Nexmo Integration

Index

Introduction

The purpose of this documentation is to define the integration of Inbenta's Chatbot and Hyperchat solution with Whatsapp using an API provided by Nexmo.

Features

These are the supported answer types and features:

- Text answers
- Polar Questions
- Chained answers
- Forms
- URL
- Images
- Audios
- Static location
- Files

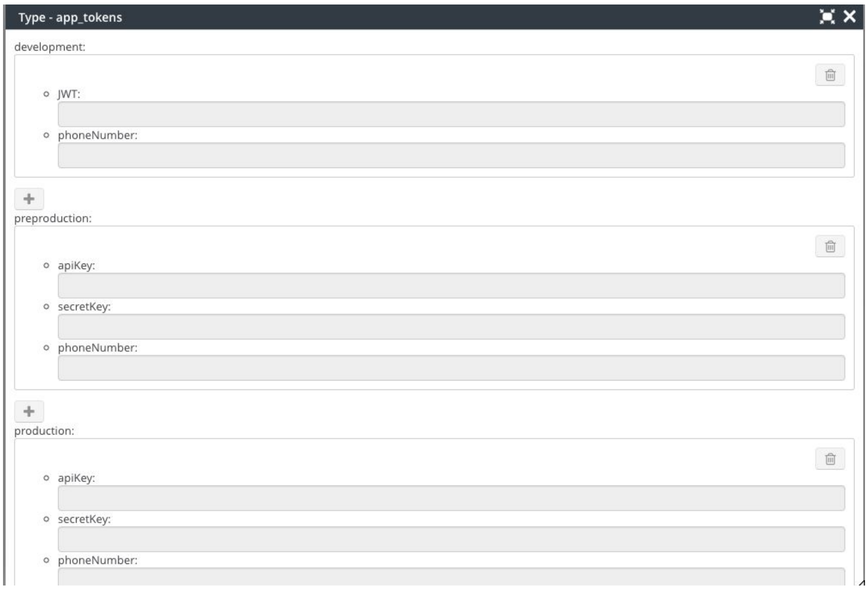
Other important features:

- Content ratings (yes/no, selected by text)
- HyperChat escalation after X no-results answers (triesBeforeEscalation)
- Escalate to Hyperchat when the FAQ has 'ESCALATE' setting set to 'TRUE'
- Retrieve Nexmo tokens from ExtraInfo
- Retrieve translations from ExtraInfo
- Retrieve agents timetable from ExtraInfo
- Retrieve only chat mode from ExtraInfo
- Using a keyword to allow the users close the chat

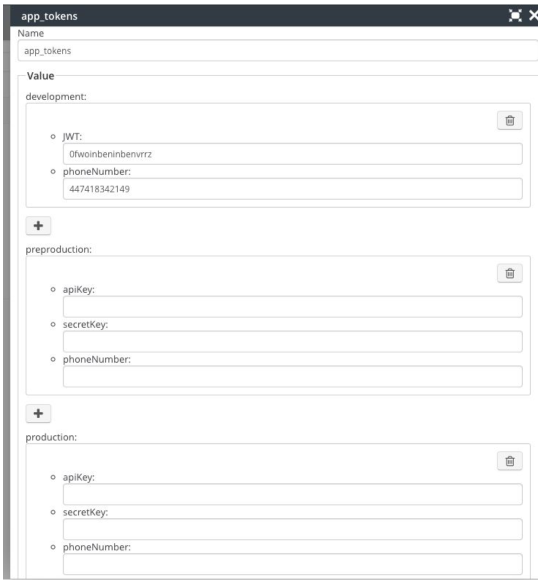
Prepare your Ibenta instances

Create tokens object in ExtraInfo

1. In your Backstage instance, go to Knowledge → Extra Info and click on '*Manage groups and types*'. Then click on *'Add a new group'* and name it **nexmo**.
2. Then click *'Add type'* in your group and name it **'app_tokens'** and click on *'OK*'.
3. Manage the app_tokens type clicking in the edit button. Create three properties: '**development**', '**preproduction**' and '**production**', all with *'multiple'* type. Add all the following 'text' type properties for each environment. If any of the environments will be used in Nexmo Sandbox mode, set the properties described in **a. sandbox**, if the environment won't use the Nexmo Sandbox mode, use the properties described in **b. Normal environments**. When you finish click on *'OK*':
    a. **sandbox:**
        i. **JWT:** JWT code provided by **Nexmo**
        ii. **phoneNumber:** The testing phone number provided by **Nexmo**
    b. Normal **development**, **preproduction** and **production** environments**:**
        i. **apiKey:** Your **Nexmo** Application Api Key
        ii. **secretKey:** Your **Nexmo** Application Secret Key
        iii. **phoneNumber:** Your **Nexmo** Application phone number



4. In Knowledge → Extra Info click in *'New Entry'*. Set **'app_tokens'** as the name. Select the group '**nexmo**' and select the type '**app_tokens**'. When you have your application tokens you can fill them in this object and click on *'OK'*:



Group	Name	Value
nexmo	app_tokens	• **development:** 1. ▪ **JWT:** 0fwoinbeninbenvrrz ▪ **phoneNumber:** 447418342149 • **preproduction:** 1. ▪ **apiKey:** *Empty values* ▪ **secretKey:** *Empty values* ▪ **phoneNumber:** *Empty values* • **production:** 1. ▪ **apiKey:** *Empty values* ▪ **secretKey:** *Empty values* ▪ **phoneNumber:** *Empty values*

5. Remember to publish your ExtraInfo changes by clicking the *'Post'* button.

```
Information
Remember to publish your ExtraInfo changes by clicking the 'Post' button.
```

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' → nexmo → 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'
- **close_chat_key_word** =>’/close'
- **out_of_time** => ‘_There are no agents connected_’
- **queue_warning** => 'Wait until an agent is connected before making a question.'

Close chat keyword

A close chat keyword must be set to allow users to close the chat. It can be customized in each language by modifying the extraInfo label ‘**close_chat_key_word**’.

Messages format

Whatsapp allows you to add format to your messages. You can take the following table as a reference:

Format	Symbol	Example
Bold	Asterisk(*)	Your total is *$10.50*.
Italic	Underscore(_)	Welcome to _WhatsApp_!
~~Strike-through~~	Tilde(~)	This is ~better~ best!
`Code`	Three backticks(```)	``` print 'Hello World'; ```
Carriage Return	\n	Hello \n World

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

Create Agents Timetable object in ExtraInfo (optional)

You can manage the agents timetable from Extra Info. Here are the steps to create the timetable object:

1. In your Backstage instance, go to Knowledge → Extra Info and click on 'Manage groups and types' → nexmo → Add type. Name it '**timetable**' and add a new property with type 'Multiple' named with the weekdays (monday, tuesday, wednesday...).
2. Inside every weekday object, add all the work periods. Each work period should be a 'text' type entry and they should follow **hh:mm-hh:mm** pattern as it is shown in the example.

3. Save your timetable object.

Now you can create the ExtraInfo object by clicking the **New entry** button, selecting the 'timetable' type and naming it as 'timetable'. Then, fill each period with your desired working hours and remember to publish ExtraInfo by clicking the **Post** button.

nexmo	timetable	• **monday:** 1.▪ **period_1:** 08:00-18:00 • **tuesday:** 1.▪ **period_1:** 08:00-18:00 • **wednesday:** 1.▪ **period_1:** 08:00-18:00 • **thursday:** 1.▪ **period_1:** 08:00-18:00 • **friday:** 1.▪ **period_1:** 08:00-18:00 • **saturday:** 1.▪ **period_1:** 08:00-18:00 • **sunday:**

```
Information
Remember to publish your ExtraInfo changes by clicking the 'Post' button.
```

Create Only Chat setting in ExtraInfo (optional)

You can manage the only chat mode from Extra Info. Here are the steps to create the setting object:

1. In your Backstage instance, go to Knowledge → Extra Info and click on 'Manage groups and types' → nexmo → Add type. Name it '**settings**' and add a new property with type 'text' called **'only_chat_mode'** and click on *'OK'*.
2. In Knowledge → Extra Info click in *'New Entry'*. Set '**settings**' as the name. Select the group '**nexmo**' and select the type '**settings**'. Fill the **'only_chat_mode'** field with **true** or **false** and click on *'OK'*.

```
Information
```

```
Remember to publish your ExtraInfo changes by clicking the 'Post' button.
```

HyperChat integration (optional)

If you use HyperChat you must subscribe your UI to the Hyperchat events. Go to your Case Management instance in Case Management → Settings → Chat → Webhooks. Here, in the 'Events' column type "queues:update,invitations:new,invitations:accept,forever:alone,chats:close, messages:new,users:activity,system:info". In the 'Target' column paste your UI's URL, then click on the '+' button on the right.

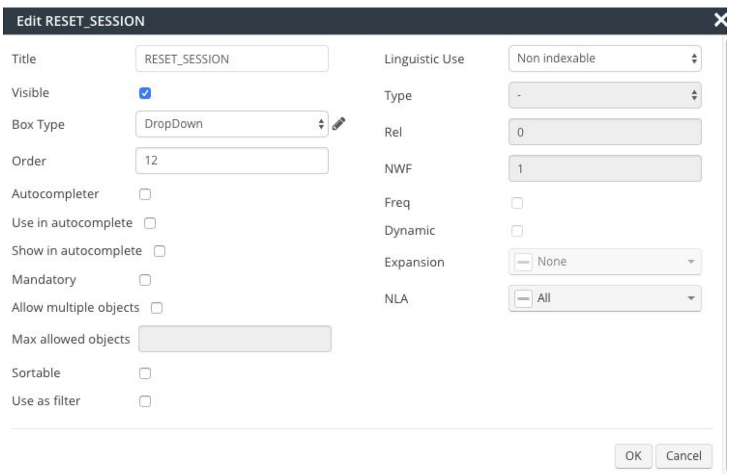
Dynamic settings

The template need the following dynamic settings to work properly:

RESET_SESSION

Because of whatsapp conversations can not be closed, this setting is needed to reset the session when the user agent conversation is finished in only chat mode. For example, if the conversation flow starts with a question (like some policy agreement) and the user disagrees, the shown content must have set the RESET_SESSION as **true**, and it will allow chatbot to ask throw the question again if the user wants to start a conversation with a human.

To enable this setting go to Settings → Dnamic → New entry and fill the title as **RESET_SESSION.**



Building the Nexmo template

Setup Chatbot Nexmo Template

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 found it in your instance → Case Management → Settings → Chat.
    - **secret**: Your HyperChat instance application secret. You can found it in your instance → Case Management → 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 → Case Management → 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 → Case Management → Settings → Sources.
    - **server**: The Hyperchat server URL assigned to your instance. The Hyperchat server URL is ***https://hyperchat-REGION.inbenta.chat***, where “REGION” is your Hyperchat server region. You can find it in your instance → Case Management → Settings → Chat → Region.
    - **server_port**: The port where to communicate with the Hyperchat server. It’s defined in your instance → Case Management → Settings → Chat -->Port
    - **queue**:
        - **active**: Enable or disable the queue system (“**true**” or “**false**”). It **MUST** be enabled in your instance too (Case Management → Settings → Chat → Queue mode).
    - **timetable**: Agents timetable by periods of work by weekday using **dd:hh-dd-hh** pattern.
    - **surveyId**: Survey ID that will be used when the conversation finishes.
- **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.

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 Nexmo the value is an empty array ("**array()**").
        - **answerAttributes:** Dynamic settings to show as bot answer. The default is ["ANSWER_TEXT"]. Setting multiple dynamic settings generates a bot answer with concatenated values with a newline character (\n). Limited to **2000 characters**.
        - **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.
- **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/*").

SANDBOX (sandbox.php)

This file allows configuring the sandbox environment custom settings.

- **force_sandbox_mode:** if the bot should use the Nexmo sandbox mode
- **jwt:** JWT key provided by **Nexmo**
- **endpoint**: **Nexmo** API endpoint where the calls will be made.

Deployment

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

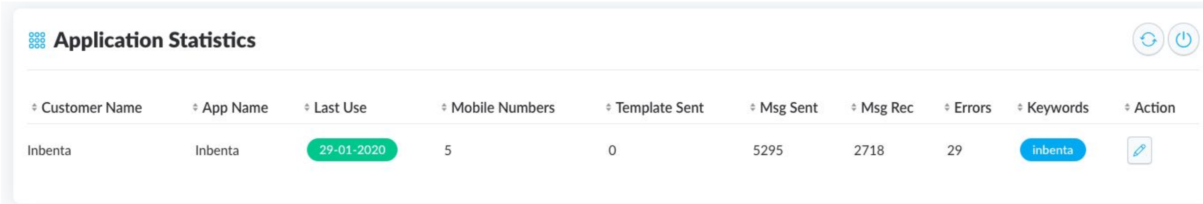
- Apache 2.4
- PHP 7.3
- PHP Curl extension
- Non-CPU-bound
- The latest version of **Composer** (Dependency Manager for PHP) to install all dependencies that Inbenta requires for the integration.
- If the client has a **distributed infrastructure**, this means that multiple servers can manage the user session, they must adapt their SessionHandler so that the entire session is shared among all its servers.

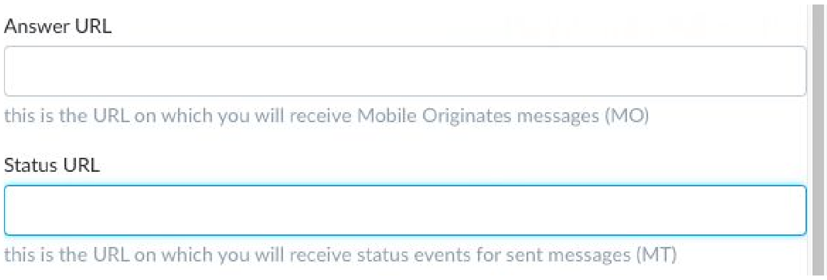
Prepare the Nexmo environment

Nexmo Sandbox Account

Nexmo Sandbox Account allows Nexmo customers to test their applications. In this account, Nexmo will provide a **JWT** code, a **keyword** and a **phone number**.

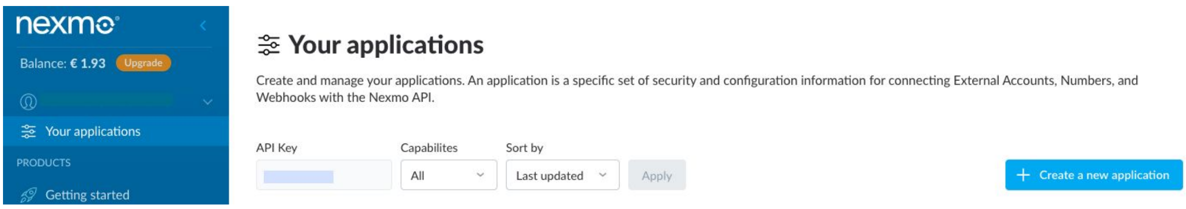
Go to https://sandbox.nexmodemo.com/ui/ to set webhooks that will manage the messages.



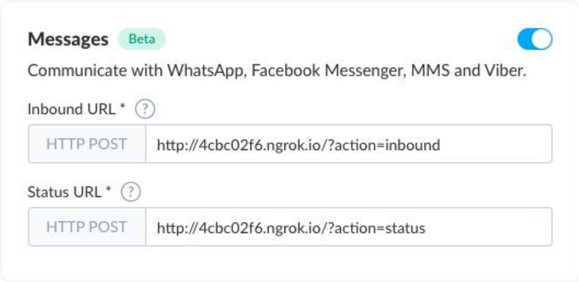


Create a Nexmo APP

To be able to create an application you need a Nexmo Account. Go to https://dashboard.nexmo.com/sign-up and fill the required information. Once your account is working, sign in here https://dashboard.nexmo.com/sign-in and create an application:



Fill the requested fields and generate the **API keys** and the **Inbound** and the **Status** url.




```
Information
The status url is needed but never used by the template. It sends an event
every time that that a nexmo message sent changes its status.
```

Relevant URL's

- Nexmo API Reference Portal <https://developer.nexmo.com/api/messages-olympus>
- Nexmo Sandbox Portal <https://sandbox.nexmodemo.com/ui/>
- Nexmo Messages Documentation <https://developer.nexmo.com/messages/overview>
- Nexmo Dashboard <https://dashboard.nexmo.com/sign-in>

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)

Check if your UI URL is configured in your Nexmo App. You can check the current webhook URL in your Nexmo App → Messages → Inbound URL. Also, check if the tokens in your instance ExtraInfo are valid.

The bot is not answering (application cache)

If the two previous tips don'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*".

The bot is answering in a only chat mode (dynamic settings configuration)

Check if RESET_SESSION dynamic has been set in settings. If it has been done and it still happening, check if the content before the bot answer has it set as **true**.

How to test it?

Test your UI from your Whatsapp write a message to your Whatsapp phone. Then, your bot should be answering your messages.