

**inbenta**

# Skype Integration

<b>Features</b>	<b>3</b>
<b>Building a Skype app</b>	<b>4</b>
Setup chatbot Skype template	4
Step 1: download template	4
Step 2: setting a basic configuration	4
Create token objects in extraInfo	5
Step 1: create the app_id type	5
Step 2: create the app_password type	6
Step 3: Store the credentials	6
Create translations object in extraInfo (optional)	8
Azure environment	9
Create the Resource group	9
Create a Skype application	11
<b>How to test the bot?</b>	<b>14</b>
<b>How to publish the bot?</b>	<b>16</b>

# Features

These are the supported answer types and features:

- Simple answers
- Multiple options
- Polar questions
- Chained answers
- Content ratings (yes/no + comment)
- Custom FAQ title in the button when displaying multiple options
- Display a button to open a URL

Other important features:

- Escalate to HyperChat after a number of no-results answers
- Escalate to HyperChat when matching with an 'Escalation FAQ'
- Send information to webhook through forms
- Retrieve Skype tokens from ExtraInfo
- Create translations object in ExtraInfo

# Building a Skype app




## Setup chatbot Skype template

### Step 1: download template

You can find the PS Chatbot Smooch template in Git. Download it from [https://github.com/inbenta-integrations/skype\\_chatbot\\_template](https://github.com/inbenta-integrations/skype_chatbot_template) and move it into your infrastructure.

### Step 2: setting a basic configuration

Once downloaded, in the root directory you'll find a **conf** folder with the following structure:

-  **conf**
  -  **custom**
    - **api.php**
    - **environments.php**
  -  **default**
    - **api.php**
    - **chat.php**
    - **conversation.php**
    - **environments.php**
  - **configuration\_files.php**
  - **README.md**

Open the **README** doc if you want more details about how configuration works, but in summary:

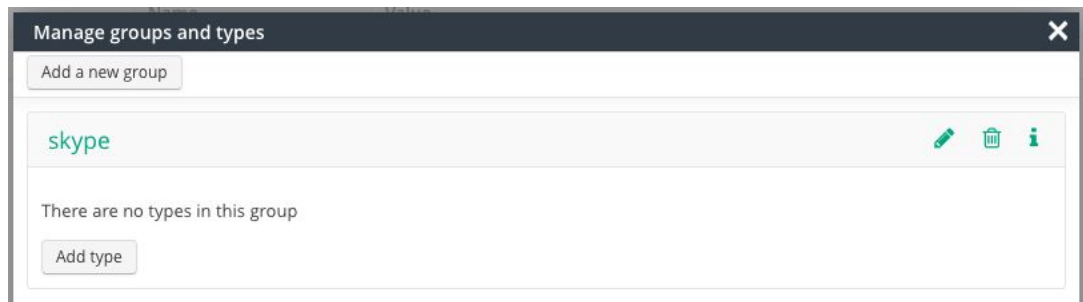
- The **default** folder contains all the settings. It is not necessary to complete all the files.
- The **custom** folder contains all the custom settings. Keep in mind, all settings defined here will override the default ones. It's good practice to describe all the project particularities here.
- If no environment is defined, the UI will assume the **production** env.
- If you want to modify several configuration parameters, copy the desired file from **conf/default** into **conf/custom** and modify the values.
- When filling in the **key** and the **secret** values, do not use the **default/api.php** file but the **custom/api.php** instead.

As we have external dependencies, remember to execute the command "composer install" in the root directory.

## Create token objects in extralInfo

You need to define the **app\_id** and the **app\_password** in order to successfully start the UI:

1. Log in to your Backstage instance
2. Go to Knowledge → Extra Info and click in **Manage groups and types**
3. Click on **Add a new group** and name it **skype**.



Now you have the skype group created, it's time to define the **app\_password** and **app\_id** types.

### Step 1: create the app\_id type

1. Click on the **Add type** button
2. Complete the form with the following values:
  1. Name: app\_id
  2. Group: skype
  3. You need to define one property called **value** and set its type to "text"
3. Click on the **OK** button once finished



## Step 2: create the app\_password type

1. Click on the **Add type** button
2. Complete the form with the following values:
  1. Name: app\_password
  2. Group: skype
  3. You need to define three properties: **development**, **preproduction** and **production** and set their type to "text"
3. Click on the **OK** button once finished

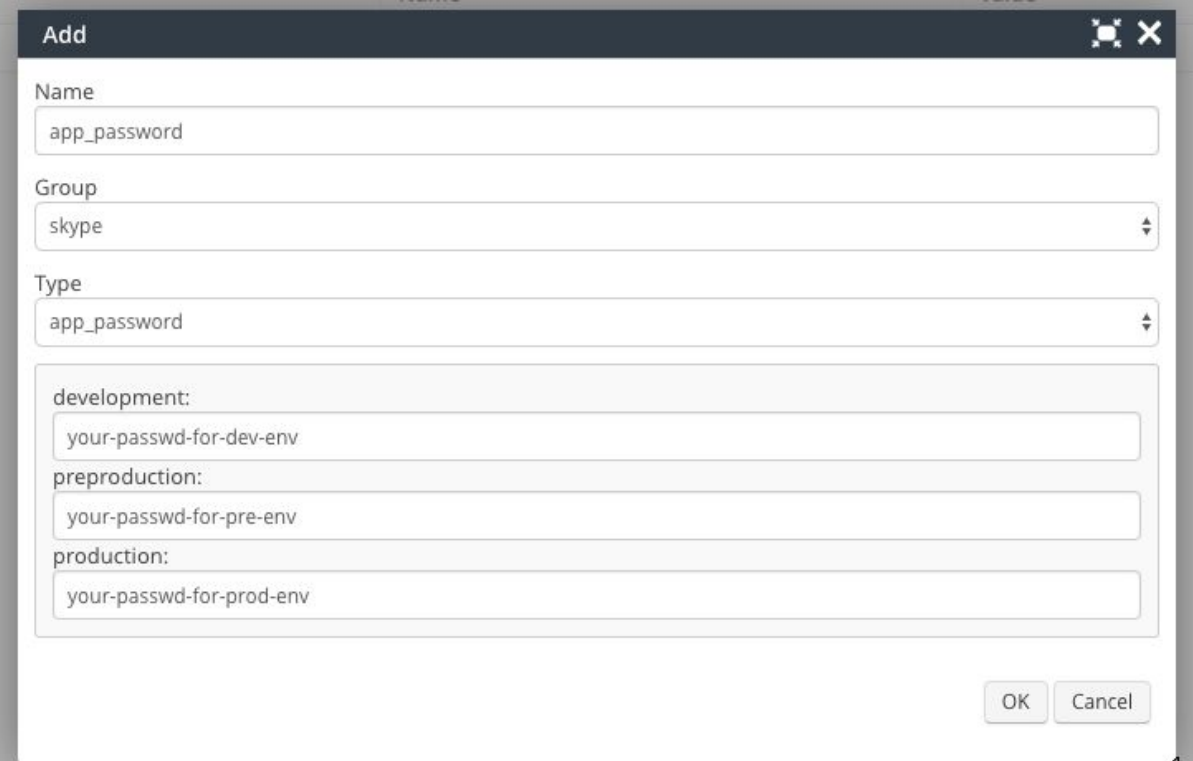


## Step 3: Store the credentials

1. Click on **New entry** (located at the left menu)
2. Select "skype" on the **Group** dropdown
3. Type "app\_id" in the **Name** field
4. Paste your skype app\_id in the **value** field
5. Click on the **OK** button



Repeat the steps in order to define the app\_password, but keep in mind you'll need to set the **Name** and the **Type** to "app\_password" and you'll need to provide one password for each environment:



The screenshot shows a dialog box titled "Add" with a dark header bar containing window control icons. The dialog has three main sections: "Name", "Group", and "Type". Each section has a text input field. Below these is a section for environment-specific passwords, with three sub-sections: "development:", "preproduction:", and "production:". Each sub-section has a text input field. At the bottom right, there are "OK" and "Cancel" buttons.

Field	Value
Name	app_password
Group	skype
Type	app_password
development:	your-passwd-for-dev-env
preproduction:	your-passwd-for-pre-env
production:	your-passwd-for-prod-env

You are done, don't forget to publish your changes by clicking on the **Post** button in the left menu.

The "app\_id" and the "app\_password" are given by the Azure website. See the section "**Create the Skype application**"

## Create translations object in extraInfo (optional)

You can manage the translation labels from Extra Info. Please keep in mind that labels defined in extraInfo will override the labels defined in the lang files.

1. Log in to your Backstage instance
2. Go to Knowledge → Extra Info and click on **Manage groups and types**
3. Select the **skype** group and click on the **Add type** button
4. Name it '**translations**' and add a new property with type 'Multiple' named with your chatbot's language label (en, es, it...).
5. Inside the language object, add all the labels that you want to override. Each label should be a 'text' type entry (you'll find the label list below).
6. Save your translations object.

Now you can create the ExtraInfo object by clicking the **New entry** button (as you did in the previous section), 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.

The screenshot shows a web form titled 'translations'. It has a 'Name' field containing 'translations'. Below it is a 'Value' section. Inside 'Value', there is a label 'en:' followed by a list of labels and their corresponding English values. The labels are: 'agent\_joined' with value '(EI) Agent \$agentName has joined the conversation.', 'api\_timeout' with value '(EI) Please, reformulate your question.', 'ask\_rating\_comment' with value '(EI) Please tell us why', 'ask\_to\_escalate' with value '(EI) Do you want to start a chat with a human agent?', and 'chat\_closed' with value '(EI) Chat closed'. Each label has a text input field next to it. There is a trash icon next to the list of labels.

Here you have 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?'
- **thanks** => 'Thanks!'
- **yes** => 'Yes'

## Azure environment

Like other platforms (such as Facebook, Twitter, Instagram, ...), it is necessary to register a new application to have a bot in the Skype environment. All the configuration must be done through the [Azure portal](#), so it is necessary to have an account.

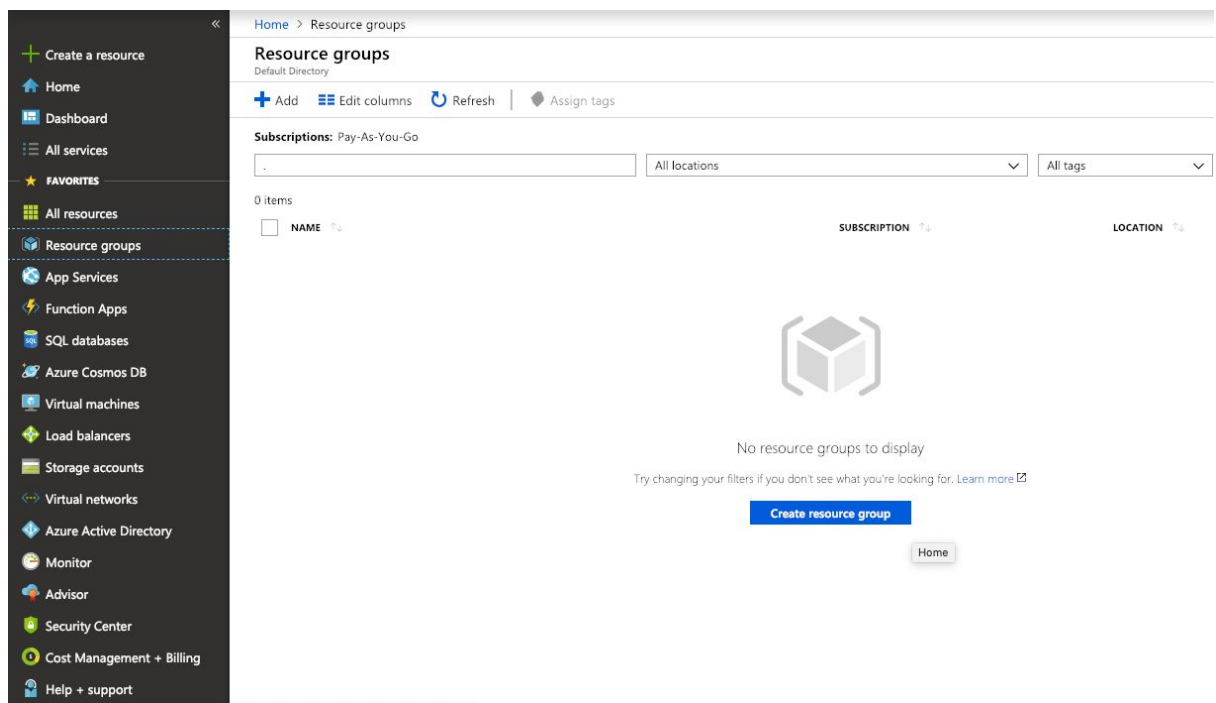
Once inside, there are two steps:

1. Create a *Resource group*.
2. Create the application itself.

## Create the Resource group

Before creating the application, a container must be created for them. In Azure, these containers are called "**Resource groups**" and the following are the steps to create one:

1. Click on the section "**Resource groups**" and click on **[+]Add**.



2. Fill in all of the required information and click on **"Review + Create"**.

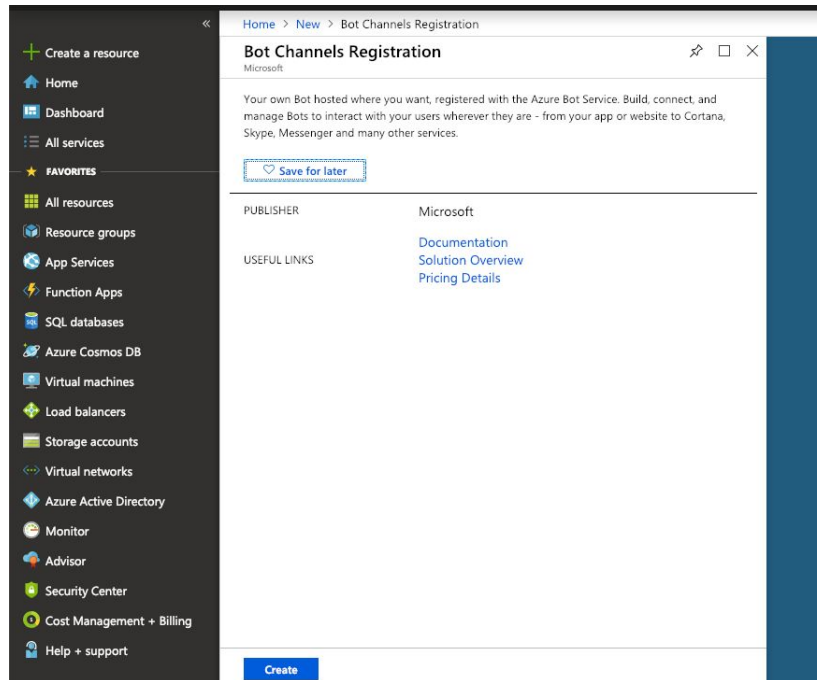
The screenshot displays the Azure portal interface for creating a new resource group. On the left, a navigation sidebar lists various Azure services, with 'Resource groups' highlighted. The main content area is divided into two sections. The left section, titled 'Resource groups', shows a table with the header 'NAME' and a message indicating that no resource groups are currently displayed, accompanied by a 'Create resource group' button. The right section, titled 'Create a resource group', contains a form with three tabs: 'Basics', 'Tags', and 'Review + Create'. The 'Basics' tab is selected, showing 'PROJECT DETAILS' with fields for 'Subscription' (set to 'Pay-As-You-Go') and 'Resource group' (set to 'resource-group-name'). Below this, the 'RESOURCE DETAILS' section shows the 'Region' set to 'West Europe'. At the bottom of the right pane, there are two buttons: 'Review + Create' and 'Next : Tags'.

If a subscription is required for one of the processes, the best option is **"Pay-As-You-Go"** since it is not necessary to host the application. Of course, other subscriptions can be used to register as well.

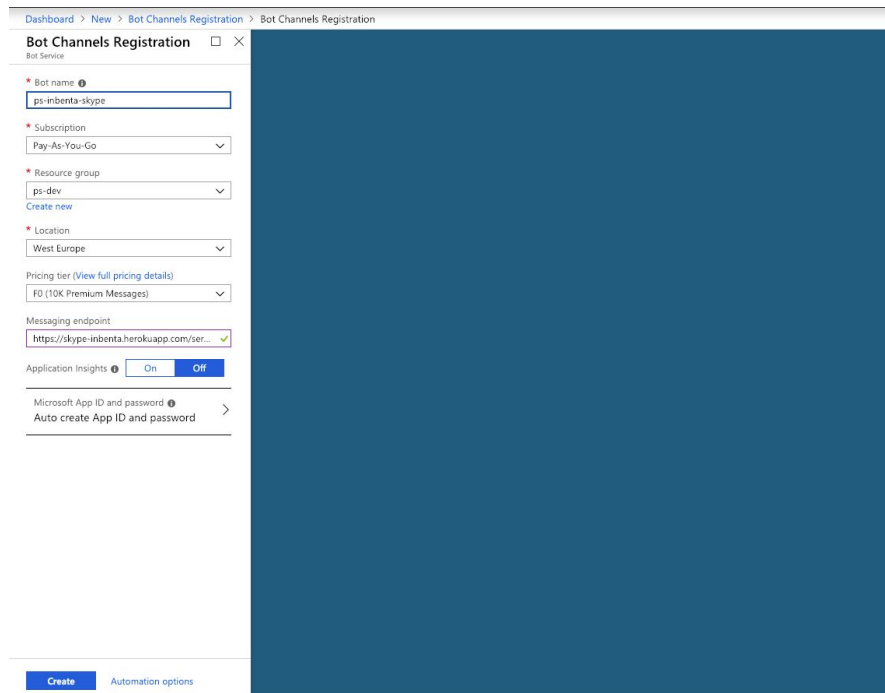
## Create a Skype application

Once the resource group has been created, the next step is to create the application.

1. Click on the sidebar button named "[+]Create a resource" and search for "Bot channel Registration". Then click on "Create".



2. Fill all the information to create the channel and then click on "Create". Below you will find the description of each field.



- **Bot name** => his name is useful for identification inside the Azure portal.
  - **Subscription** => select the right subscription. The recommendation is to always use Pay-As-You-Go.
  - **Resource group** => select the group created beforehand.
  - **Location** => select the nearest area where the user will be using the bot.
  - **Pricing tier** => select the free tier "F0".
  - **Messaging endpoint** => the endpoint where the application is located. At this time, the application must already return a "200 OK" response.
  - **Application Insights** => This is not necessary, select "Off".
3. Once the application is created, the information related to the ID of the application and the Secret that must be configured in the extraInfo variables can be found in the section called "Configuration".

**ps-inbenta-skype - Settings**  
Bot Channels Registration

Search (Ctrl+/) Save Discard

**Bot profile**

Icon  
Upload custom icon 30K max, png only

\* Display name ⓘ  
ps-inbenta-skype

Bot handle ⓘ  
ps-inbenta-skype

Description ⓘ

**Configuration**

Messaging endpoint  
[Redacted]

\* Microsoft App ID [Manage](#) ⓘ  
[Redacted]

**Analytics**

Application Insights Instrumentation key ⓘ  
[Redacted]

Application Insights API key ⓘ  
.....

Application Insights Application ID ⓘ  
Application ID (Application Insights Application ID)

The relevant fields for the application configuration are:

- **Microsoft App ID:** the identifier of the application. Must be put it in the extra info field called "**app\_id**"

- **Secret:** to obtain the secret, click on the link "Managed" (*next to App ID field*) and create a new password. It is important to save it because nowadays it is not possible to recover it. Also, the secret should be added to the *extraInfo* field named "app\_password".

## ps-inbenta-skype Registration

[Click here for help integrating your application with Microsoft.](#)

### Properties

Name

ps-inbenta-skype

Application Id

49a6[REDACTED]

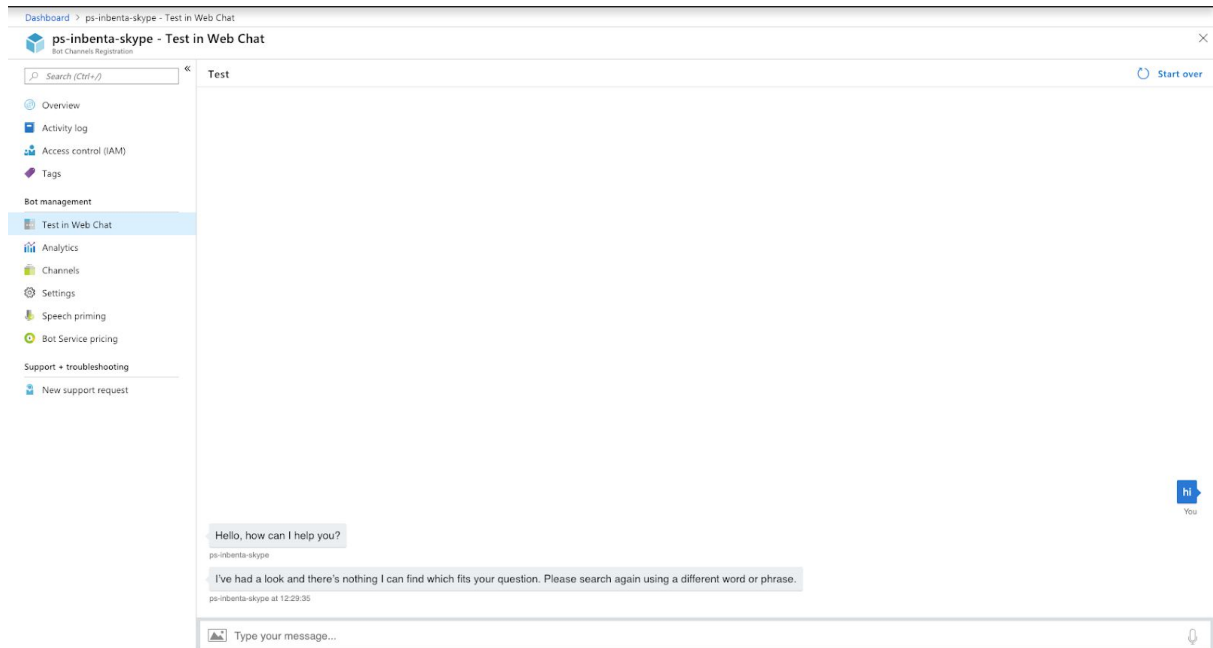
### Application Secrets

[Generate New Password](#) [Generate New Key Pair](#) [Upload Public Key](#)

Type	Password/Public Key	Created	
Password	sep*****	Jan 25, 2019 12:24:38 PM	<a href="#">Delete</a>
Password	qV*****	Jan 25, 2019 12:09:30 PM	<a href="#">Delete</a>

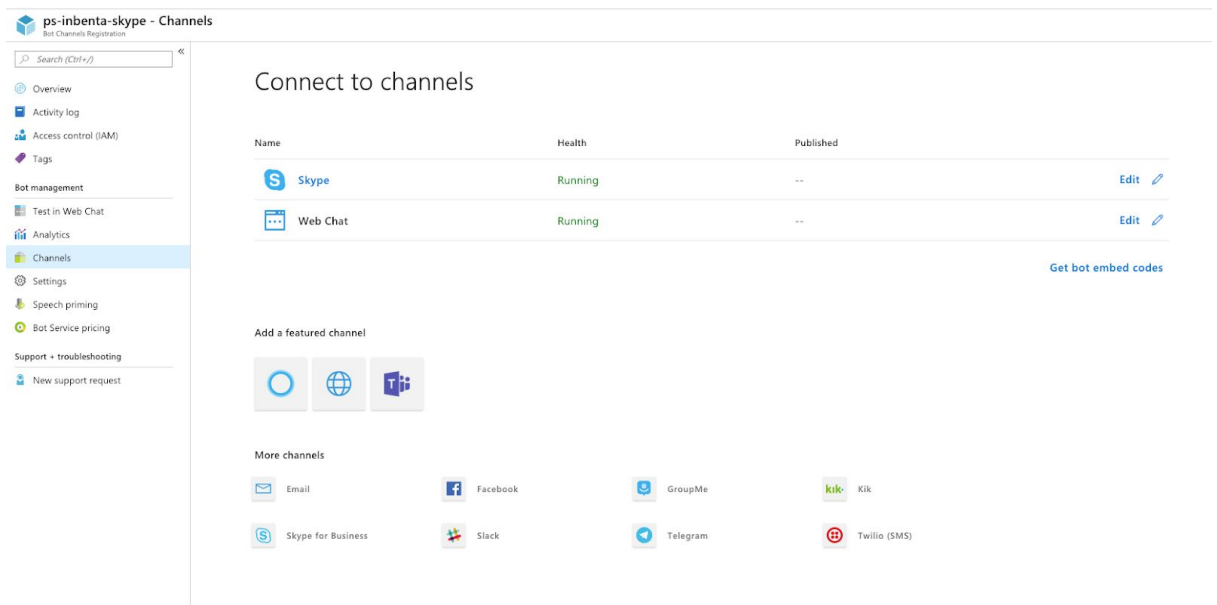
# How to test the bot?

There are several ways to test the interaction between the application and the integration of the bot. Azure provides one in which the user can test if everything works as expected. The section is **"Test in web chat"**.

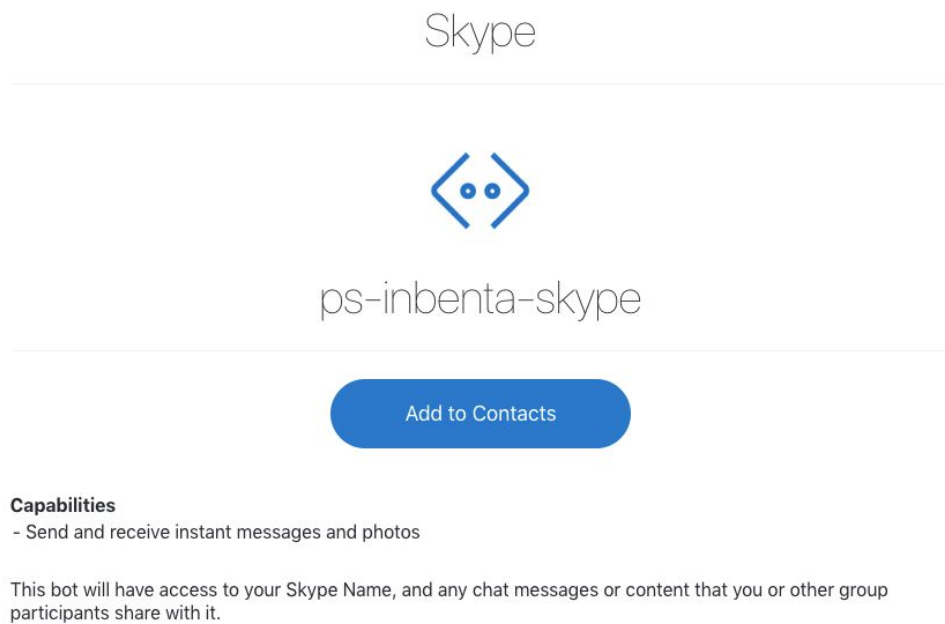


Additionally, the bot can be tested in the Skype Application. By default, when a "Bot Channel" is created, the [Skype platform](#) is added as well. To make it accessible the steps are:

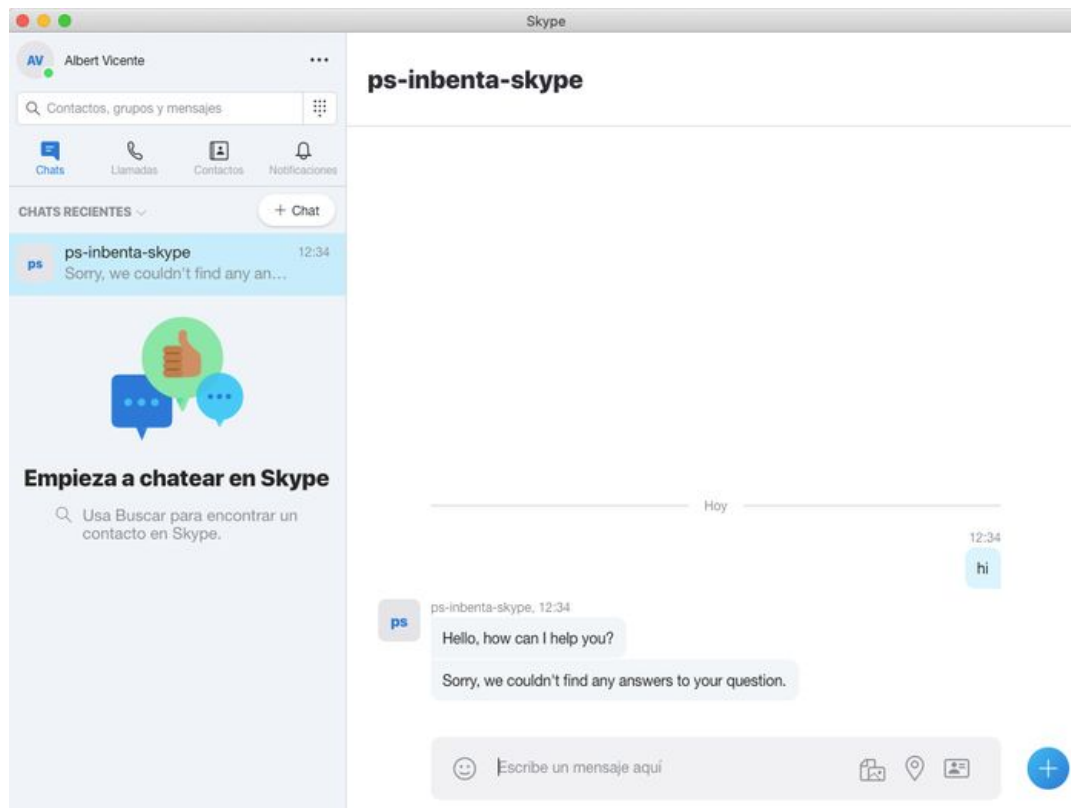
1. First of all, open a new tab and log in to the [Skype website](#).
2. Once inside, go to the Azure portal and click on the section called "Channels".



3. Then, click on the platform "**Skype**" and a new tab will be opened automatically.



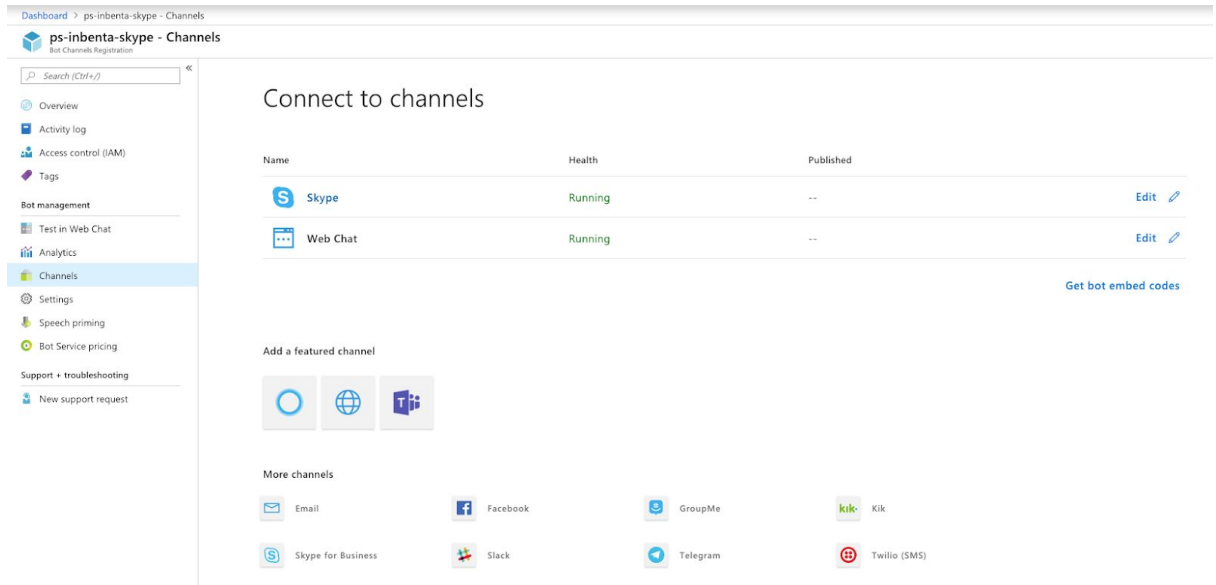
4. Click on "Add to Contacts" and the bot will be able to start a conversation with it.



## How to publish the bot?

At this point, the bot will not be visible to all Skype users, just for those people that have added the bot as a contact. To make it public, the publishing process has to be applied. The steps are:

1. Go to the section called Channels, inside the Bot Channel.
2. Click on the edit button for the Skype channel.



3. In the section "Publish", fill in all requested information and click on the button "Submit for Review". Before doing so, ensure that the bot application follows the criteria defined [here](#).

