inbenta

Twilio Integration

Index

Introduction

The purpose of this document is to define the integration of Inbenta's Chatbot and Hyperchat solution with Twilio, for Whatsapp and SMS.

Features

These are the supported answer types and features:

- FAQ Intents with images, videos and Hyperlinks as external links.
- Related Contents.
- Multiple options.
- Polar Questions.
- Dialogs.
- Forms, Actions & Variables.
- Hyperchat Integration.

Other important features to note:

- HyperChat escalation after X no-results answers (triesBeforeEscalation)
- Escalate to HyperChat when a FAQ matched has directCall set to 'escalationStart'.
- Retrieve translations from ExtraInfo

NOTE: WhatsApp / SMS doesn't allow displaying buttons, so **Multiple options**, **Polar questions**, **ratings** and **escalation offer** will be asked without buttons, the options will be numbered and user will enter the number of their choice.

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 Backstage instance, go to Knowledge → Extra Info and click on 'Manage groups and types' → twilio→ Add type. Name it '**translations**' and add a new property with type '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.

Now you can 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.

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'
- **queue_estimation_first** => ‘There is one person ahead of you.’
- **queue_estimation** => ‘There are $queuePosition people ahead of you.’
- **rate_content_intro** => 'Was this answer helpful?'
- **thanks** => 'Thanks!'
- **yes** => 'Yes'
- **ticket_response_intro** => "Hi, I'm the agent that you spoke to a while ago. My response to your question"
- **ticket_response_info** => "Here is the ticket number for your reference"

- **ticket_response_end** => "You can now continue chatting with the chatbot. If you want to talk to someone, type 'agent'. Thank you!"

Information
Remember to publish your ExtraInfo changes by clicking the ‘Post’ button.

Even if you have created the ExtraInfo translation, it is mandatory that the lang file exists in the “lang” folder.

HyperChat integration (optional)

If you use HyperChat you must subscribe your UI to the Hyperchat events. Open your *Messenger* instance. Go to Messenger → Settings → Chat → Webhooks. Here, in the ‘Events’ column type “queues:update,invitations:new,invitations:accept,forever:alone,chats:close, messages:new,users:activity”. In the ‘Target’ column paste your UI’s URL, then click on the ‘+’ button on the right.

Building the Twilio Connector

Setup Chatbot Twilio Connector

Download Twilio Connector

You can download the Chatbot Twilio Connector from GitHub. You can download or clone the connector from https://github.com/inbenta-integrations/twilio-chatbot

Required Configuration

In your UI directory, go to **conf**. Here, you have a readme file with some structure and usage explanations. If you only want to build a chatbot, fill the **key** and **secret** values inside the **conf/custom/api.php** file with your Inbenta Chatbot API credentials. 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

There are some optional features that can be enabled from the configuration files. Every optional configuration file should be copied from **/conf/default** and store the custom version in **/conf/custom**. The bot will detect the customization and it will load the customized version. These are the optional configuration files and the configuration fields description.

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 instance in which the chat opens. You can find it in your instance → Messenger → Settings → Chat.
    - **secret**: Your HyperChat instance application secret. You can find it in your instance → Messenger → Settings → Chat.
    - **roomId**: The room where the chat opens. This is mapped directly to a Backstage queue ID. Numeric value, not a string. You can find your rooms list it in your instance → 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. Numeric value, not a string. The default value is **3 - Chat**. You can find your sources list it in your instance → Messenger → Settings → Sources.
    - **regionServer**: The geographical region where the HyperChat app lives.
    - **server**: The Hyperchat server URL assigned to your instance. Ask your Inbenta contact for this configuration parameter.
    - **server_port**: The port where to communicate with the Hyperchat server. It’s defined in your instance → Messenger → Settings → Chat -->Port
    - **queue**:
        - **active**: Enable or disable the queue system (“**true**” or “**false**”). It **MUST** be enabled in your instance too (Messenger → Settings → Chat → Queue mode).
- **triesBeforeEscalation**: Number of no-result answers in a row after the bot should escalate to an agent (if available). Numeric value, not a string. Zero means it’s disabled.
- **negativeRatingsBeforeEscalation**: Number of negative content ratings in a row after the bot should escalate to an agent (if available). Numeric value, not a string. Zero means it’s disabled.
- **messenger**: Setting that allow replying to tickets after the agent conversation is closed.
    - **key**: API Key of the Messenger Instance. You can find it in Administration → API → [Production | Development].

- **secret**: Secret Key token of the Messenger Instance. You can find it in Administration → API → [Production | Development].
- **webhook_secret**: Secret token, defined when the configuration of External Ticket Source is made.

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 Twilio the content is shown after the main answer.
        - **answerAttributes:** Dynamic settings to show as the bot answer. The default 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.
    - **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 he wants 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 Backstage knowledge base. Minimum:0. Default:0. You can find your profile list in your Chatbot Instance → Settings → User Types.
- **source**: Source identifier (e.g. “twilio”) used 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 → Settings → Ratings. Remember that your rating type should be "**content**".

        - **label:** Key of the label translation to display within the rating option button. The available labels can be configured from **/lang/**. Also can be modified from Backstage as described in section **Create translations object in ExtraInfo (optional)**
        - **comment:** If **true**, asks for a comment for the rating. It's useful when a user rates a content negatively in order to ask why the negative rating.
        - **isNegative:** If **true**, the bot will increment the negative-comments counter in order to escalate with an agent (if HyperChat **negativeRatingsBeforeEscalation** is configured).

ENVIRONMENTS (environments.php)

This file allows configuring 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

The Twilio template must be served by a public web server in order to allow Twilio to send the events to it. The environment where the template has been developed and tested has the following specifications

- Apache 2.4
- PHP 7.3
- PHP Curl 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 Twilio environment

Create a Twilio Account

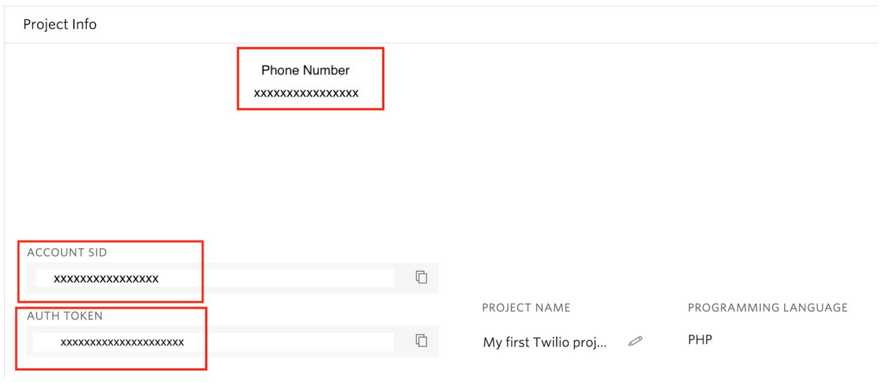
You need a Twilio Account in order to start with the implementation.

Login to your Twilio Account, or, if you do not have one, create an account following the self-explanatory instructions.

Get a phone number and SID / Auth Token.

Go to **Console Dashboard**. You can find your Phone Number, Account SID and Auth Token.

Twilio Dashboard



Project Info

Phone Number
xxxxxxxxxxxxxxxx

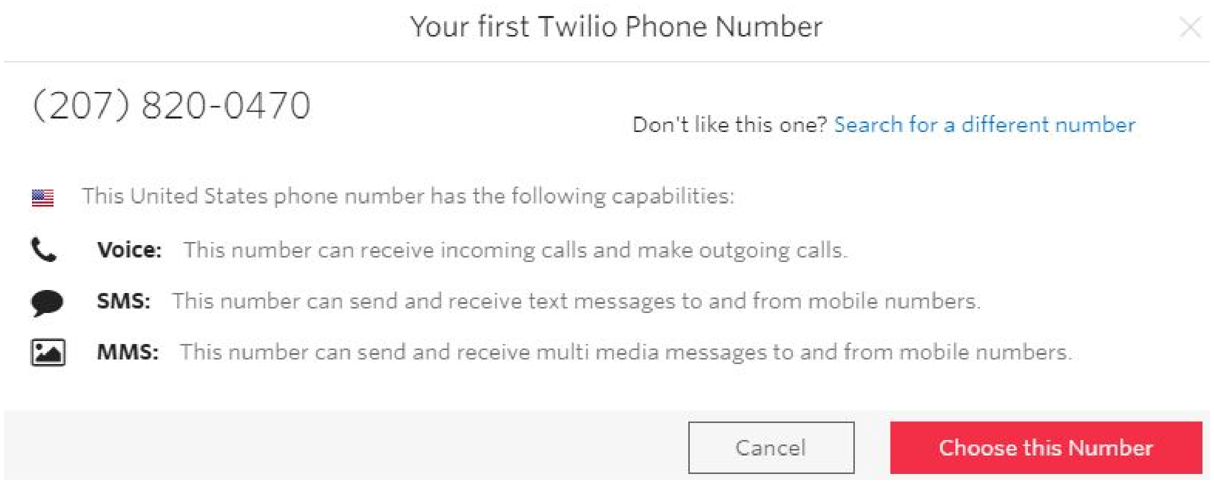
ACCOUNT SID
xxxxxxxxxxxxxxxx

AUTH TOKEN
xxxxxxxxxxxxxxxxxxxx

PROJECT NAME
My first Twilio proj...

PROGRAMMING LANGUAGE
PHP

If you are a new user, Click on “ get your phone number:



Your first Twilio Phone Number

(207) 820-0470

Don't like this one? Search for a different number

This United States phone number has the following capabilities:

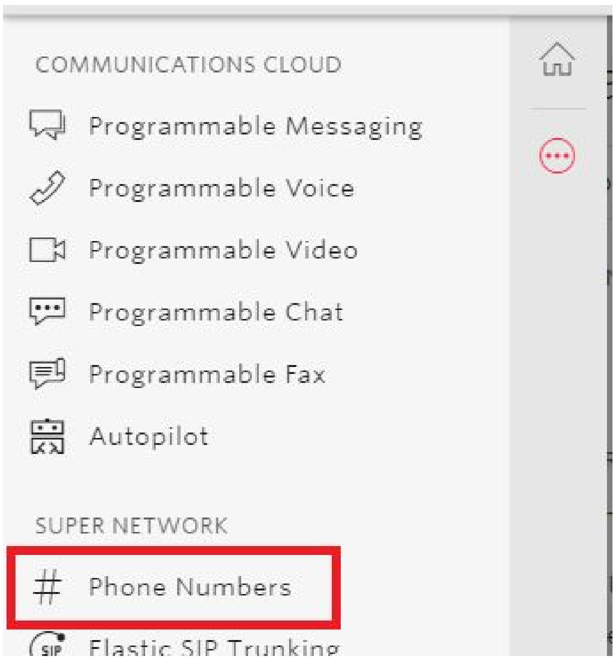
Voice: This number can receive incoming calls and make outgoing calls.

SMS: This number can send and receive text messages to and from mobile numbers.

MMS: This number can send and receive multi media messages to and from mobile numbers.

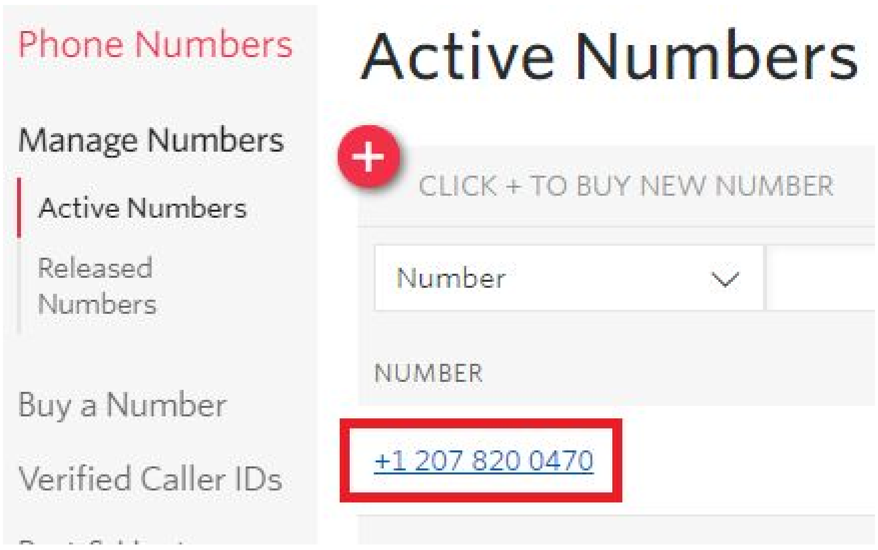
Set the environment for SMS

For set the SMS environment, press the three dots icon (All products & Services) and choose “Phone Numbers”



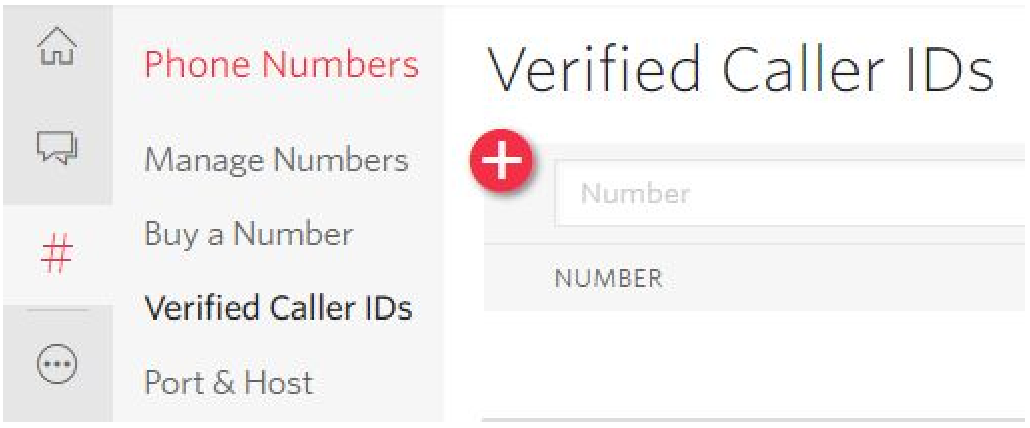
Trial Number

In the option "Manage Numbers → Active Numbers", press on the previously getted trial phone number:

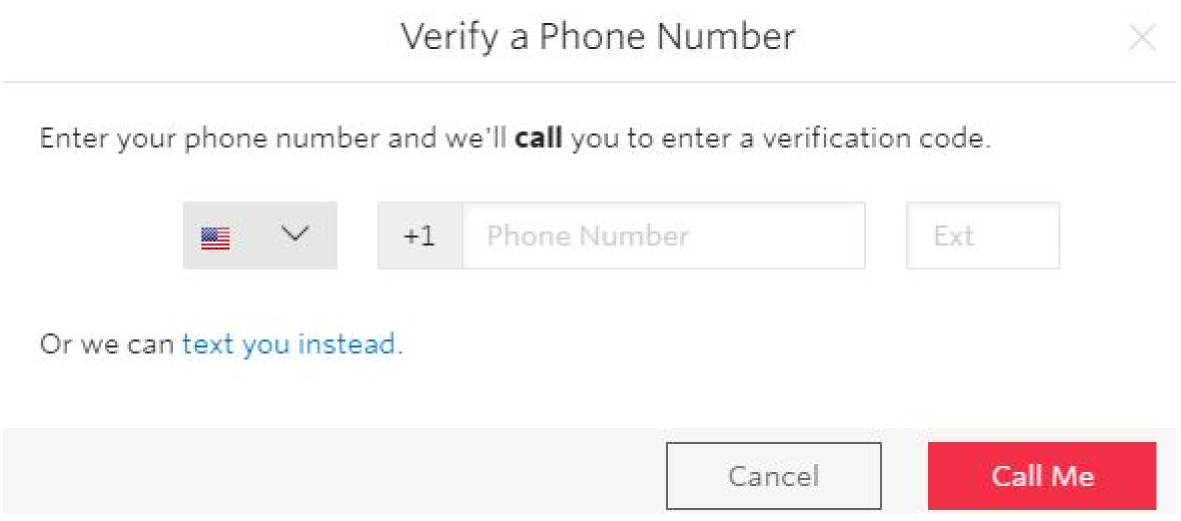


Add Verified Caller ID

In "Phone Numbers → Verified Caller IDs" you need to add a valid phone number to start sending messages to the Chatbot. Press the "+" red button



And then, in the new opened window, add your phone number and validate with the code that Twilio will provide (by call or SMS)



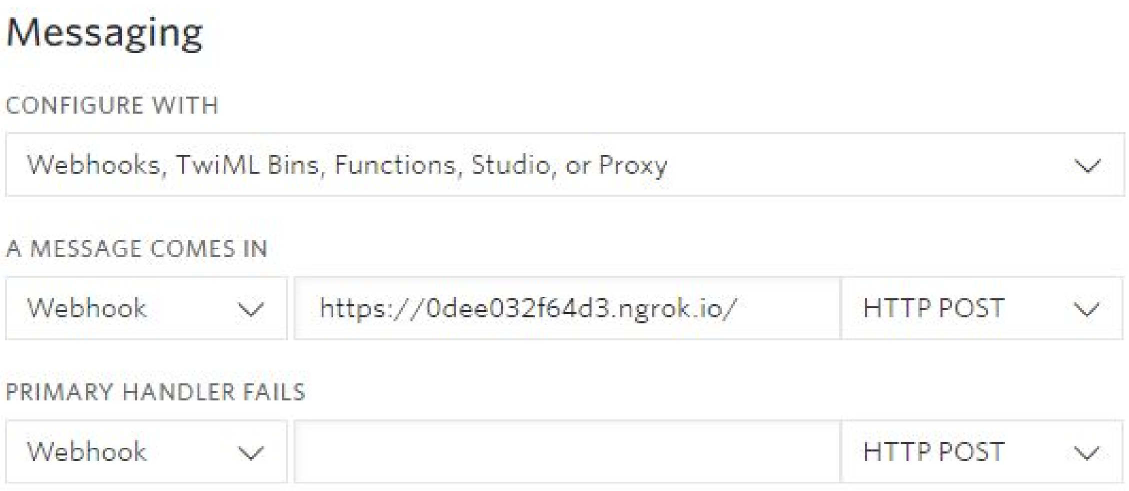
Enter your phone number and we'll **call** you to enter a verification code.

Or we can text you instead.

Webhook

On the next screen, you need to add the url of the Webhook where the Chatbot Twilio Connector is installed. Select the option "*HTTP Post*".

If you want, you can add a second Webhook, in case the main fails.


Messaging

CONFIGURE WITH

Webhooks, TwiML Bins, Functions, Studio, or Proxy

A MESSAGE COMES IN

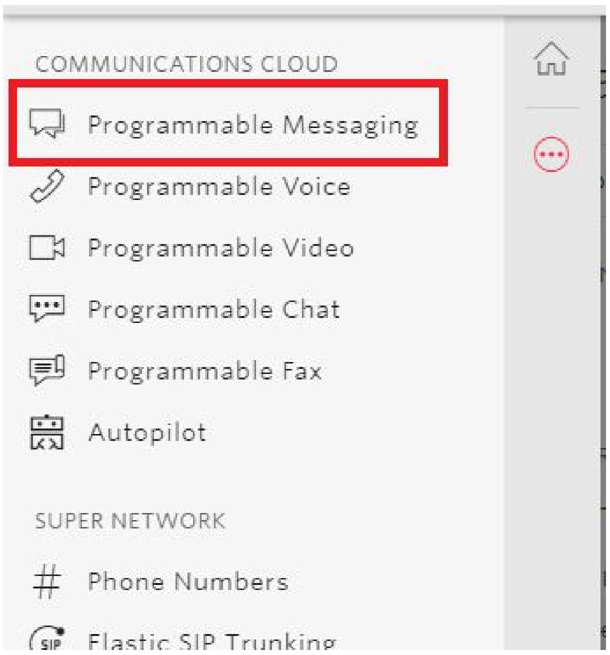
Webhook | https://0dee032f64d3.ngrok.io/ | HTTP POST

PRIMARY HANDLER FAILS

Webhook | | HTTP POST

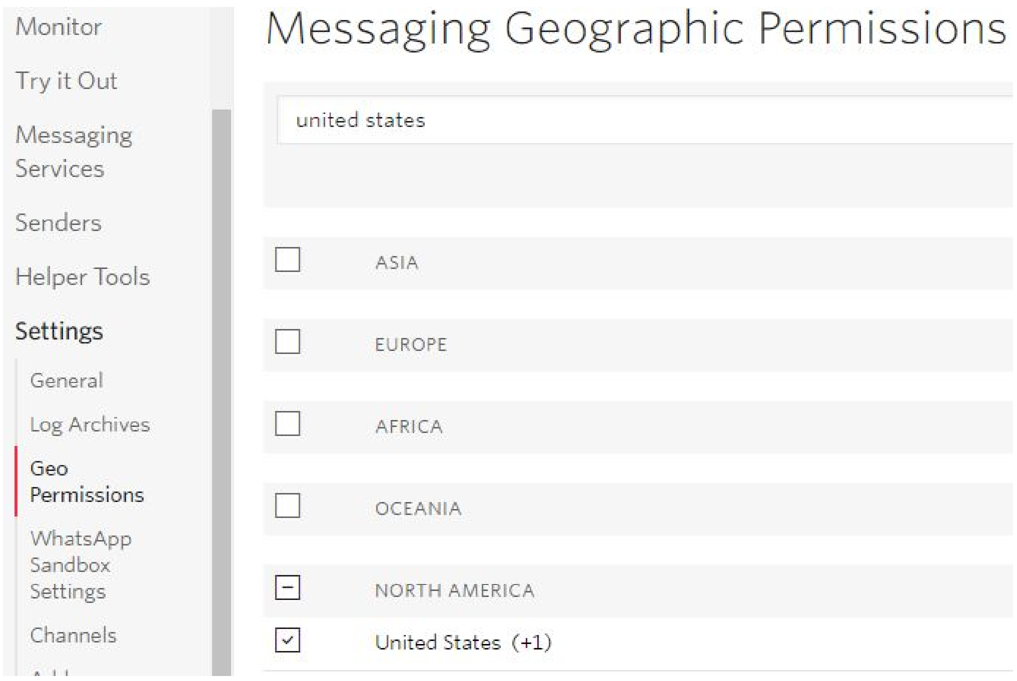
Set the environment for Whatsapp

For Whatsapp configuration, go to **All products & Services** icon (three dots icon on the left) and choose **Programmable Messaging**



Geo Permissions

And then go to **Settings** → **Geo Permissions**, and select the countries in which you want to use the service:



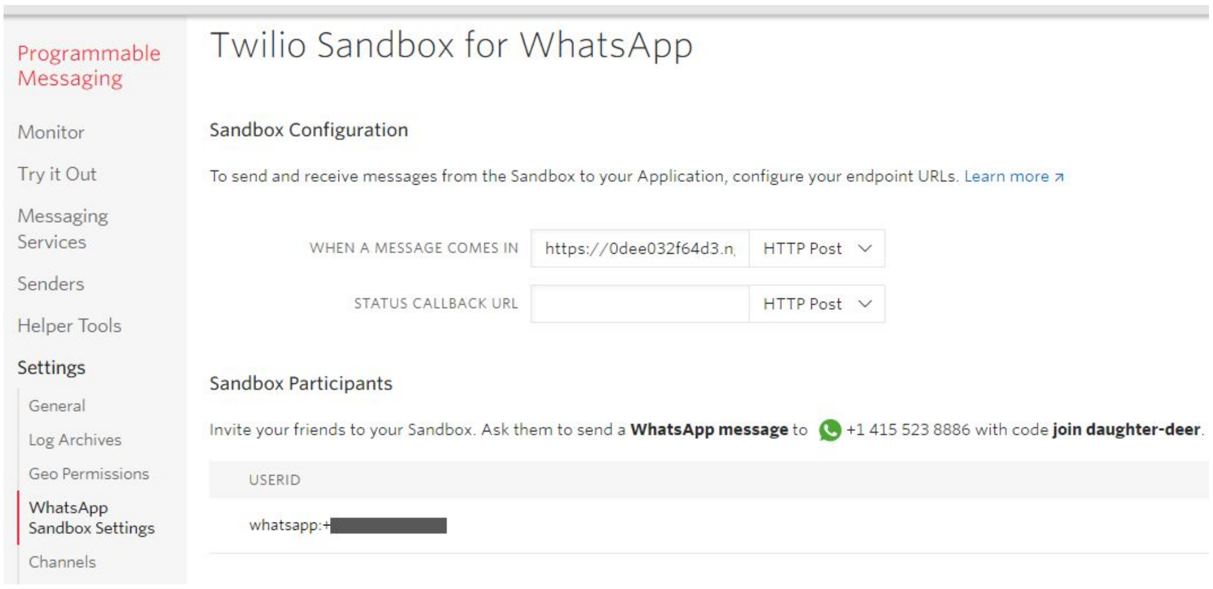
Messaging Geographic Permissions

united states

- [ ] ASIA
- [ ] EUROPE
- [ ] AFRICA
- [ ] OCEANIA
- [-] NORTH AMERICA
- [✓] United States (+1)

Webhook

After, go to “Settings → Whatsapp Sandbox Settings” and add the Webhook in which you have installed the Chatbot Twilio Template (the same SMS configuration):



Twilio Sandbox for WhatsApp

Sandbox Configuration

To send and receive messages from the Sandbox to your Application, configure your endpoint URLs. Learn more ↗

WHEN A MESSAGE COMES IN: https://0dee032f64d3.n, HTTP Post

STATUS CALLBACK URL: , HTTP Post

Sandbox Participants

Invite your friends to your Sandbox. Ask them to send a **WhatsApp message** to +1 415 523 8886 with code **join daughter-deer**.

USERID
whatsapp:+[redacted]

Additionally you can set a Webhook to call in case the main fails.

In order to start with tests, you need to join the participant list in Sandbox mode, sending a Whatsapp message to the given number with the starting phrase, located in the “Whatsapp Sandbox Settings” screen (ex: *join daughter-deer*).

Add the SID / Auth Token to Chatbot Twilio Template

In **conf/custom/twilio.php** file, you have to add your SID and Auth Token

```
return [

    'credentials' => [

        'sid' => '[SID]',

        'auth_token' => '[AUTH_TOKEN]'

    ]

];
```

Relevant URLs

- Twilio Register <https://www.twilio.com/try-twilio>
- Twilio Console <https://www.twilio.com/console>
- Twilio Documentation <https://www.twilio.com/docs>

Troubleshooting

Missing HyperChat messages

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

The bot is not answering (webhook configuration)

You can check the current webhook URL in your SMS or Whatsapp configuration. Also, check if the tokens in **conf/custom/api.php** (key and secret) are valid.

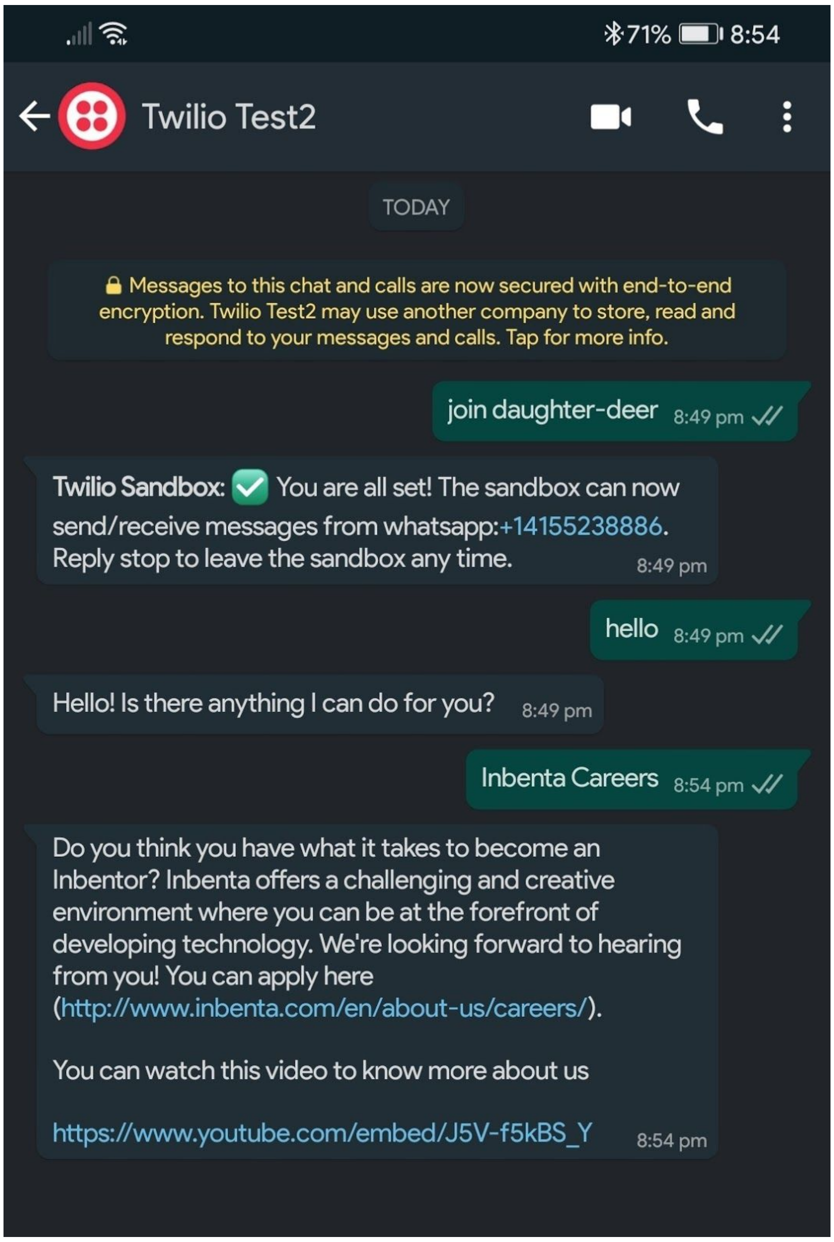
The bot is not answering (application cache)

If the previous tip doesn't work, maybe your application is caching older token values from extraInfo and you should 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. When you locate the directory, remove all the files named like "*cached-accesstoken-XXXX*" and "*cached-appdata-XXX*".

How to test it?

For SMS, just send a message to the trial number, and is all that you need to start a conversation

For Whatsapp, start the conversation with the given number (see Webhooks), sending the start phrase:



🔒 Messages to this chat and calls are now secured with end-to-end encryption. Twilio Test2 may use another company to store, read and respond to your messages and calls. Tap for more info.

join daughter-deer 8:49 pm

Twilio Sandbox: ✅ You are all set! The sandbox can now send/receive messages from whatsapp:+14155238886. Reply stop to leave the sandbox any time. 8:49 pm

hello 8:49 pm

Hello! Is there anything I can do for you? 8:49 pm

Inbenta Careers 8:54 pm

Do you think you have what it takes to become an Inbentor? Inbenta offers a challenging and creative environment where you can be at the forefront of developing technology. We're looking forward to hearing from you! You can apply here (http://www.inbenta.com/en/about-us/careers/).

You can watch this video to know more about us

https://www.youtube.com/embed/J5V-f5kBS_Y 8:54 pm