inbenta

Line Integration

Index

Introduction **3**

Features **3**

Prepare Line environment **4**
- Create a Line Application 4
- Line Application with Switch 6

Building a Line-Chatbot Connector APP **8**
- Setup Chatbot Line Template 8
  - Download Line Template 8
  - Configuration (required) 8
  - Configuration (Switcher and advanced) 8

Inbenta's backstage instances setup **11**
- Chatbot Instance 11
  - Create ExtraInfo object with Line's Channel ID and Secret 11
  - Create translations object in ExtraInfo (optional) 13

Important note on the Switcher API **14**
- Escalation tracking events 14

Relevant URL's **15**

Troubleshooting **15**

How to test it? **16**

Document History

Version number	Modification date
1.0	08/06/2019

Introduction

The purpose of this document is to help Inbenta customers create a chatbot using the Inbenta APIs and the switching API provided by LINE in the simplest way possible.

The document describes all the features that are integrated by default and *how/what* changes must be made to both the Line developer portal and the Inbenta Backstage to get a complete and well-configured Chabot.

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

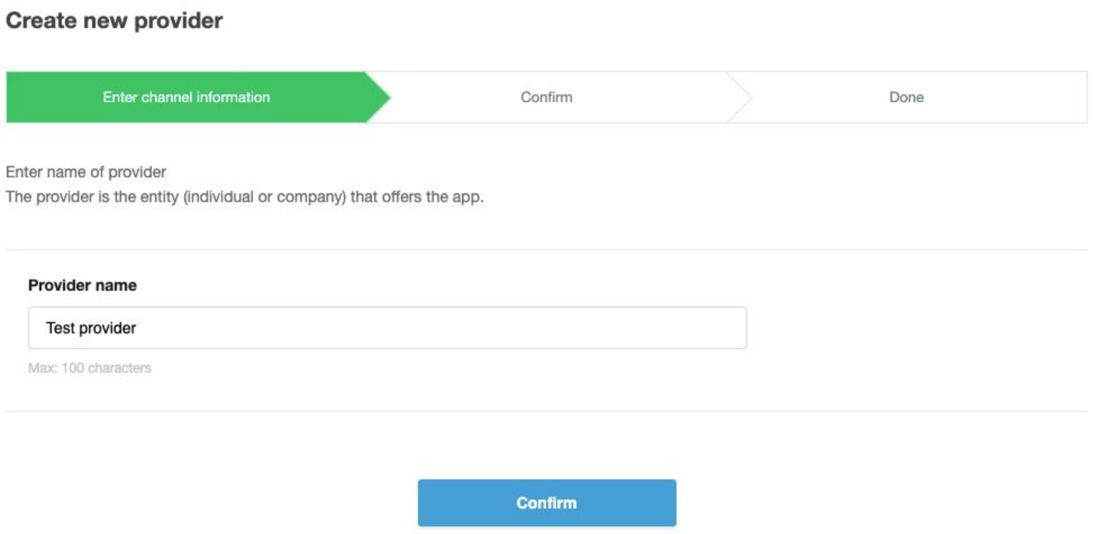
Other important features:

- Content ratings (yes/no + comment)
- Line Switching API trigger after X no-results answers (triesBeforeEscalation)
- Line Switching API trigger after X negative ratings (negativeRatingsBeforeEscalation)
- Line Switching API trigger when the FAQ has 'ESCALATE' setting set to 'TRUE'
- Retrieve Line's Channel ID and Secret and other optional tokens from ExtraInfo
- Retrieve translations from ExtraInfo

Prepare Line environment

Create a Line Application

1. Login to your Line developer account. You will need a Line user account (for example, Line for Android or IOs) or will have to create one.
2. In your Line Home, click on **[Create new Provider]**, and choose a name for your provider.



Create new provider

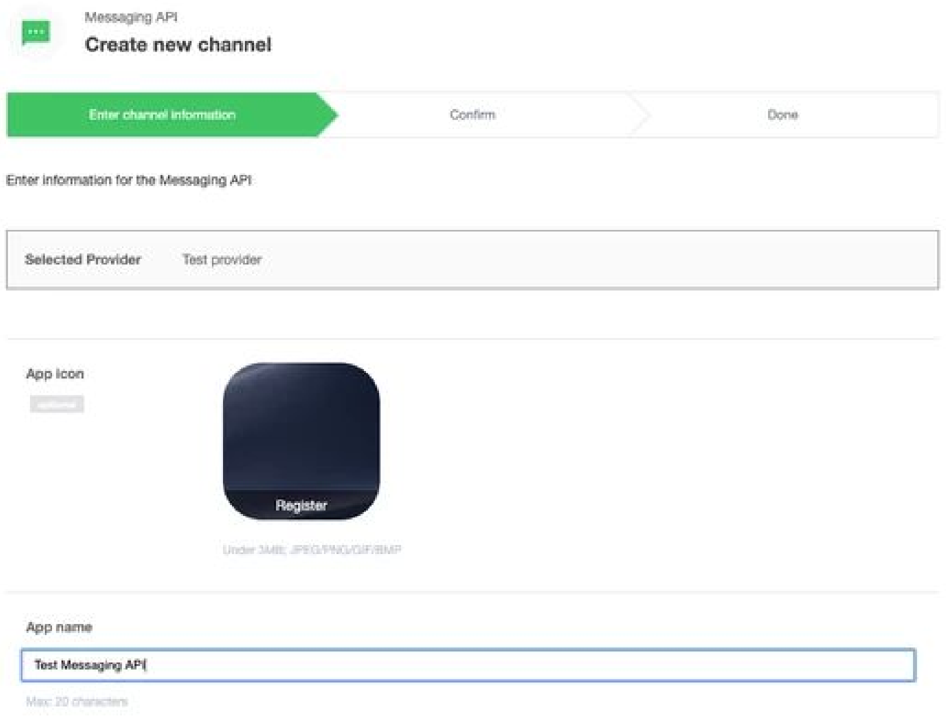
Enter channel information | Confirm | Done

Enter name of provider
The provider is the entity (individual or company) that offers the app.

Provider name
Test provider
Max: 100 characters

Confirm

3. Inside your provider [Create a Channel] for a **Messaging API**. Choose a name, description and image for your Channel API (this info will be seen by Line users). Email and ToS can be changed or use the default ones.



Messaging API
Create new channel

Enter channel information | Confirm | Done

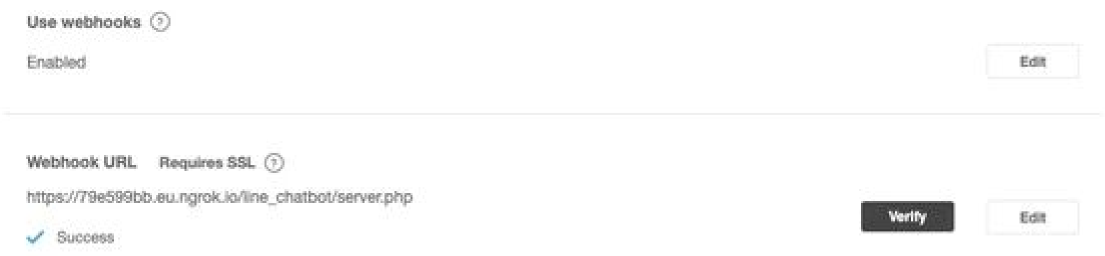
Enter information for the Messaging API

Selected Provider Test provider

App icon
optional
Register
Under 3MB; JPEG/PNG/GIF/BMP

App name
Test Messaging API
Max: 20 characters

4. Click on your channel (in this example, "Test Messaging API") and stay on "Channel Settings" tab.
5. You will need this "**Channel ID**" and "**Channel Secret**" to connect this Line Channel to an Inbenta chatbot *(See the section: Create ExtraInfo object with Line's Channel ID and Secret)*.
6. Enable the **Use of webhooks** and fill in your **Webhook URL** (requires SSL).



Use webhooks

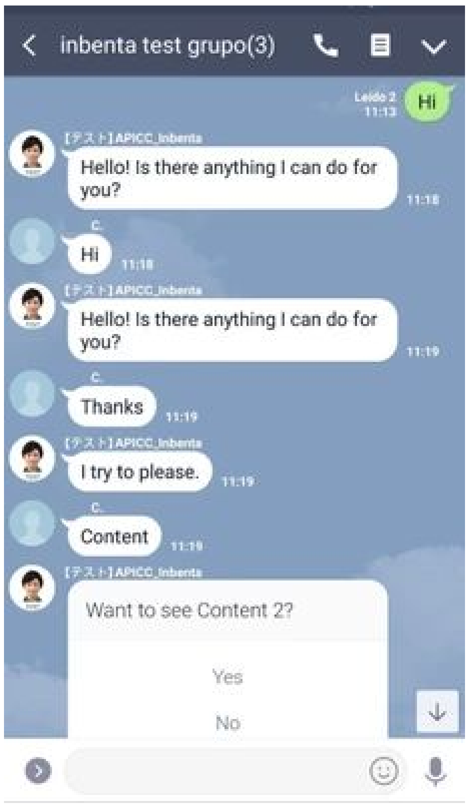
Enabled

Webhook URL Requires SSL

https://79e599bb.eu.ngrok.io/line_chatbot/server.php

✓ Success

7. **Disable Auto-reply** messages.
8. You can choose to disable **Greeting** messages or set a custom Greeting for when the user connects with the bot for the first time.
9. At the bottom of the page, you can find a QR code that will give the users quick access to the bot.
10. To use the bot in chats with multiple users (groups or rooms), enable the configuration **Allow bot to join group chats**.



inbenta test grupo(3)

Hi

[テスト]APICC_Inbenta: Hello! Is there anything I can do for you? 11:18

C.: Hi 11:18

[テスト]APICC_Inbenta: Hello! Is there anything I can do for you? 11:19

C.: Thanks 11:19

[テスト]APICC_Inbenta: I try to please. 11:19

C.: Content 11:19

[テスト]APICC_Inbenta: Want to see Content 2?

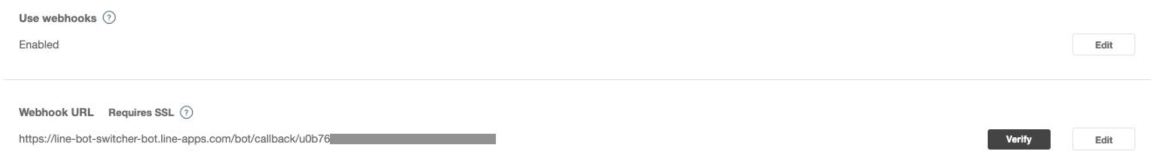
Yes

No

Line Application with Switch

Line's Switching API capability is still to be officially launched by Line and at the moment there's no open way to get it. It needs to be requested to and provided by Line case by case.

1. After you have been granted access to a provider and **Channel with Switcher API**, go to the Channel API and [Channel settings] tab.
2. In the Channel settings you should see the options "Use Webhooks" → enabled, and "Webhook Url" → a url pointing to ***"https://line-bot-switcher.../callback/..."*** or something similar. *DO NOT EDIT THIS WEBHOOK URL.*



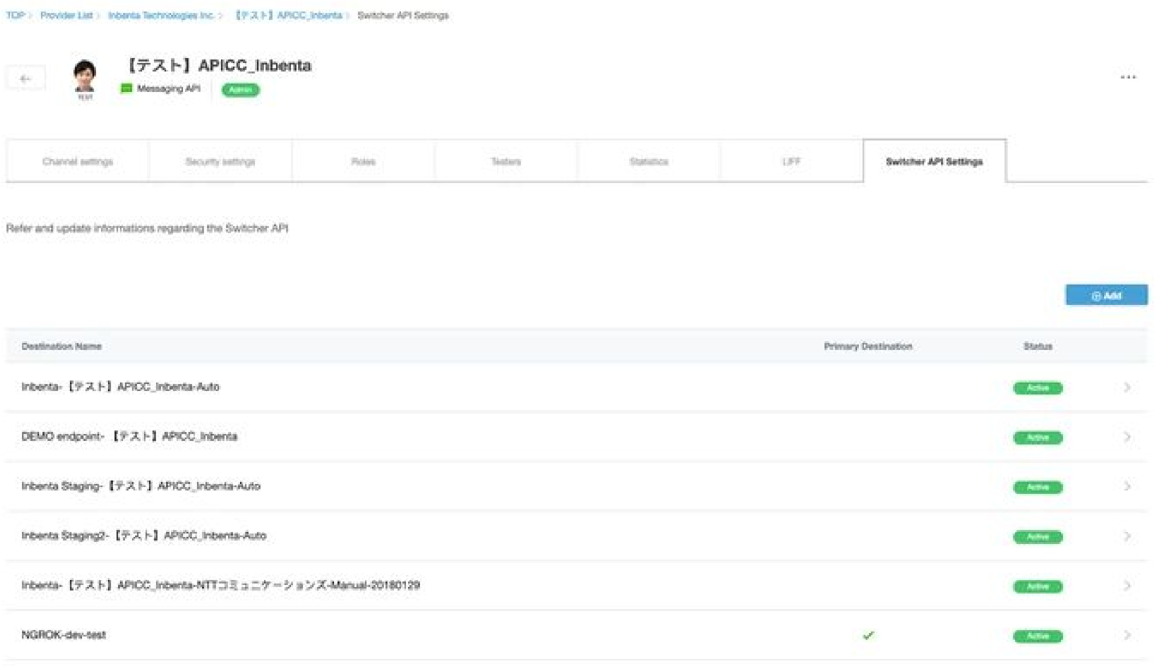
Use webhooks

Enabled

Webhook URL Requires SSL

https://line-bot-switcher-bot.line-apps.com/bot/callback/u0b76[redacted]

3. You will need the "**Channel ID**" and "**Channel Secret**" of this tab to connect this Line Channel to an Inbenta chatbot *(See the section: Create ExtraInfo object with Line's Channel ID and Secret)*.
4. To configure the Switcher, go to the tab **[Switcher API Settings]**. In this tab you will see a list of the different "destinations" (endpoints) the Line API can re-direct the messages and events sent by the users and received in the Line Channel:



TOP > Provider List > Inbenta Technologies Inc. > 【テスト】APICC_Inbenta > Switcher API Settings

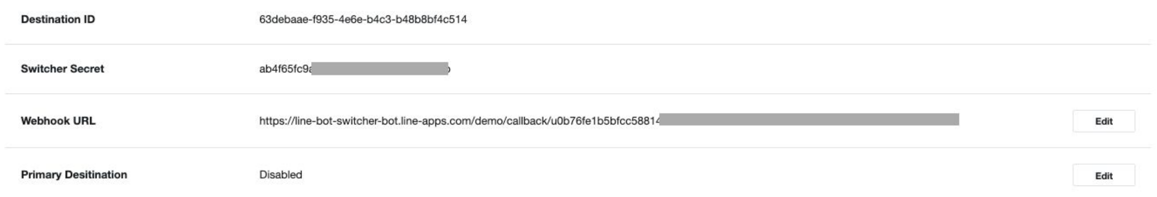
【テスト】APICC_Inbenta

Refer and update informations regarding the Switcher API

Destination Name	Primary Destination	Status
Inbenta-【テスト】APICC_Inbenta-Auto		Active
DEMO endpoint-【テスト】APICC_Inbenta		Active
Inbenta Staging-【テスト】APICC_Inbenta-Auto		Active
Inbenta Staging2-【テスト】APICC_Inbenta-Auto		Active
Inbenta-【テスト】APICC_Inbenta-NTTコミュニケーションズ-Manual-20180129		Active
NGROK-dev-test	✓	Active

5. Select (or add) the destination where you want to configure the *Line-Chatbot Connector*. Set this destination as the **Primary Destination**.
6. Set in this Primary Destination the **Webhook Url** pointing to your *Line-Chatbot Connector* and set the destination status to **active**.
7. Select (or add) the **secondary destination** where the messages flow will be directed to after the **Switching** is triggered (i.e.: Escalation). The status should be set to *Active*.

8. The Webhook URL of this secondary destination should be “***https://line-bot-switcher.../callback/…”*** or something similar if a Line agent or Bot if behind this destination. DO NOT EDIT THIS WEBHOOK (note: a url pointing to a non-line endpoint could, in theory, be used also).



Destination ID	63debaae-f935-4e6e-b4c3-b48b8bf4c514	
Switcher Secret	ab4f65fc9[redacted]	
Webhook URL	https://line-bot-switcher-bot.line-apps.com/demo/callback/u0b76fe1b5bfcc5881[redacted]	Edit
Primary Desitination	Disabled	Edit

9. You will need to set the "**Destination ID**" (and maybe "Switcher Secret" if switcher auth is enabled) of this secondary destination in the Inbenta Chatbot instance extraInfo *(See the section: Create ExtraInfo object with Line's Channel ID and Secret)*.

Building a Line-Chatbot Connector APP

Setup Chatbot Line Template

Download Line Template

You can find the PS Chatbot Line template in GitHub. Download it from https://github.com/inbenta-integrations/line_chatbot_template and move it into your repository.

Configuration (required)

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

- **CHATBOT-Line basic connection** (api.php)
    - The basic line-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 Line's Channel that will be used, by setting the Channel ID/Secret pair that will be retrieved from extraInfo *(See the section: "Create ExtraInfo object with Line's Channel ID and Secret")*
        2. The logging environment in the chatbot instance for the events in the conversation.

Configuration (Switcher and advanced)

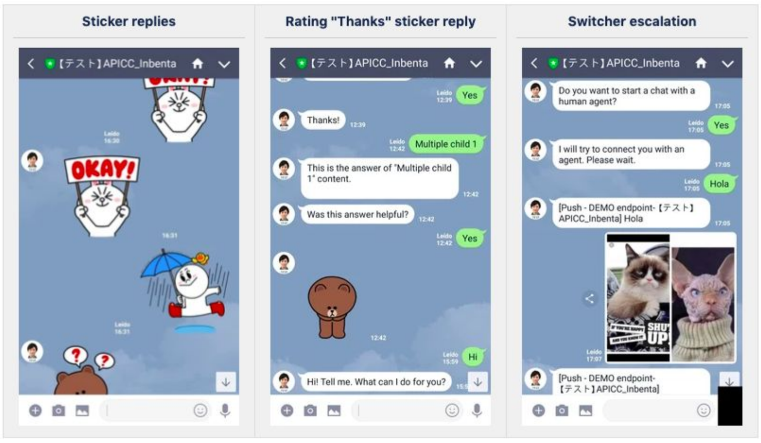
There are several advanced features that can be enabled using the optional configuration files *chat.php* and *conversation.php*.

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 Line 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.
        - **sticker**: Enable the use of stickers instead of a "Thanks!" answer to ratings. Thes stickers to be used are set in *thanksRatingStickers.*
        - **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 Line*).
        - **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.
    - **stickerReplies:**
        - **availablePackages**: (Array) if the sticker received belongs to one of the official packages set in this array, the bot will answer with the same sticker. Source: https://developers.line.biz/media/messaging-api/sticker_list.pdf
        - **unknownStickerReplies**: (Array) if the sticker received does not belongs to one of the official packages, the bot will answer with one of the stickers set in this array
        - **thanksRatingStickers**: (Array) if content_ratings->sticker is set to true, the bot will answer with one of the stickers set in this array to a rating received.
- **SWITCHER** (chat.php)
    - **chat**
        - **enabled**: Enables or disables Chat.
        - **useExternal**: True by default. It tells the connector that a non-Hyperchat external chat service should be used on escalation, in this case the Lines Switcher API.
    - **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.*



Yes
Thanks!
Multiple child 1
This is the answer of "Multiple child 1" content.
Was this answer helpful?
Yes
Hi
Hi! Tell me. What can I do for you?

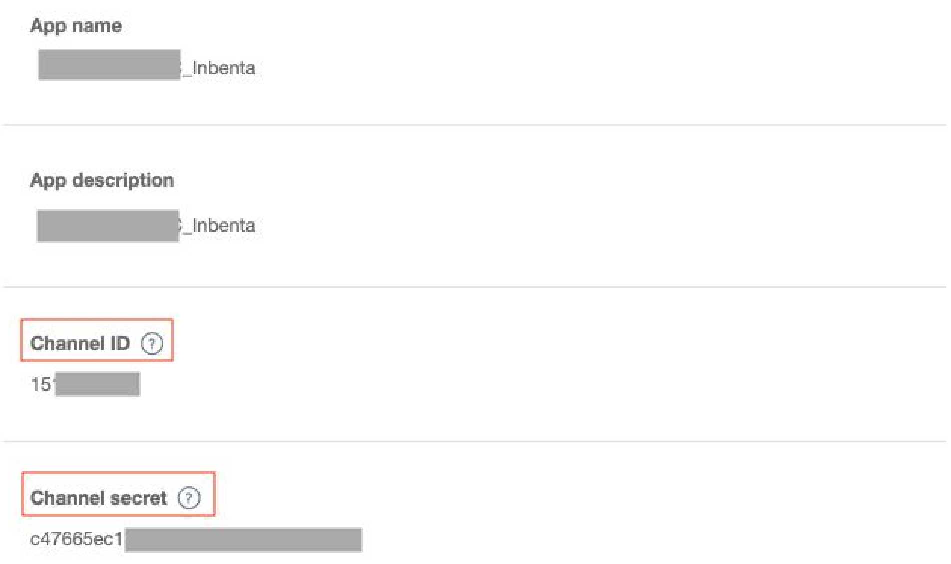
Do you want to start a chat with a human agent?
Yes
I will try to connect you with an agent. Please wait.
Hola
[Push - DEMO endpoint-【テスト】APICC_Inbenta] Hola
[Push - DEMO endpoint-【テスト】APICC_Inbenta]

Inbenta's backstage instances setup

Chatbot Instance

Create ExtraInfo object with Line's Channel ID 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 **'line'**.
3. While still in the "Manage groups and types" modal, click on [Add type] inside the "Line" group section:
    a. Name the new type '**line_conf**'.
    b. Make sure the group is 'line'.
    c. Create 3 properties with names '**development**', '**preproduction**' and '**production**' and type *multiple*.
    d. Inside each of those 3 properties, create 5 type *text* sub-properties with names:
        i. **channel_id** (required): Id of the Line's Channel where messages and event will be sent to.
        ii. **channel_secret** (required): Secret of the Line's Channel. The Id and Secret will be used to generate an access token.



App name

[redacted]_Inbenta

App description

[redacted]_Inbenta

Channel ID

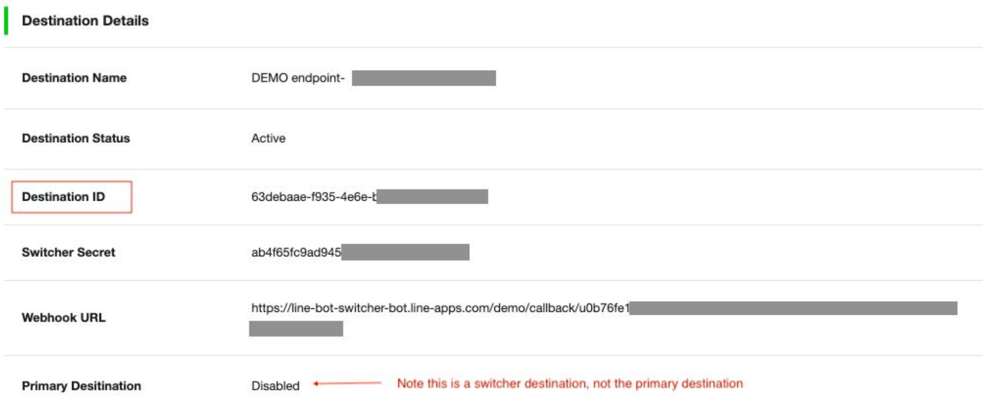
15[redacted]

Channel secret

c47665ec1[redacted]

        iii. **switcher_destination** (if using Line's Switcher API) - **Destination ID** that identifies the destination where the messages will be sent to after Line's Switch capability has been triggered.

iv. **switcher_secret** (optional) - For auth purposes in the *switcher_destination*.



Destination Details

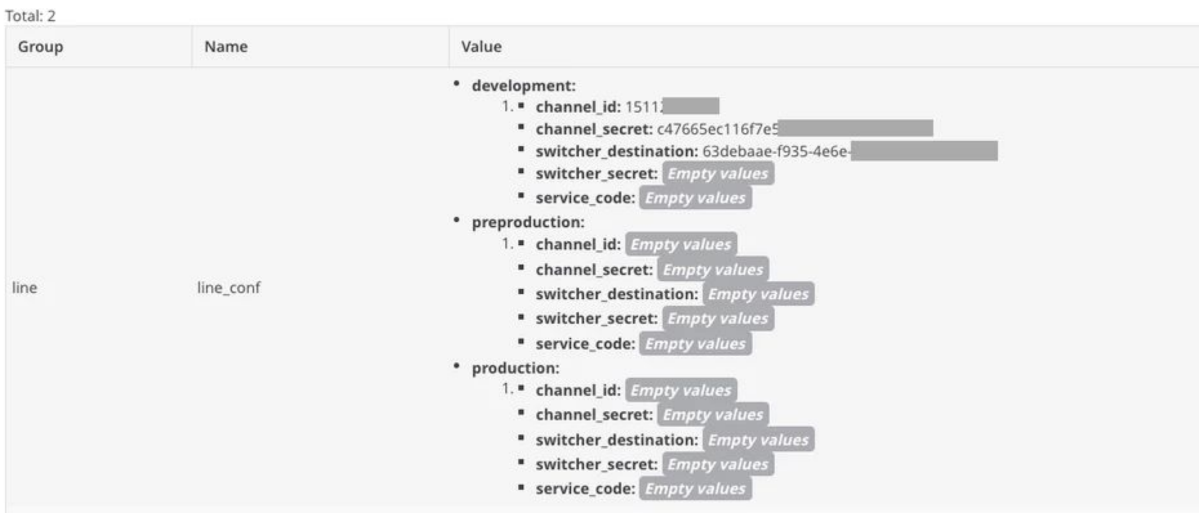
Destination Name	DEMO endpoint- [redacted]
Destination Status	Active
Destination ID	63debaae-f935-4e6e-b[redacted]
Switcher Secret	ab4f65fc9ad945[redacted]
Webhook URL	https://line-bot-switcher-bot.line-apps.com/demo/callback/u0b76fe1[redacted]
Primary Desitination	Disabled ← Note this is a switcher destination, not the primary destination

v. **service_code** (optional) - For Line statistics.

e. Click on [OK] to save your 'line_conf' object.

4. While in "Knowledge" → "Extra Info", click on [New Entry]:
   a. Name the entry '**line_conf**'.
   b. Select the group '**line**' and the type '**line_conf**'.
   c. Insert the Line Channel **ID/Secret**, the secondary **switcher Destination** if you are going to use the Switcher API, and optionally the Switcher Secret and Service Code if required *(See the section: Create a Line Application)*.
5. Note that you can have different Line Channels 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:



Total: 2

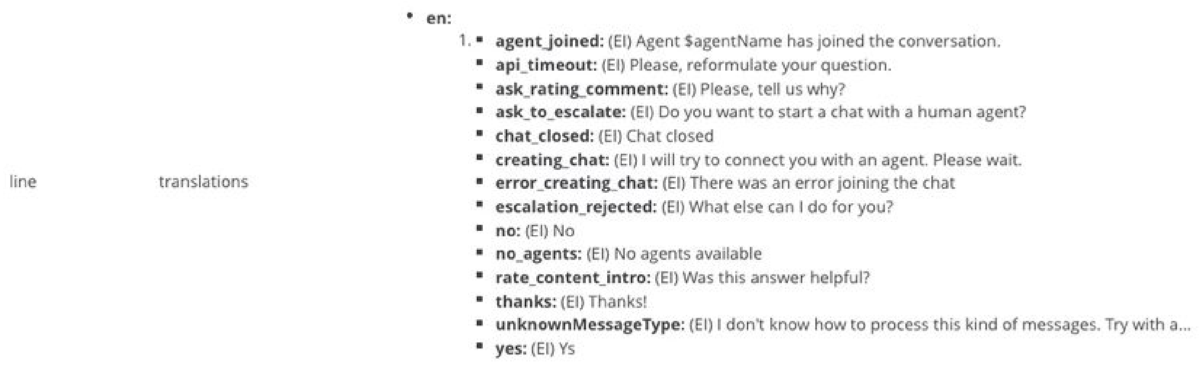
Group	Name	Value
line	line_conf	• development: 1. ▪ channel_id: 1511[redacted] ▪ channel_secret: c47665ec116f7e5[redacted] ▪ switcher_destination: 63debaae-f935-4e6e-[redacted] ▪ switcher_secret: Empty values ▪ service_code: Empty values • preproduction: 1. ▪ channel_id: Empty values ▪ channel_secret: Empty values ▪ switcher_destination: Empty values ▪ switcher_secret: Empty values ▪ service_code: Empty values • production: 1. ▪ channel_id: Empty values ▪ channel_secret: Empty values ▪ switcher_destination: Empty values ▪ switcher_secret: Empty values ▪ service_code: Empty values

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 'line' group (if there is no 'line' group, create it).
    a. Name the new type '**translations**'.
    b. Make sure the group is 'line'.
    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!'
        - **unknownMessageType** => 'I don\'t know how to process this kind of messages. Try with another question.',
        - **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 '**line**' 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:

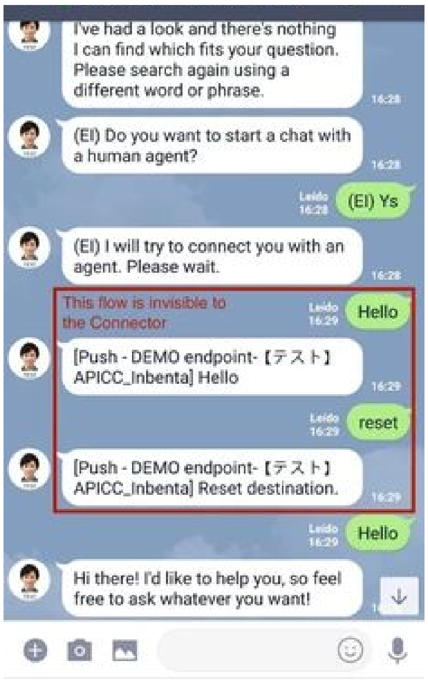


line translations

- **en:**
    1.
        - **agent_joined:** (EI) Agent $agentName has joined the conversation.
        - **api_timeout:** (EI) Please, reformulate your question.
        - **ask_rating_comment:** (EI) Please, tell us why?
        - **ask_to_escalate:** (EI) Do you want to start a chat with a human agent?
        - **chat_closed:** (EI) Chat closed
        - **creating_chat:** (EI) I will try to connect you with an agent. Please wait.
        - **error_creating_chat:** (EI) There was an error joining the chat
        - **escalation_rejected:** (EI) What else can I do for you?
        - **no:** (EI) No
        - **no_agents:** (EI) No agents available
        - **rate_content_intro:** (EI) Was this answer helpful?
        - **thanks:** (EI) Thanks!
        - **unknownMessageType:** (EI) I don't know how to process this kind of messages. Try with a...
        - **yes:** (EI) Ys

Important note on the Switcher API

Once the Switching process has been triggered, the Chatbot-Line Connector will remain idle and will be blind to all events and messages in the new flow between the user and the secondary destination. Inbenta's chatbot backstage instance will, therefore, not receive any data either after the Switching. The last action the Connector is aware of is the postback *"Yes" (I want to escalate),* to which it responds with: "I will now try to connect you with an agent". After that, the Connector will not know when or if an agent is actually responding in the secondary destination. It will not know either when the Switching process has been reverted and the flow of the conversation passes agin through itself. It will only be aware of it if the user sends a new message that this time will land in the Connector again.



I've had a look and there's nothing I can find which fits your question. Please search again using a different word or phrase. 16:28

(El) Do you want to start a chat with a human agent? 16:28

Leido 16:28 (El) Ys

(El) I will try to connect you with an agent. Please wait. 16:28

This flow is invisible to the Connector

Leido 16:29 Hello

[Push - DEMO endpoint- 【テスト】 APICC_Inbenta] Hello 16:29

Leido 16:29 reset

[Push - DEMO endpoint- 【テスト】 APICC_Inbenta] Reset destination. 16:29

Leido 16:29 Hello

Hi there! I'd like to help you, so feel free to ask whatever you want!

Escalation tracking events

- **Contact rejected**: The Line Connector will log a CONTACT_REJECTED track if the user responds *"No"* to an escalation offer.
- **Contact attended**: Since the last event the Line Connector is "aware" of during the escalation process is the the postback *"Yes" (I want to escalate),* to which it answers with: "I will now try to connect you with an agent", at this point the connector will log a CONTACT_ATTENDED track, no matter whether the petition is actually attended or not at the other side of the Line Switch.

        - **UNUSED - Contact unattended:** Since the Connector will not know when or if an agent is actually responding in the secondary Switch destination, it is assumed that the escalation is always responded and, therefore, the CONTACT_UNATTENDED track is never logged.

Due to the fact that Once the Switching process has been triggered, the Chatbot-Line Connector will remain idle and will be blind to all events and messages in the new flow between the user and the secondary destination.

Relevant URL's

- Line developers (and login) <https://developers.line.biz/>
- Line messaging API docs <https://developers.line.biz/en/reference/messaging-api/>
- Line official stickers <https://developers.line.biz/media/messaging-api/sticker_list.pdf>

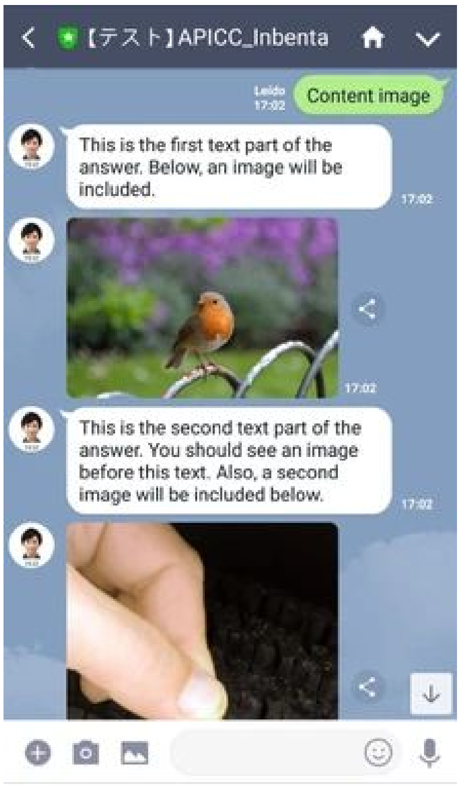
Troubleshooting

- **Bot is not answering (Line webhook configuration)**: Check if your Line-Chatbot Connector App URL is configured in your 'Line Channel' →'Channel Settings' settings has the 'Webhook Url' configured and webhook is enabled.
- **Bot is not answering (Chatbot backstage configuration):** Make sure your Line Channel ID and Secret are correctly configured in your instance ExtraInfo and are valid.
- **Switcher → bot is not working (Line Switcher configuration):** Check in the Line's Channel Switcher configuration the primary webhook url points to your Chatbot-Line connector and is enabled.
- **Switcher → switch to secondary destination (escalation) is not working (connector configuration):** Make sure that conf/custom/chat.php you have 'enabled' → *true* and 'useExternal' → *true*.
- **Switcher → switch to secondary destination (escalation) is not working (backstage configuration):** Make sure that the secondary Switcher Destination is correctly configured in your instance ExtraInfo and is valid.

How to test it?

Do the previous steps in order to configure the LINE application account, the Backstage instance and the UI. Then, do the following:

1. Install the LINE application in a mobile phone.
2. Log in with your LINE account.
3. Add the LINE application account as a friend, for example, using the QR code available in the LINE application account settings.
4. Start a conversation with the bot and test it.

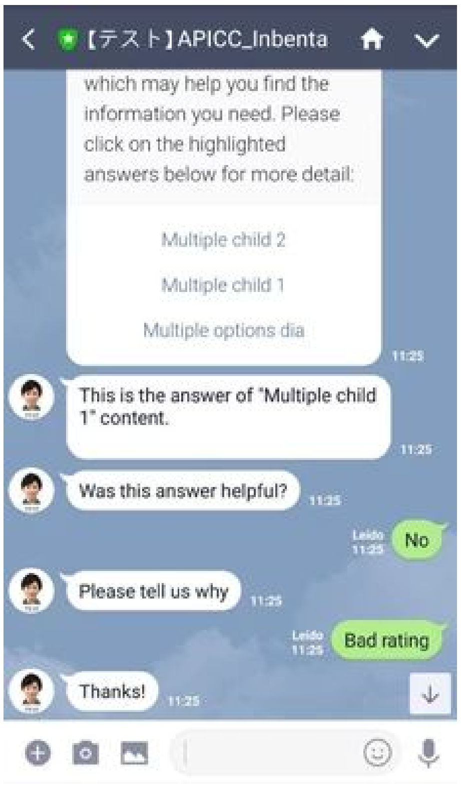


Leido 17:02 Content image

This is the first text part of the answer. Below, an image will be included. 17:02

17:02

This is the second text part of the answer. You should see an image before this text. Also, a second image will be included below. 17:02



which may help you find the information you need. Please click on the highlighted answers below for more detail:

Multiple child 2

Multiple child 1

Multiple options dia

11:25

This is the answer of "Multiple child 1" content. 11:25

Was this answer helpful? 11:25

Leido 11:25 No

Please tell us why 11:25

Leido 11:25 Bad rating

Thanks! 11:25