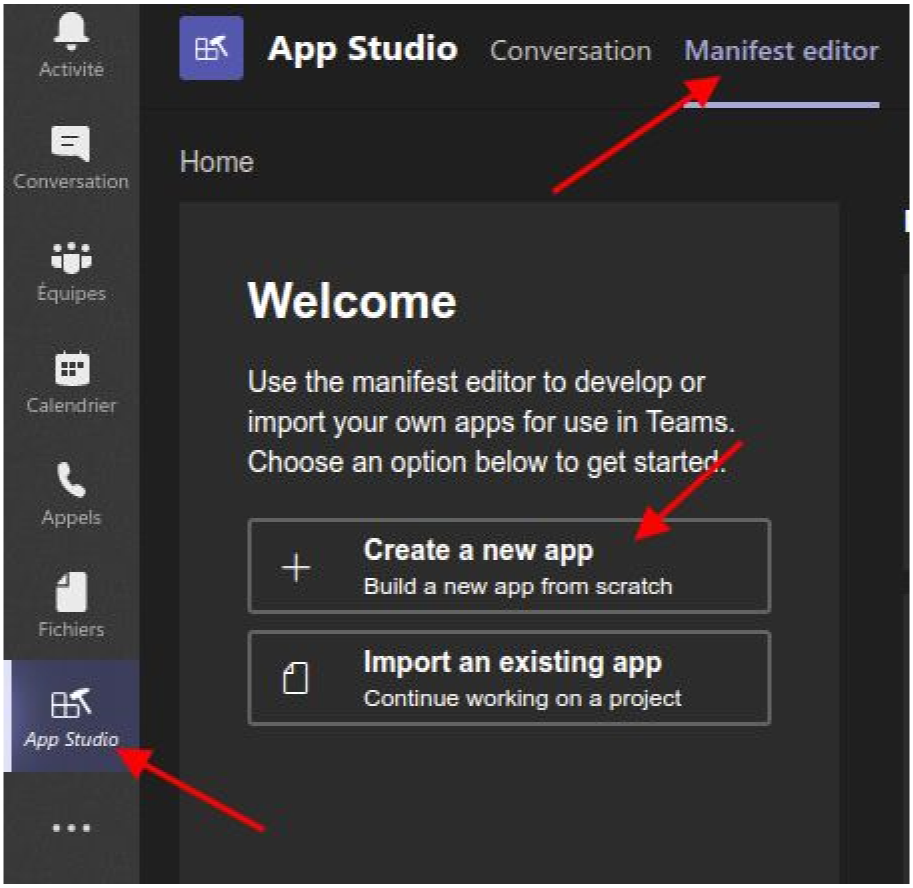
## Add App Studio

Click on the dots and search for **App Studio** and install it.



## Import App from Azure

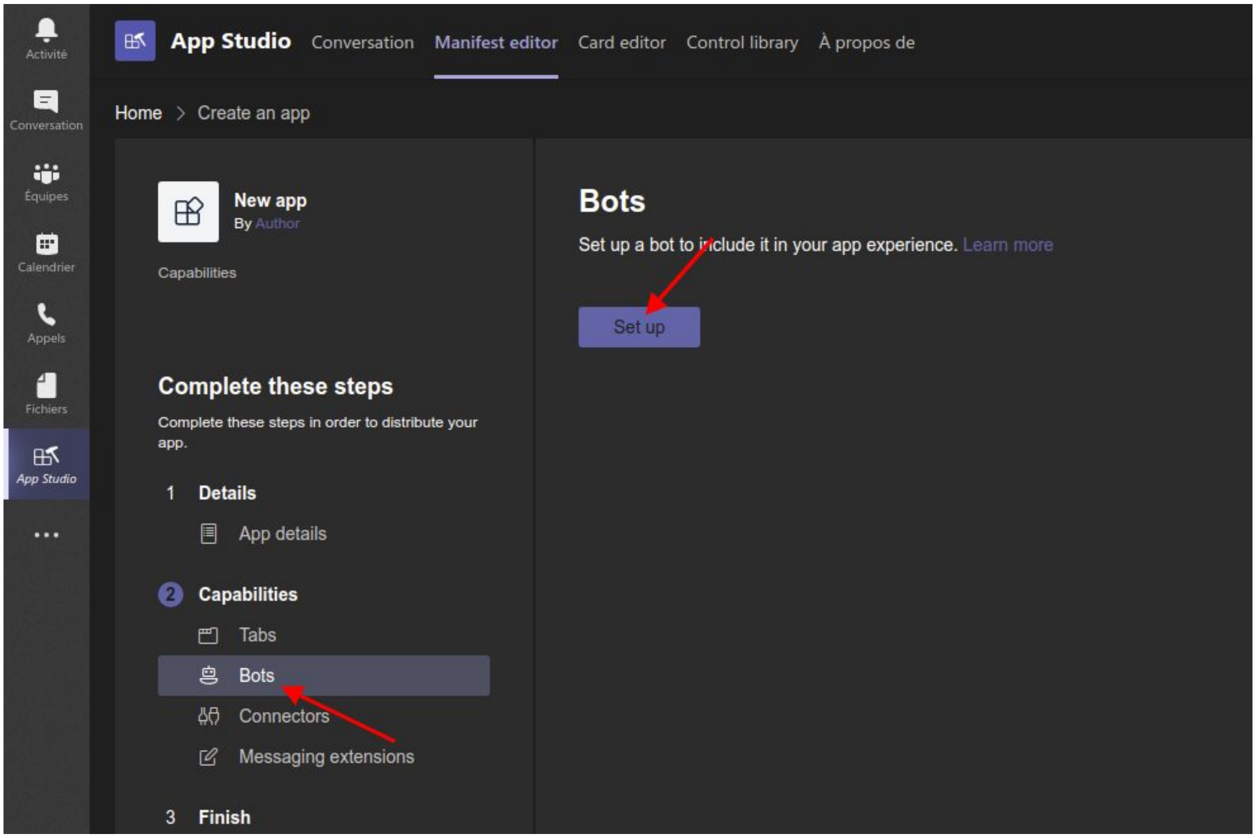
On the **App Studio**, select **Manifest Editor** tab and click on **Create a new app**.



Welcome

Use the manifest editor to develop or import your own apps for use in Teams. Choose an option below to get started.

Enter information about the app on the list of fields and in the left menu, click on the **Bots** tab in the second step, then on the **Set up** button.



Complete these steps in order to distribute your app.

Set up a bot to include it in your app experience. Learn more

In the new menu, select the **Existing Bot** tab, and fill information.

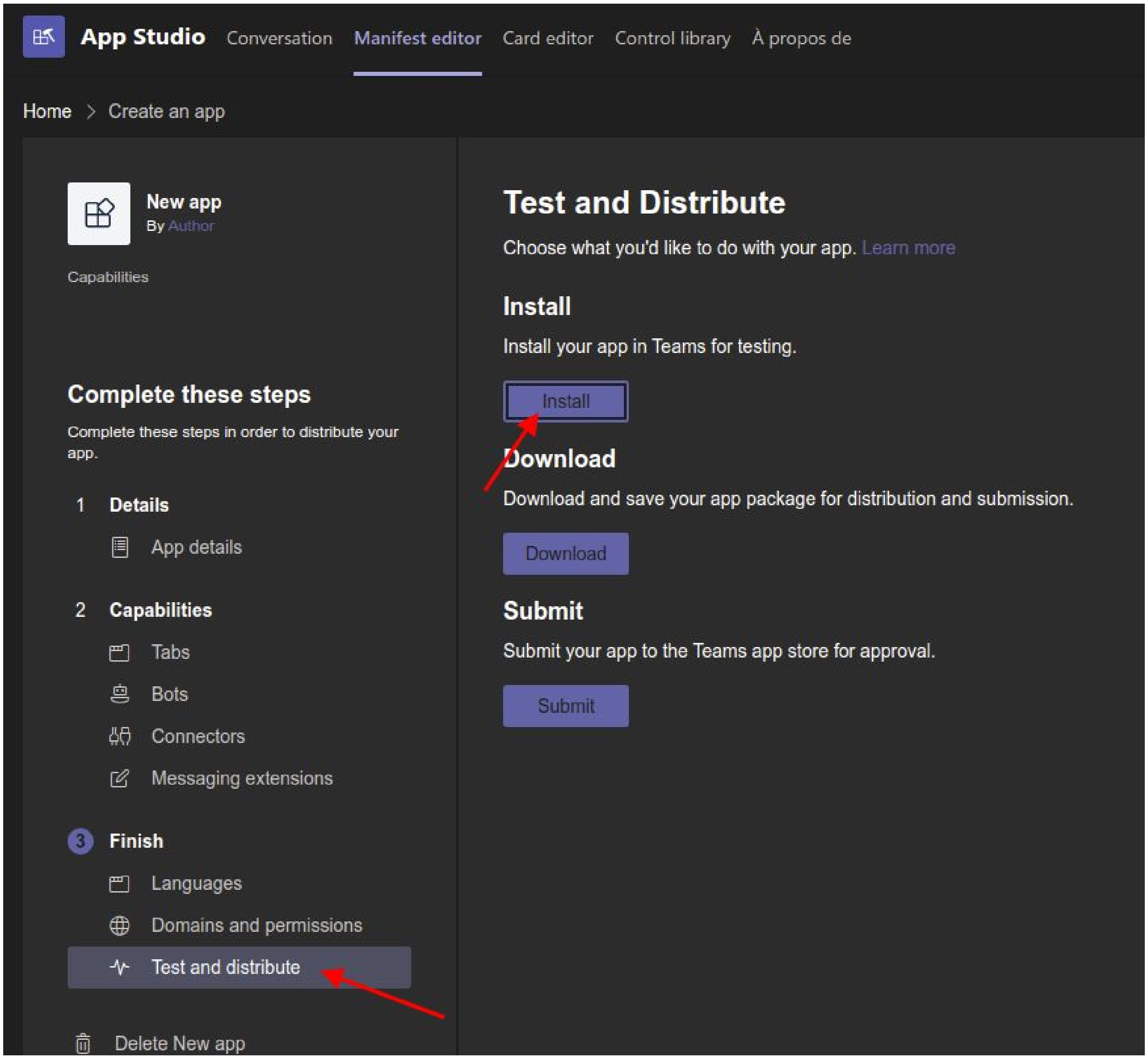
- **Bot ID**: Paste App ID from Azure (*Azure* > *App Registration* > *Select App* > *App Client ID*)
- **Messaging Bot**: Select ‘My bot supports uploading and downloading files’
- **Scope**: Select ‘Personal’, ‘Team’ and ‘Group Chat’

Click on the **Save** button.

In the left menu on the third step, click on **Test and distribute** then on the **Install** button.

Again click on the **Install** button. Open the dropdown menu and click on the arrow then select **Add to a team**.

Add the team you want your bot to appear. To finish it, click on **Set-up a bot**.



## Complete these steps

Complete these steps in order to distribute your app.

1 Details
- App details

2 Capabilities
- Tabs
- Bots
- Connectors
- Messaging extensions

3 Finish
- Languages
- Domains and permissions
- Test and distribute

## Test and Distribute

Choose what you'd like to do with your app. Learn more

### Install

Install your app in Teams for testing.

### Download

Download and save your app package for distribution and submission.

### Submit

Submit your app to the Teams app store for approval.

## Relevant URL’s

- Azure Portal <https://portal.azure.com/>
- Microsoft Teams Web App <https://teams.microsoft.com/#go>
- Microsoft Guide to Build a bot for Teams <https://docs.microsoft.com/en-us/microsoftteams/platform/bots/how-to/create-a-bot-for-teams>
- Microsoft Guide to Integrate a Bot on Teams <https://docs.microsoft.com/en-us/microsoftteams/platform/bots/how-to/add-power-virtual-agents-bot-to-teams>

# Troubleshooting

## Missing HyperChat messages

Check that your UI is subscribed to HyperChat webhooks, as described in **HyperChat integration (optional)**. You must also check if the HyperChat settings in the UI are valid in **conf/custom/chat.php**.

## The bot is not answering (azure configuration)

Check that you configured the URL of your UI correctly in the **Azure Bot Services**. You can check the current webhook URL in the **Bot Services** in the **Azure** portal under *Your Bot > Settings*. You must also check if the **microsoft_appid** and **microsoft_secret** set in **conf/custom/api.php** are valid.

## The bot is not answering (application cache)

If the previous tip does not bring expected results, maybe your application is caching older token values. If this is the case, delete the cached session files in your server. These files are stored in the configured system temporary path returned by the PHP function **sys_get_temp_dir()**. Usually, it’s “/tmp” or “/var/tmp” but may vary depending on your server system and configuration.

# How to test it?

Test your UI from the Teams web application or the downloaded application and click on the dots in the left and search the name of your bot.