inbenta

Smooch

Integration

Index

Features

These are the supported answer types and features:

- Text answers
- Multiple Options
- Polar Questions
- Chained answers
- Forms
- Display a button to open a URL
- Custom FAQ title in buttons when displaying multiple options

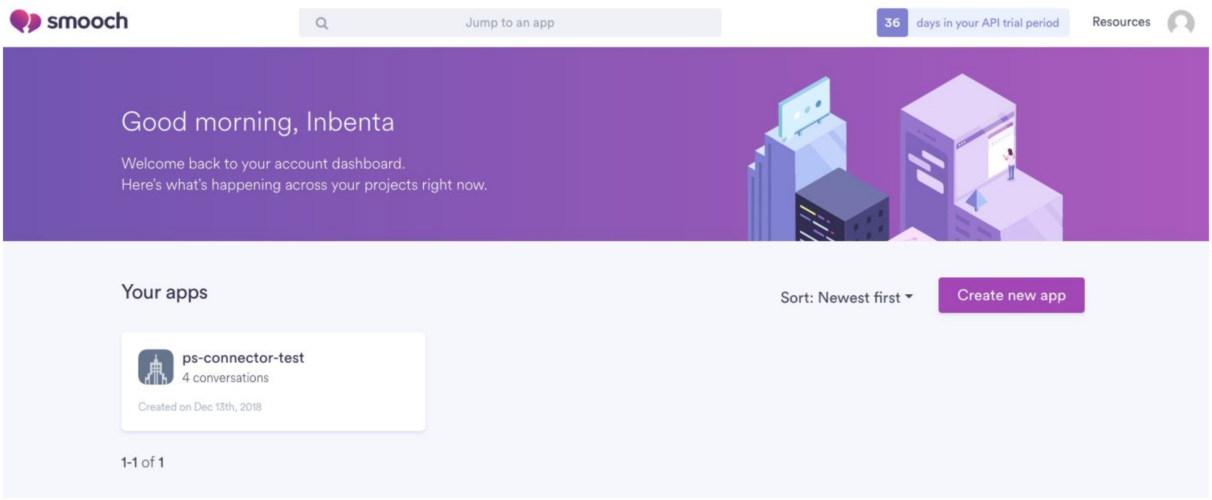
Other important features:

- 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'
- Retrieve Smooch API Key and Secret from ExtraInfo
- Retrieve translations from ExtraInfo

Prepare Smooch environment

Create a Smooch Application

1. Login to Smooch account (or create one).
2. In your Smooch Home, click on **[Create new Smooch App]**, and choose a name for your new Smooch app.



Good morning, Inbenta

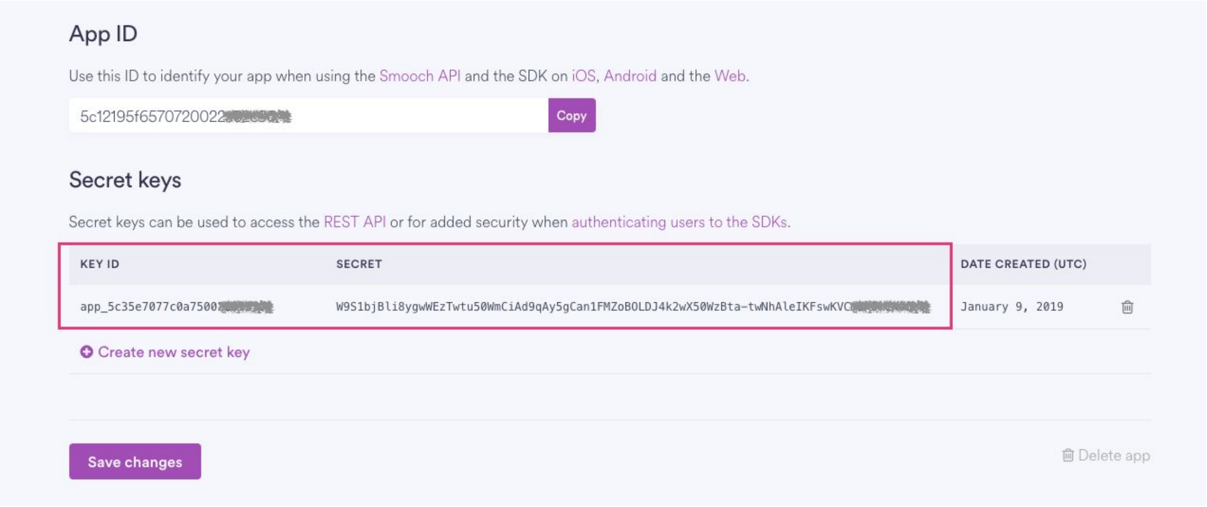
Welcome back to your account dashboard.
Here's what's happening across your projects right now.

Your apps

ps-connector-test
4 conversations
Created on Dec 13th, 2018

1-1 of 1

3. You will see your created app in the apps list.
4. Click on your application (in this example, "ps-connector-test") and go to the "Settings" tab.
5. Scroll to the "Secret keys" section and click on [Create a new secret key]
6. You will need these App **Key** and **Secret** to connect this Smooch app to an Inbenta's chatbot *(See the section: Create ExtraInfo object with Smooch's API Key and Secret)*.



App ID

Use this ID to identify your app when using the Smooch API and the SDK on iOS, Android and the Web.

5c12195f6570720022[redacted]

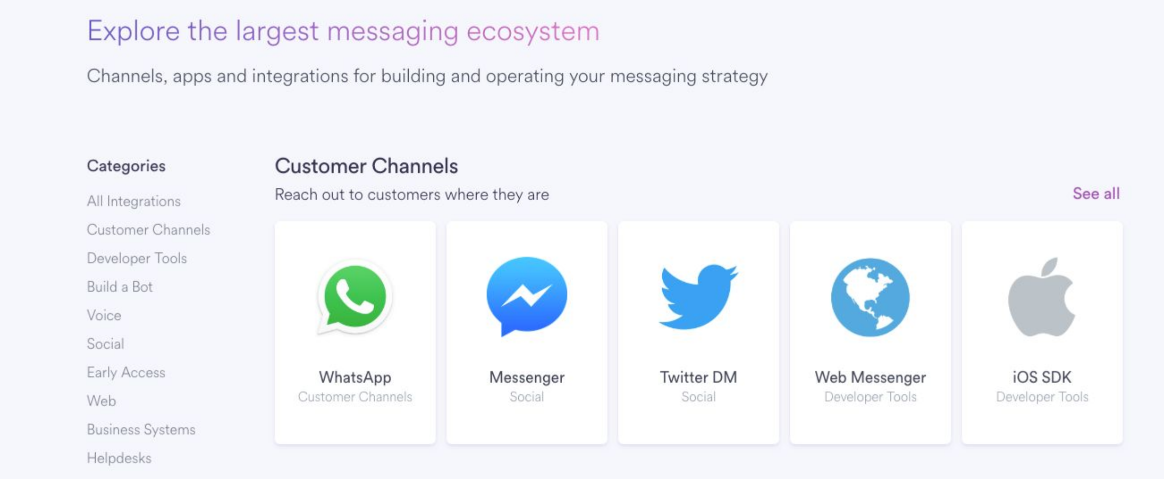
Secret keys

Secret keys can be used to access the REST API or for added security when authenticating users to the SDKs.

KEY ID	SECRET	DATE CREATED (UTC)
app_5c35e7077c0a7500[redacted]	W9S1bjBli8ygwWEzTwtu50WmCiAd9qAy5gCan1FMZoBOLDJ4k2wX50WzBta-twNhAleIKFswKVC[redacted]	January 9, 2019

Connect Channels to the Smooch Application

1. Click on your application (in this example, "ps-connector-test").
2. Click on **[Connect channels]** while in the "Overview" tab (or go to the integrations tab).
3. Select the channel you want to connect to your smooch application from the "Customer Channels" list:

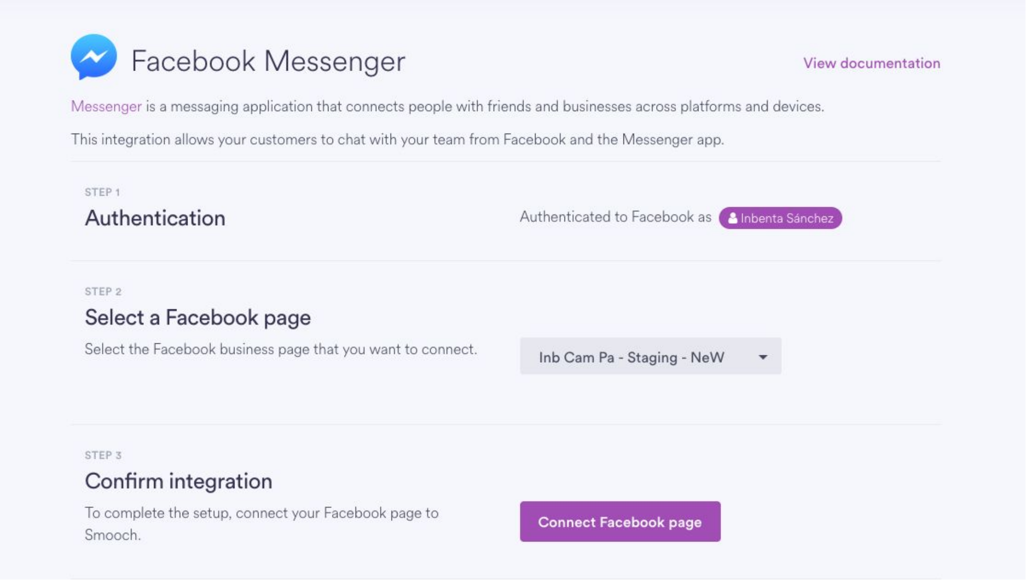


Explore the largest messaging ecosystem

Channels, apps and integrations for building and operating your messaging strategy

4. Click on the Channel you want to connect to your Smooch application and follow Smooch's instructions to connect it. Example for **Messenger/Facebook connection**:

   a. Click on [Messenger] in the Smooch Customer Channel list and then click on [Connect].
   b. Step 1: Authenticate into your Facebook account with management access to the Facebook Page you want to connect or verify that you are logged in with the correct profile.
   c. Step 2: Select from the Dropdown the Facebook page you want to connect
   d. Step 3: Click on [Connect Facebook Page].



Facebook Messenger

Messenger is a messaging application that connects people with friends and businesses across platforms and devices.

This integration allows your customers to chat with your team from Facebook and the Messenger app.

STEP 1
Authentication
Authenticated to Facebook as Inbenta Sánchez

STEP 2
Select a Facebook page
Select the Facebook business page that you want to connect.
Inb Cam Pa - Staging - NeW

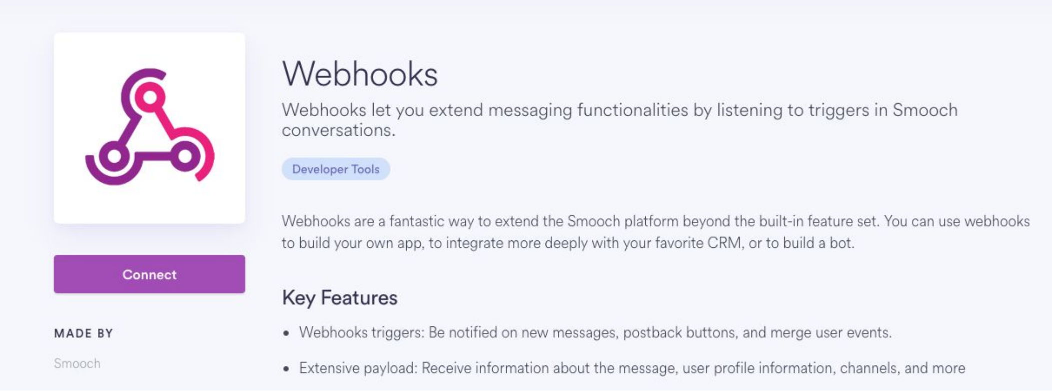
STEP 3
Confirm integration
To complete the setup, connect your Facebook page to Smooch.

That is it, your Messenger/Facebook Page is connected to smooch and all the messages written by users will be sent to smooch.

Note that some Customer channels, like Messenger, have a very intuitive and straight forward connection system, others need a bit more work and follow closely Smooch's steps and documentation.

Connect Smooch to your Smooch-Chatbot Connector APP

1. Inside your Smooch application (in this example, "ps-connector-test"), click on **[Connect channels]** while in the "Overview" tab (or go to the integrations tab).
2. Scroll down to "Developer Tools" and click on **[Webhooks]**, and then [Connect]:



Webhooks

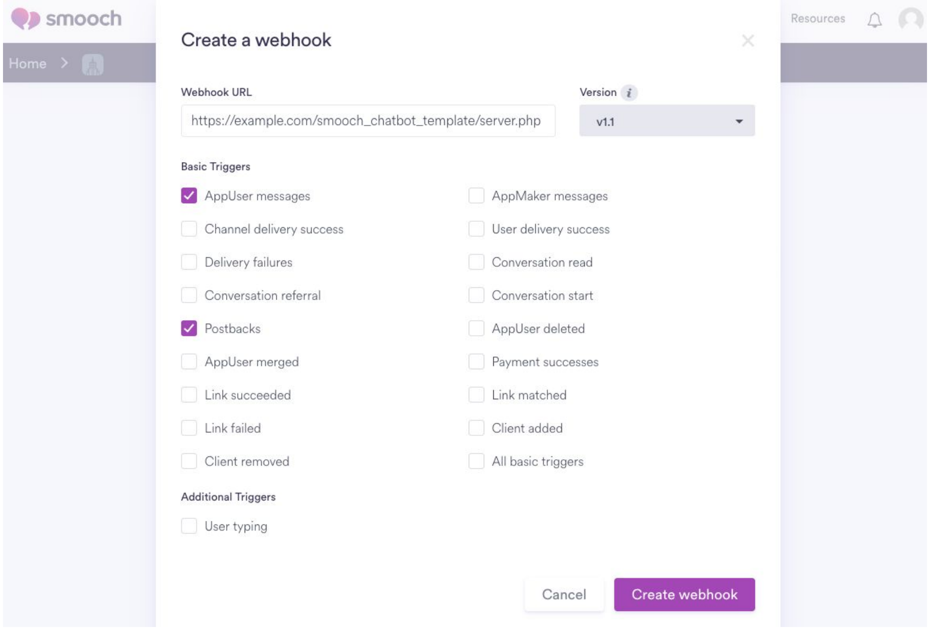
Webhooks let you extend messaging functionalities by listening to triggers in Smooch conversations.

Webhooks are a fantastic way to extend the Smooch platform beyond the built-in feature set. You can use webhooks to build your own app, to integrate more deeply with your favorite CRM, or to build a bot.

Key Features

- Webhooks triggers: Be notified on new messages, postback buttons, and merge user events.
- Extensive payload: Receive information about the message, user profile information, channels, and more

3. Click on **[Create a new webhook]**.
4. Insert the URL pointing to the Smooch-Chatbot Connector APP, example: *https://example.com/smooch_chatbot_template/server.php (See the section: [Building a Smooch-Chatbot Connector APP](#)).*
5. Subscribe to the **appUser:Messages** (roughly, user text messages) and **Postback** (roughly, clicks on buttons) triggers:



Create a webhook

Webhook URL: https://example.com/smooch_chatbot_template/server.php

Version: v1.1

Basic Triggers

☑ AppUser messages
□ Channel delivery success
□ Delivery failures
□ Conversation referral
☑ Postbacks
□ AppUser merged
□ Link succeeded
□ Link failed
□ Client removed
□ AppMaker messages
□ User delivery success
□ Conversation read
□ Conversation start
□ AppUser deleted
□ Payment successes
□ Link matched
□ Client added
□ All basic triggers

Additional Triggers

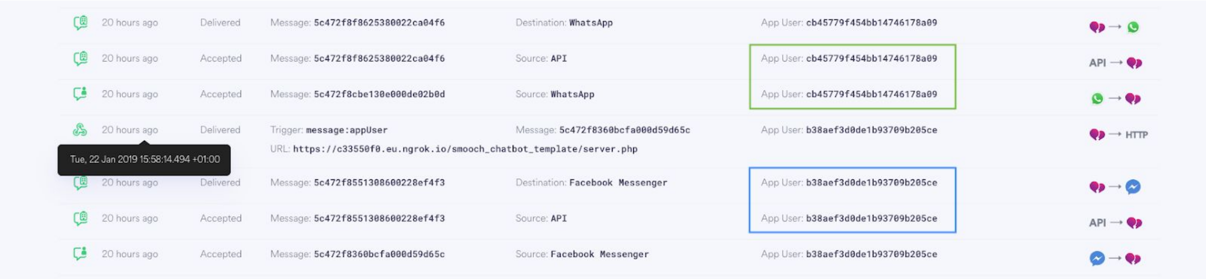
□ User typing

6. Create the webhook.

From the moment the webhook is created, all the messages (appUser:Messages and Postbacks) received in the channels to which Smooch is connected (Messenger, Whatsapp, Skype, etc.), will be sent to this webhook.

FYI on Smooch's App Users, conversations and logs

Smooch logs every conversation message and interaction sent from and to all the channels connected to your Smooch app. To see these logs go to your Smooch application (in this example "ps-connector-test") and to the logs tab.



```
20 hours ago   Delivered   Message: 5c472f8f8625380022ca04f6   Destination: WhatsApp             App User: cb45779f454bb14746178a09   →
20 hours ago   Accepted    Message: 5c472f8f8625380022ca04f6   Source: API                       App User: cb45779f454bb14746178a09   API →
20 hours ago   Accepted    Message: 5c472f8cbe130e000de02b0d   Source: WhatsApp                  App User: cb45779f454bb14746178a09   →
20 hours ago   Delivered   Trigger: message:appUser            Message: 5c472f8360bcfa000d59d65c App User: b38aef3d0de1b93709b205ce   → HTTP
                           URL: https://c33550f0.eu.ngrok.io/smooch_chatbot_template/server.php
Tue, 22 Jan 2019 15:58:14.494 +01:00
20 hours ago   Delivered   Message: 5c472f8551308600228ef4f3   Destination: Facebook Messenger   App User: b38aef3d0de1b93709b205ce   →
20 hours ago   Accepted    Message: 5c472f8551308600228ef4f3   Source: API                       App User: b38aef3d0de1b93709b205ce   API →
20 hours ago   Accepted    Message: 5c472f8360bcfa000d59d65c   Source: Facebook Messenger        App User: b38aef3d0de1b93709b205ce   →
```

The actual contents of the conversations can be retrieved via API with the Message-ID.

Note that Smooch creates a **unique App User ID for every user in every channel**.

Building a Smooch-Chatbot Connector APP

Download Smooch Template

You can find the PS Chatbot Smooch template in Git. Download it from https://github.com/inbenta-integrations/smooch_chatbot_template and move it into your infrastructure.

Configuration (required)

In your *smooch_chatbot_template* directory, go to **conf**. Here, you have a README file with some structure and usage explanations.

- **CHATBOT-SMOOCH basic connection** (api.php)
    - The basic smooch-chatbot connection can be configured by adding your chatbot's instance **key** and **secret** values in the corresponding parameters of the *conf/custom/api.php* file.
- **ENVIRONMENTS** (environments.php)
    - The connector App can be set to work in either 'development', 'preproduction' or 'production' environment.
    - By default, the connector App will assume 'production' environment, unless the rules that set 'development' or 'preproduction' environment are fulfilled (info on how to set those rules can be found in the *environments.php* file itself).
    - The set environment affects:
        1. The Smooch's App that will be used, by setting the Key/Secret pair that will be retrieved from extraInfo *(See the section: Create ExtraInfo object with Smooch's API Key and Secret)*
        2. The logging environment in the chatbot instance for the events in the conversation.

Configuration (advanced)

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.

There are several advanced features that can be enabled using the optional configuration files *chat.php* and *conversation.php*.

To activate the use of these optional configurations, copy the desired file(s) from **/conf/default/** to **/conf/custom/** (please, always leave a default version unchanged in /conf/default/). The bot will detect the new file(s) inside the /conf/custom/ directory and load the customized version(s).

Below a full list of the optional configuration files and its fields described:

- **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 Smooch 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).
            - **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 consecutively 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 content-rating feature.
        - **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*).
            - **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).

    - **digester**
        - **button_title**: Dynamic setting which value will be displayed as the content title in multiple-options buttons. It's useful to have an alternative title with less than 20 characters because Facebook limits buttons text to 20 chars.
        - **url_buttons:** Displays a button which opens one desired URL (*specific for Smooch*).
            - **attribute_name**: It should be a grouped dynamic setting which child values contain the button URL and title.
            - **button_title_var**: Child value in the grouped dynamic setting which value contains the url-button title.
            - **button_url_var**: Child value in the grouped dynamic setting which value contains the url-button URL.

- **HYPERCHAT** (chat.php)
    - **chat**
        - **enabled**: Enables or disables HyperChat.
        - **version**: HyperChat version. The default and latest one is 1.
        - **region**: The geographical region where the HyperChat app "lives". Valid values are: "eu" for Europe and "us" for the United States. You can find your region in your instance → Case Management → Settings → Chat.
        - **appId**: The ID of the HyperChat app. This defines the Case Management instance in which the Hyperchat will open. 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 points. 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.
    - **triesBeforeEscalation**: Number of no-result answers in a row after which the bot should escalate to an agent (if available). Numeric value, not a string. *Zero means disabled*.
    - **negativeRatingsBeforeEscalation**: Number of negative content ratings in a row after which the bot should escalate to an agent (if available). Numeric value, not a string. *Zero means disabled.*

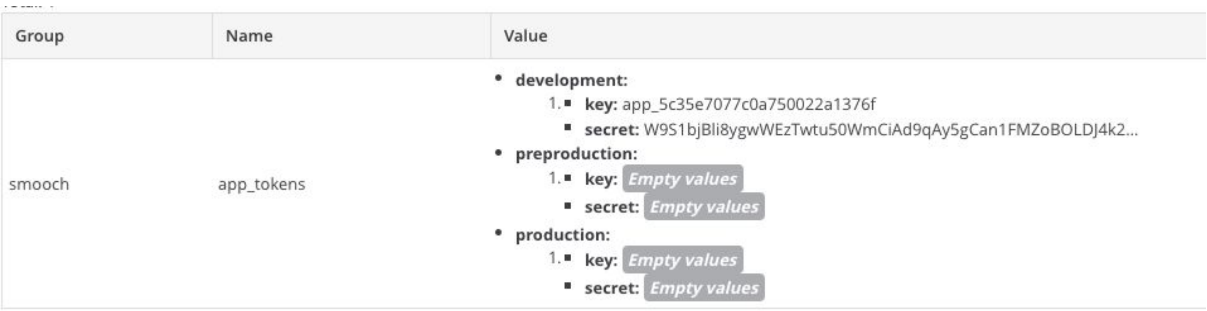
Inbenta's backstage instances setup

Chatbot Instance

Create ExtraInfo object with Smooch's API Key and Secret

1. In your **chatbot backstage** instance, go to "Knowledge" → "**Extra Info**".
2. Click on "Manage groups and types" and then on [Add a new group]. Name the new group **'smooch'**.
3. While still in the "Manage groups and types" modal, click on [Add type] inside the "smooch" group section:
    a. Name the new type '**app_tokens**'.
    b. Make sure the group is 'smooch'.
    c. Create 3 properties with names **'development'**, **'preproduction'** and **'production'** and type multiple.
    d. Inside each of those 3 properties, create 2 sub-properties with names **'key'** and **'secret'** and type **text**.
    e. Click on [OK] to save your **'app_token'** object.
4. While in "Knowledge" → "Extra Info", click on [New Entry]:
    a. Name the entry '**app_tokens**'.
    b. Select the group '**smooch**' and the type '**app_tokens**'.
    c. Insert the Smoch's app **Key/Secret** that you can find in your Smooch's App "Settings" tab *(See the section: Create a Smooch Application)*.
5. Note that you can have different Smooch App's set in for the different environments: development, preproduction or production.
6. Publish your ExtraInfo changes by clicking the [Post] button.

It should look something like this:



Group	Name	Value
smooch	app_tokens	• **development:** 1. ▪ **key:** app_5c35e7077c0a750022a1376f ▪ **secret:** W9S1bjBli8ygwWEzTwtu50WmCiAd9qAy5gCan1FMZoBOLDJ4k2... • **preproduction:** 1. ▪ **key:** *Empty values* ▪ **secret:** *Empty values* • **production:** 1. ▪ **key:** *Empty values* ▪ **secret:** *Empty values*

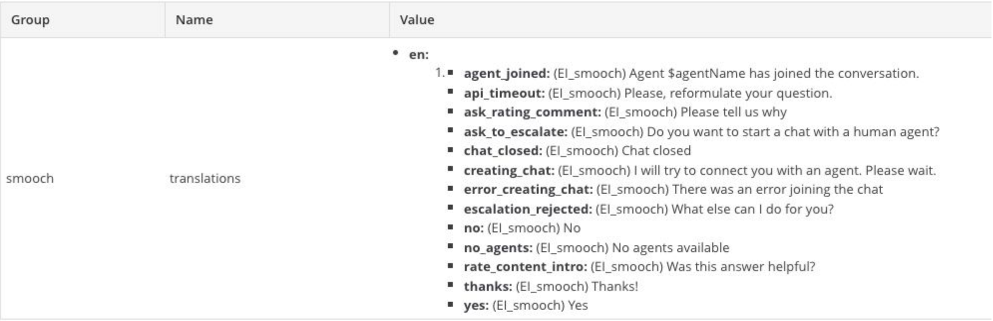
Remember to publish your ExtraInfo changes by clicking the 'Post' button.

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 **chatbot backstage** instance, go to "Knowledge" → "**Extra Info**".
2. Click on 'Manage groups and types' and the [Add type] inside the 'smooch' group (if there is no 'smooch' group, create it).
    a. Name the new type '**translations**'.
    b. Make sure the group is 'smooch'.
    c. Create a new property of type multiple and name it after your chatbot language label (en, es, it...). Example, name 'en', for English.
    d. Inside the language object, add all the labels that you want to override as sub-properties of type *text*. Example and full list:
        - **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'
    e. Click on [OK] to save your 'app_token' object.
3. While in "Knowledge" → "Extra Info", click on [New Entry]:
    a. Name the entry '**translations'**.
    b. Select the group '**smooch**' and the type '**translations**'.
    c. Insert the desired custom translations.
4. Publish your ExtraInfo changes by clicking the [Post] button.

It should look something like this:



Group	Name	Value
smooch	translations	• **en:** 1. ▪ **agent_joined:** (EI_smooch) Agent $agentName has joined the conversation. ▪ **api_timeout:** (EI_smooch) Please, reformulate your question. ▪ **ask_rating_comment:** (EI_smooch) Please tell us why ▪ **ask_to_escalate:** (EI_smooch) Do you want to start a chat with a human agent? ▪ **chat_closed:** (EI_smooch) Chat closed ▪ **creating_chat:** (EI_smooch) I will try to connect you with an agent. Please wait. ▪ **error_creating_chat:** (EI_smooch) There was an error joining the chat ▪ **escalation_rejected:** (EI_smooch) What else can I do for you? ▪ **no:** (EI_smooch) No ▪ **no_agents:** (EI_smooch) No agents available ▪ **rate_content_intro:** (EI_smooch) Was this answer helpful? ▪ **thanks:** (EI_smooch) Thanks! ▪ **yes:** (EI_smooch) Yes

Hyperchat: Case Management Instance

Create webhook to the Smooch-Chatbot connector APP and subscribe to Hyperchat events

To activate the escalation to agent possibility in your Smooch-Chatbot integration you need to configure your Case Management instance as follows:

1. Go to Case Management → Settings → Chat and scroll down to '**Webhooks**' and create a new webhook.
2. In 'Target', insert the URL pointing to the Smooch-Chatbot Connector APP, example: *https://example.com/smooch_chatbot_template/server.php (See the section: Building a Smooch-Chatbot Connector APP).*
3. In 'Events' insert the following comma-separated events list:
    - **Invitations:new,invitations:accept,forever:alone,chats:close,messages:new,users:activity**.

Relevant URL's

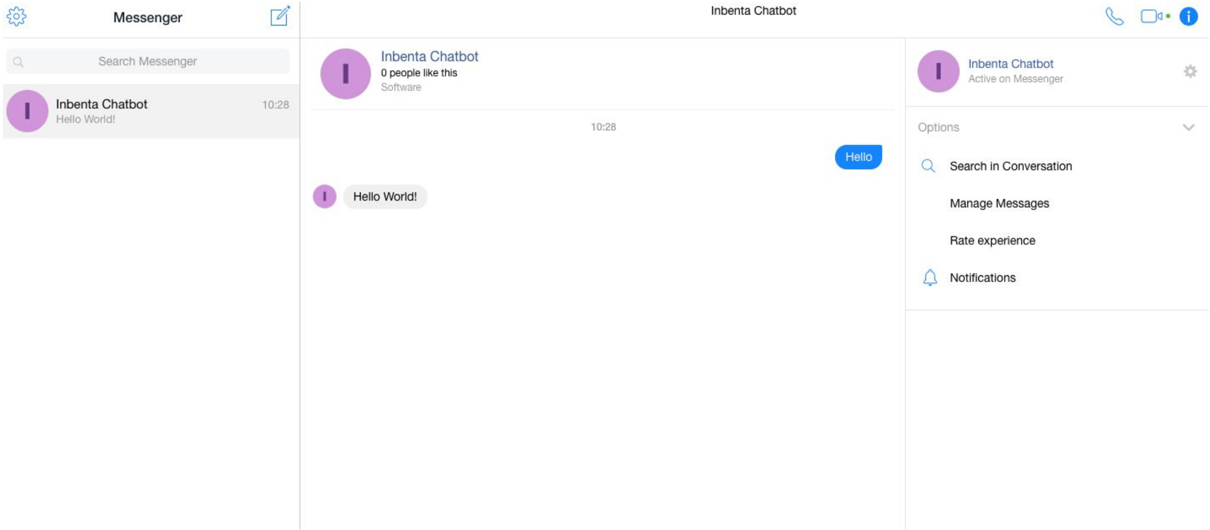
- Smooch app login <https://app.smooch.io/>
- Smooch Guide <https://docs.smooch.io/guide/intro-to-channels/>
- Smooch Guide → Channels <https://docs.smooch.io/guide/intro-to-channels/>
- Smooch API documentation <https://docs.smooch.io/rest/#introduction>

Troubleshooting

- **Bot is not answering (Smooch webhook configuration)**: Check if your Smooch-Chatbot Connector App URL is configured in your 'Smooch's App → Webhooks' section. Check that the webhook is subscribed to 'userApp:messages' and 'Postbacks' triggers.
- **Bot is not answering (Chatbot backstage configuration):** Make sure your Smooch's App Key and Secret are correctly configured in your instance ExtraInfo and are valid.
- **HyperChat integration si not working (Case Management backstage configuration)**: If you use HyperChat you want to subscribe your Smooch-Chatbot Connector App to the Hyperchat events. Go to your Case Management instance in Case Management → Settings → Chat → Webhooks. Here, in the 'Events' column type "*invitations:new,invitations:accept,forever:alone,chats:close,messages:new,users:activity*" (note the comma separation but NO blank space). In the 'Target' column paste your UI's URL, then click on the '+' button on the right.
- **Missing HyperChat messages (Smooch-Chatbot Connector App configuration)**: Check if your Smooch-Chatbot Connector App is subscribed to Hyperchat webhooks as described in the previous tip. Also, check if the Hyperchat settings in the Smooch-Chatbot Connector App are valid in **conf/custom/chat.php**.

How to test it?

Test your Smooch-Chatbot integration via any Channel you have configured in your Smooch App. For example, if you have configured Messenger as a Channel, you can test your UI from the Facebook page connected or you can go to https://www.messenger.com and search your page name in the search bar (top left). Then, Inbenta's bot should be answering your messages through Smooch.



10:28

Hello

Hello World!