

# EN Module Reference

## Table of Contents

### Introduction

- [About this document](#)
- [What is a module channel?](#)

### Link (attach) module channels

- [Attach module channels using the OAuth 2.0 authorization mechanism](#)
  - [Flow diagram](#)
  - [1. Request authorization from the LINE Official Account admin](#)
  - [2. About the linkage screen](#)
  - [3. Receive the authorization code](#)
  - [4. Linkage \(attach\) by operation of the module channel provider](#)
- [Unlink \(detach\) the module channel by the operation of the module channel administrator](#)

### Configure module channel settings

- [1. module tab](#)
- [2. Webhook settings](#)
  - [Webhook URL](#)
  - [Using webhook](#)
  - [Resend webhook](#)
  - [Error stats](#)
- [3. Redirect settings](#)
  - [Redirect URL](#)

### Control channel initiative (Chat Control)

- [What is channel control \(Chat Control\)?](#)
- [Acquire Control API](#)
- [Release Control API](#)
- [Default Active](#)
  - [Auto active](#)
  - [Exclusive control](#)

### Using the Messaging API from the module channel

- [Using the Messaging API with the module channel's channel access token](#)
  - [User ID used in the module channel](#)
  - [Module channel's channel access token](#)
  - [Call the Messaging API endpoint](#)
  - [Receiving a webhook](#)
- [Receive module channel-specific Webhook events](#)
- [Get the LINE Official Account information from the module channel](#)
  - [Get bot information](#)
  - [Get information on multiple bots](#)

### Notes

# Introduction

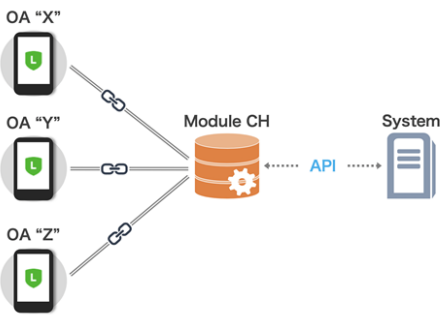
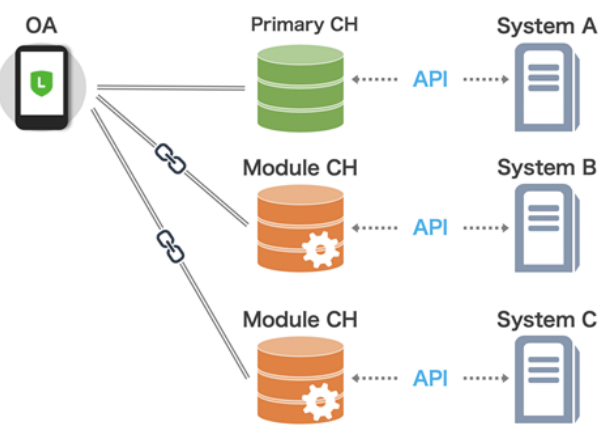
## About this document

Issue date	Version	History
April 1, 2022	v1.0	<ul style="list-style-type: none"><li>• Release version</li></ul>
April 1, 2022	Ver 1.1	<ul style="list-style-type: none"><li>• Correct details</li></ul>
April 19, 2022	Ver 1.2	<ul style="list-style-type: none"><li>• Add updated module channel descriptions and screenshots from the LINE Developers Console</li></ul>

## What is a module channel?

Module channels are a new type of channel that you can link (attach) to your existing LINE Official Account.

In general, there are two main ways to use the system.

<b>Attach the same service (module channel) to multiple LINE Official Accounts (OA "X," OA "Y," OA "Z")</b>	<b>Attach multiple services (module channels) to a single LINE Official Account (OA)</b>
<p>For example, even if you're managing separate LINE Official Accounts for each store, you can provide the same functionality to all stores by simply preparing one module channel.</p>  <p>Legend</p> <ul style="list-style-type: none"> <li>• <b>OA "X", OA "Y", OA "Z"</b>: LINE Official Account</li> <li>• <b>Module CH</b>: Module channel</li> <li>• <b>System</b>: *Module CH Webhook destination and bot server</li> </ul>	<p>You can provide various functions to end users while switching between services.</p>  <div style="border: 1px solid red; padding: 10px; margin-top: 10px;"> <p><b>⚠ These methods aren't currently available:</b></p> <ul style="list-style-type: none"> <li>• Using the Messaging API from the Messaging API channel of the LINE Official Account attached to the module channel</li> <li>• Attaching multiple module channels to one LINE Official Account</li> </ul> </div> <p>Legend</p> <ul style="list-style-type: none"> <li>• <b>OA</b>: LINE Official Account</li> <li>• <b>Primary CH</b>: Primary Channel (Standard Messaging API channel linked to the LINE Official Account)</li> <li>• <b>Module CH</b>: Module channel</li> <li>• <b>System A</b>: Primary CH Webhook destination and bot server</li> <li>• <b>System B, C</b>: *Module CH Webhook destination and bot server</li> </ul>

\*The server is prepared by the company in charge of developing the module channel. In the figure, the server set as the Webhook destination and the server that calls the Messaging API, etc., are depicted as the same, but they don't have to be the same.

## Link (attach) module channels

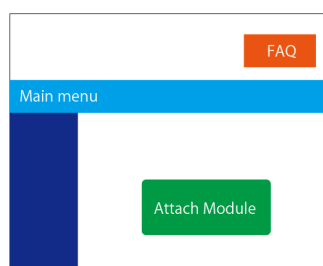
To use the module channel feature, the admin of the LINE Official Account needs to attach the module channel by following these steps:

### Attach module channels using the OAuth 2.0 authorization mechanism

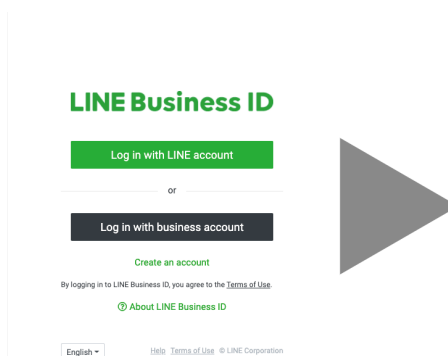
Following the flow of the OAuth 2.0 authorization mechanism, the admin of the LINE Official Account can connect (attach) the module channel.

#### Flow diagram

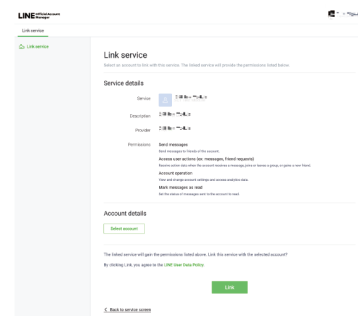
The first screen and the fifth screen should be prepared by the company in charge of developing the module channel.



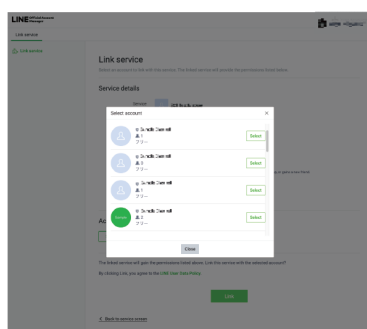
① 【In Your Service】 Click to Attach



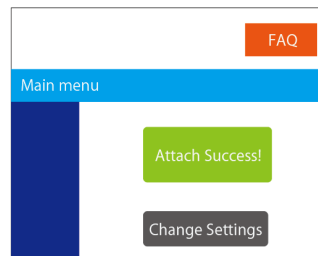
② LINE Official Account Manager(OAM) Login



③【OAM】 Select Attach Account



④ 【OAM】 Confirm and Attach(LINK)



⑤ 【In Your Service】 Attach done

#### **Restrictions on attaching multiple module channels to the LINE Official Account**

Module channels with the Default Active feature can't be attached to a single LINE Official Account.

## 1. Request authorization from the LINE Official Account admin

By having the admin of the LINE Official Account access the URL for authentication and authorization (authentication URL `https://manager.line.biz/module/auth/v1/authorize` with query parameters), the process of attaching the module channel to your LINE Official Account will begin.

### Example URL for authentication and authorization

```
https://manager.line.biz/module/auth/v1/authorize?response_type=code&client_id=1234567890&redirect_uri=https%3A%2F%2Fexample.com%2Fcallback&scope=message%3Asend%20message%3Areceive&state={CSRF token}&region=JP&basic_search_id={LINE Official Account basic ID}&brand_type=premium
```

Generally, you'll set a link to access this URL on the page to start linking module channels, and then ask the LINE Official Account admin to click the link. In the flow chart in the above example, the URL can be accessed when you click the "Attach Module" button on the "In Your Service" link start page.

Query parameters	Required	Description				
response_type	Required	code				
client_id	Required	The channel ID of the module channel. A channel-specific identifier issued by LINE.				
redirect_uri	Required	<p>Redirect URL. The URL (URL-encoded value) for the module channel developer to receive the authorization code. After authentication and authorization (operation on the link screen), the LINE Official Account admin will be redirected to this URL.</p> <p>This URL should be provided by the module channel developer. This URL must match the <a href="#">redirect URL</a> that you previously registered for the module channel in the LINE Developers Console.</p>				
scope	Required	<p>Specify at permission (scope) that you want to request the LINE Official account admin to allow. These are the scopes that can be specified. To specify multiple scopes, separate them with a URL-encoded space (%20).</p> <table><tr><th>Scope</th><th>APIs available for the module channel</th></tr><tr><td>Not specified (default)</td><td><ul style="list-style-type: none"><li><a href="#">Issue link token (/v2/bot/user/{userId}/linkToken)</a>.</li></ul><p>*It can be used without a scope.</p></td></tr></table>	Scope	APIs available for the module channel	Not specified (default)	<ul style="list-style-type: none"><li><a href="#">Issue link token (/v2/bot/user/{userId}/linkToken)</a>.</li></ul> <p>*It can be used without a scope.</p>
Scope	APIs available for the module channel					
Not specified (default)	<ul style="list-style-type: none"><li><a href="#">Issue link token (/v2/bot/user/{userId}/linkToken)</a>.</li></ul> <p>*It can be used without a scope.</p>					

Query parameters	Required	Description	
		Scope	APIs available for the module channel
		message%3Asend (message:send)	<ul style="list-style-type: none"> <li>• <a href="#">Send reply message (/v2/bot/message/reply)</a></li> <li>• <a href="#">Send push message (/v2/bot/message/push)</a></li> <li>• <a href="#">Send multicast message (/v2/bot/message/multicast)</a></li> <li>• <a href="#">Send broadcast message (/v2/bot/message/broadcast)</a></li> <li>• <a href="#">Send narrowcast message (/v2/bot/message/narrowcast)</a> and related APIs</li> <li>• <a href="#">Managing audience (/v2/bot/audienceGroup/***)</a></li> <li>• <a href="#">Get the target limit for additional messages (/v2/bot/message/quota)</a></li> <li>• <a href="#">Get number of messages sent this month (/v2/bot/message/quota/consumption)</a></li> </ul>
		message%3Areceive (message:receive)	<ul style="list-style-type: none"> <li>• Receive Webhook events for Messaging API and Module Channel API               <ul style="list-style-type: none"> <li>• <a href="#">Webhook</a></li> <li>• <a href="#">Webhook event objects</a></li> </ul> </li> <li>• <a href="#">Channel control (Chat Control)</a></li> </ul>
		account%3Amanage (account:manage)	<ul style="list-style-type: none"> <li>• <a href="#">Set default rich menu (/v2/bot/user/all/richmenu/{richMenuId})</a></li> <li>• <a href="#">Get number of message deliveries (/v2/bot/insight/message/delivery?date={date})</a></li> <li>• <a href="#">Get number of followers (/v2/bot/insight/followers?date={date})</a></li> <li>• <a href="#">Get friend demographics (/v2/bot/insight/demographic)</a></li> <li>• <a href="#">Get user interaction statistics (/v2/bot/insight/message/event?requestId={requestId})</a></li> </ul>
		message%3Amark_as_read (message:mark_as_read)	<ul style="list-style-type: none"> <li>• <a href="#">Mark messages from users as read (/v2/bot/message/markAsRead)</a></li> </ul>

Query parameters	Required	Description	
		Scope	APIs available for the module channel
		profile%4Aread (profile:read)	<ul style="list-style-type: none"> <li>• <a href="#">Get profile (/v2/bot/profile/{userId})</a>.</li> <li>• <a href="#">Get group summary (/v2/bot/group/{groupId}/summary)</a></li> <li>• <a href="#">Get group member profile (/v2/bot/group/{groupId}/member/{userId})</a></li> <li>• <a href="#">Get room member profile (/v2/bot/room/{roomId}/member/{userId})</a></li> <li>• <a href="#">Get number of users in a group (/v2/bot/group/{groupId}/members/count)</a></li> <li>• <a href="#">Get number of users in a room (/v2/bot/room/{roomId}/members/count)</a></li> </ul>
		crm%3Amanage (crm:manage)	<ul style="list-style-type: none"> <li>• Required only when using Chat Plugin. Otherwise, don't specify. The Chat Plugin is currently only available to select corporate users.</li> </ul> <div> <p><b>i</b></p> <ul style="list-style-type: none"> <li>• This scope can only be specified for module channels that use the Chat Plugin function.</li> <li>• If this scope isn't specified for a module channel that uses the Chat Plugin functions, the functions provided by the Chat Plugin may not be available in the future.</li> </ul> </div>
state	Required	A unique alphanumeric string to prevent <a href="#">cross-site request forgery</a> (CSRF). The system should randomly generate unique values in the system of the company responsible for the development of the module channel. URL-encoded strings can't be used.	
region	Optional	The region of the LINE Official Account to which the module channel is attached. Specify JP or TW .	
basic_sear ch_id	Optional	LINE Official Account basic ID. Specify to limit the number of LINE Official Accounts to which the module channel is attached.	

Query parameters	Required	Description
<code>brand_type</code>	Optional	<p>Specify to limit the account types of LINE Official Accounts that can be attached.</p> <p>LINE Official Account Account Type  <a href="https://www.linebiz.com/jp/service/line-official-account/account-type/">https://www.linebiz.com/jp/service/line-official-account/account-type/</a></p> <ul style="list-style-type: none"> <li>Premium Account: <code>premium</code></li> <li>Verified Account: <code>verified</code></li> <li>Unverified Account: <code>unverified</code></li> </ul> <p>To specify multiple account types, concatenate whitespace characters with URL-encoded %20.</p> <p>For example, to limit the attachment of only premium accounts and authenticated accounts, you would specify:</p> <pre>brand_type=premium%20verified</pre>
<code>code_challenge</code>	Optional	<p>Specify when using PKCE (Proof Key for Code Exchange) defined in the OAuth2.0 extension specification as a countermeasure against authorization code interception attacks.</p> <p>Compliant with RFC 7636.</p>
<code>code_challenge_method</code>	Optional	<p>S256</p> <p>Specify when using PKCE (Proof Key for Code Exchange) defined in the OAuth2.0 extension specification as a countermeasure against authorization code interception attacks.</p> <p>Compliant with RFC 7636.</p>

### Specify the URL-encoded value for `redirect_uri`

The value specified for `redirect_uri` should be URL-encoded.

**Example of specifying <https://example.com/auth?param1=value1&param2=value2> as the `redirect_uri` in the authentication URL**

```
https://manager.line.biz/module/auth/v1/authorize?response_type=code&client_id=1234567890&redirect_uri=https%3A%2F%2Fexample.com%2Fauth%3Fparam1%3Dvalue1%26param2%3Dvalue2&scope=message%3Asend%20message%3Areceive&state={CSRF token}&region=JP&basic_search_id={LINE Official Account basic id}&brand_type=premium
```

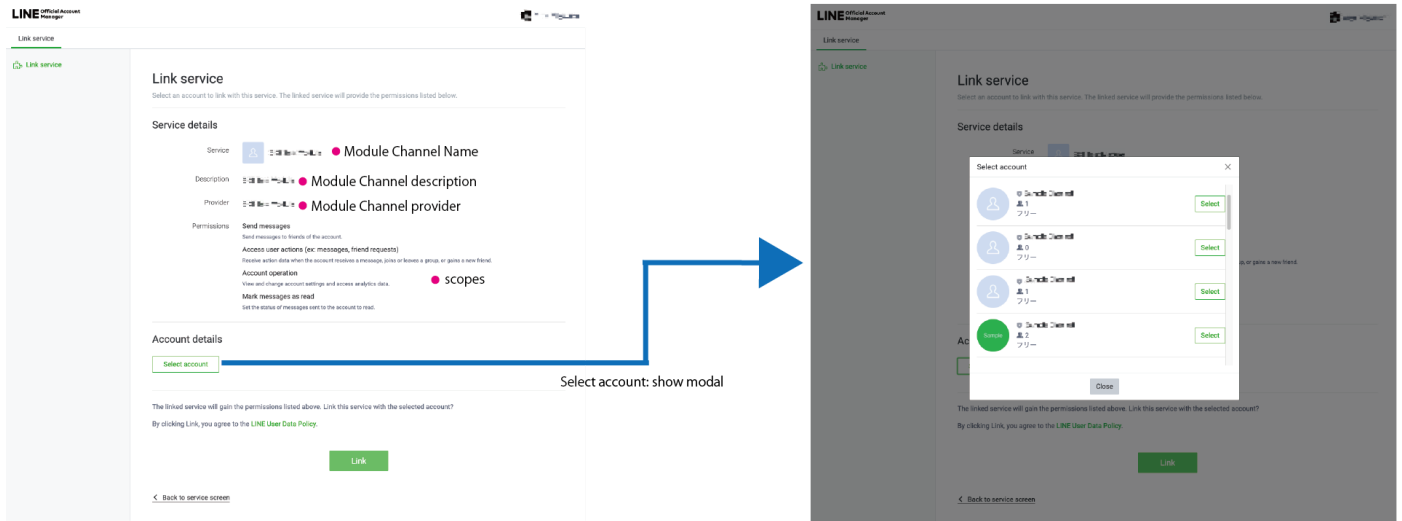
In particular, if you forget the URL encoding of the query parameters, the second and subsequent query parameters will be recognized as query parameters for the authentication URL and won't be passed to the redirect destination.

## 2. About the linkage screen

When the admin of the LINE Official Account accesses the URL for verification and authorization, the LINE Official Account Manager linkage screen will be displayed. The linkage screen shows what you applied for when creating a



module channel. You can check the settings in the [LINE Developers Console](#).



### 3. Receive the authorization code

When the admin of the LINE Official Account completes authentication and authorization, the authorization code and error code are passed to the redirect URL ( `redirect_uri` ) specified in the URL for authentication and authorization through these query parameters. In the flow diagram presented in the above example, the authorization code and error code are passed when the [Link] button is clicked on the "[OAM] Confirm and Attach" screen.

#### When authentication is successful

After successful authentication by the admin of the LINE Official Account, you'll be redirected to the redirect URL ( `redirect_uri` ) with these query parameters.

Query parameters	Required	Description
<code>code</code>	Required	This is the authorization code required to link (attach) to the LINE Official Account. This authorization code has an expiration date and can be used only once.
<code>state</code>	Required	Anti-CSRF string specified in the <code>state</code> query parameter of the URL for authentication and authorization. Make sure they are the same string.

#### When authentication fails

If authentication by the admin of the LINE Official Account fails, you'll be redirected to the redirect URL ( `redirect_uri` ) with these query parameters.

Query parameters	Required	Description
<code>error</code>	Required	Error code.
<code>error_description</code>	Optional	Error details.

Query parameters	Required	Description
state	Required	Anti-CSRF string specified in the state query parameter of the URL for authentication and authorization.  Make sure they are the same string.

## 4. Linkage (attach) by operation of the module channel provider

Once you get the authorization code and confirm that the string passed in the `state` query parameter is okay, attach the module channel to your LINE Official Account.

When using this API, you need to specify the channel ID and channel secret of the module channel using either the Authorization header or the request body.

### HTTP request

POST `https://manager.line.biz/module/auth/v1/token`

### Request header

<b>Content-Type</b>	<code>application/x-www-form-urlencoded</code>
<b>Authorization</b>	<p>(Only if you use the Authorization header to specify the channel ID and channel secret of the module channel)</p> <p><code>Basic {base64({Channel ID}:{Channel Secret})}</code></p> <p>For <code>{base64({Channel ID}:{Channel Secret})}</code>, specify a Base64-encoded string by concatenating "Module Channel ID" and "Module Channel Secret" with <code>:</code>. You can find the module channel's channel ID and channel secret in the LINE Developers Console.</p>

### Request body

<b>grant_type</b> String Required	<code>authorization_code</code>
<b>code</b> String Required	Specify the authorization code passed in the <code>code</code> query parameter when authentication is successful.
<b>redirect_uri</b> String Required	Specify the <code>redirect_uri</code> specified in <a href="#">1. Request authorization from the LINE Official Account admin</a> .
<b>code_verifier</b> String Optional	<p>Specify when using PKCE (Proof Key for Code Exchange) defined in the OAuth2.0 extension specification as a countermeasure against authorization code interception attacks.</p> <p>Compliant with RFC 7636.</p>
<b>client_id</b> String Optional	<p>(Only if the request body is used to specify the channel ID and channel secret of the module channel.)</p> <p>Specifies the channel ID of the module channel. You can find the channel ID of the module channel in the LINE Developers Console.</p>
<b>client_secret</b> String Optional	<p>(Only if the request body is used to specify the channel ID and channel secret of the module channel.)</p> <p>Specifies the channel secret of the module channel. You can find the channel secret of the module channel in the LINE Developers Console.</p>

<b>region</b> String Optional	If you specified a value for <code>region</code> in the URL for authentication and authorization, specify the same value again.
<b>basic_search_id</b> String Optional	If you specified a value for <code>basic_search_id</code> in the URL for authentication and authorization, specify the same value again.
<b>scope</b> String Optional	If you specified a value for <code>scope</code> in the URL for authentication and authorization, specify the same value again.
<b>brand_type</b> String Optional	If you specified a value for <code>brand_type</code> in the URL for authentication and authorization, specify the same value again.

## Response

Returns a JSON object with status code `200` and this information on success.

<b>bot_id</b> String	User ID of the bot on the LINE Official Account <ul style="list-style-type: none"> <li>The bot's user ID is used when calling the <a href="#">Messaging API</a> or the Acquire Control API.</li> </ul> <div> <b>Note</b>  The "Your user ID" displayed on the "Basic Settings" tab of the LINE Developers console for the Messaging API channel is not the bot's user ID. </div>
<b>scope</b> String	Permissions (scope) granted by the LINE Official Account admin

## Error response

When an error occurs, one of these status codes is returned:

Status code	Description
<code>400</code>	There is a problem with your request.
<code>403</code>	You don't have permission to access.

## Unlink (detach) the module channel by the operation of the module channel administrator

The module channel admin calls the Detach API to detach the module channel from a LINE Official Account.

### HTTP request

POST <https://api.line.me/v2/bot/channel/detach>

### Request header

<b>Authorization</b>	Bearer {Channel Access Token} <ul style="list-style-type: none"> <li>For {channel access token} , specify the module channel's channel access token.</li> </ul>
<b>Content-Type</b>	application/json

### Request body

<b>botId</b> String Required	User IDs of LINE Official Account bots attached to the module channel <ul style="list-style-type: none"> <li>You can get the user ID of the bot from the POST <a href="https://manager.line.biz/module/auth/v1/token">https://manager.line.biz/module/auth/v1/token</a> response or the Attached event.</li> </ul>
------------------------------------	--

### Response

Returns a 200 status code on success.

### Error response

When an error occurs, this status codes is returned:

Status code	Description
400	There is a problem with your request.

# Configure module channel settings

In the module channel, a dedicated "module" tab appears in the LINE Developers Console.

In the module tab, you can set the use of the module channel's webhook URL and webhook usage, and the `redirect_uri` specified in [Request authorization from the LINE Official Account admin](#) when attaching the module channel to the LINE official account.

**module test**

Admin | module

Basic **1** module Security Roles

### module settings

#### Webhook settings

Webhook URL ⓘ `https://example.com/module/webhook`

Use webhook ⓘ ☒

Webhook redelivery ⓘ ☐

Error statistics aggregation ⓘ ☐

#### Redirect settings

Redirect URL ⓘ `https://example.com/module/callback`

## 1. module tab

The "module" tab is a setting item dedicated only to the module channel.

## 2. Webhook settings

### Webhook URL

You can set one webhook URL for the module channel. See also [Receiving a webhook](#).

## Using webhook

---

You can set whether the module channel receives webhook events.

## Resend webhook

---

You can set whether or not to resend the webhook event from the LINE Platform when getting the webhook event fails in the webhook URL of the module channel.

## Error stats

---

You can set whether or not to display stats about webhook event reception failures on the Statistics tab.

# 3. Redirect settings

## Redirect URL

---

You can set the `redirect_uri` specified in [Request authorization from the LINE Official Account admin](#) when attaching a module channel to the LINE Official Account.

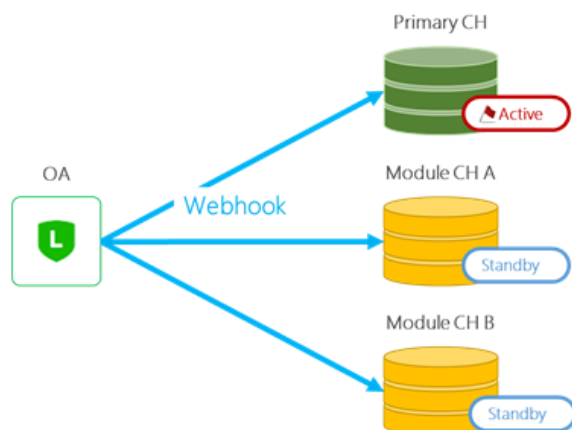
- The redirect URL scheme must be HTTPS.
- You can register multiple redirect URLs.

## Control channel initiative (Chat Control)

**⚠** The module channel currently offered automatically acquires channel initiative when it is attached to a LINE Official Account ([Default Active](#)), so there is no need to control channel initiative.

### What is channel control (Chat Control)?

To prevent multiple module channels from replying to or processing end user actions at the same time, we've introduced the concept of initiative (Chat Control) for module channels.



Initiative (Chat Control)	Description
Active Channel	<p>A channel with initiative (Chat Control). By default, Primary CH (the standard Messaging API channel associated with the LINE official account) is the "Active Channel".</p> <p><b>You can send reply messages, push messages, etc.</b> from this channel.</p> <p>Only one "Active Channel" can be created for each LINE Official Account.</p>
Standby Channel	<p>A channel that doesn't have Chat Control.</p> <p><b>Refrain from sending messages</b> from this channel.</p> <p>All channels other than Active Channel are "Standby Channel".</p>

**i Initiative (Chat Control) isn't a one-time setting for each module channel**

Initiative (Chat Control) is managed on a per-user, per-chat room, or per-group basis.

**i Module channel with "Default Active" function**

A module channel with the "Default Active" function is a module channel that automatically becomes an Active Channel when you attach it to your LINE Official Account.

See [Default Active](#) for details.



## Acquire Control API

**⚠** It isn't necessary to call this API in the currently provided module structure. So, the implementation of this API is optional.

This API is currently used only when the module channel initiative switches due to unexpected problems.

If the Standby Channel wants to take the initiative (Chat Control), it calls the Acquire Control API.

The channel that was previously an Active Channel will automatically switch to a Standby Channel.

### HTTP request

POST `https://api.line.me/v2/bot/chat/{chatId}/control/acquire`

### Request header

<b>Content-Type</b>	<code>application/json</code>
<b>Authorization</b>	Bearer <code>{Channel Access Token}</code> <ul style="list-style-type: none"> <li>For <code>{channel access token}</code>, specify the module channel's channel access token.</li> </ul>
<b>Private header</b>	User ID of the LINE Official Account bot attached to the module channel <ul style="list-style-type: none"> <li>You can get the user ID of the bot from the response of              POST <code>https://manager.line.biz/module/auth/v1/token</code> or the <a href="#">Attached event</a>.</li> </ul> <div> <b>⚠ The specific header name isn't disclosed</b>              The name (parameter name) of this header is open only to customers who participate in the <a href="#">LINE Marketplace</a>.           </div>

### Path parameter

<b>chatId</b>	<code>userId, roomId, groupId</code>
---------------	--------------------------------------

### Request body

<b>expired</b> <b>Boolean</b> Optional	<ul style="list-style-type: none"> <li><code>True</code> : After the time limit (ttl) has passed, the initiative (Chat Control) will return to the Primary Channel. (Default)</li> <li><code>False</code> : There's no time limit and the initiative (Chat Control) doesn't change over time.</li> </ul>
<b>ttl</b> <b>Number</b> Optional	The time it takes for initiative (Chat Control) to return to the Primary Channel (the time that the module channel stays on the Active Channel). The value is specified in seconds. The maximum value is one year (3600*24*365). The default value is <code>3600</code> (1 hour).  *Ignored if the value of <code>expired</code> is <code>false</code> .

### Response

Returns a `200` status code on success.

## Error response

When an error occurs, one of these status codes is returned:

Status code	Description
400	There is a problem with your request.
403	You don't have permission to access.
404	The target user isn't a friend.
423	Within a certain period of time, another channel has taken the initiative (Chat Control).

\*If another channel gets the initiative (Chat Control) within a certain period of time (a few seconds), this API will fail.

## Release Control API

**⚠** It isn't necessary to call this API in the currently provided module structure. So, the implementation of this API is optional.

This API is currently used only when the module channel initiative switches due to unexpected problems.

To return the initiative (Chat Control) of Active Channel to Primary Channel, call the Release Control API.

### HTTP request

POST `https://api.line.me/v2/bot/chat/{chatId}/control/release`

### Request header

<b>Authorization</b>	Bearer {Channel Access Token} <ul style="list-style-type: none"> <li>For {channel access token} , specify the module channel's channel access token.</li> </ul>
<b>Private header</b>	User ID of the LINE Official Account bot attached to the module channel <ul style="list-style-type: none"> <li>You can get the user ID of the bot from the response of            POST <code>https://manager.line.biz/module/auth/v1/token</code> or the <a href="#">Attached event</a>.</li> </ul> <div> <b>⚠ The specific header name isn't disclosed</b>            The name (parameter name) of this header is open only to customers who participate in the <a href="#">LINE Marketplace</a>.         </div>

### Path parameter

<b>chatId</b>	userId, roomId, groupId
---------------	-------------------------

### Response

Returns a `200` status code on success.

### Error response

When an error occurs, one of these status codes is returned:

Status code	Description
400	There is a problem with your request.
403	You don't have permission to access.
404	The target user isn't a friend.

## Default Active

The module channels offered on the LINE Marketplace are given the "Default Active" feature.

The features of the module channel with the "Default Active" feature are as follows.

**i This feature is exclusive to the LINE Marketplace**

The Default Active feature is only available for module channels published on the LINE Marketplace.

### Auto active

---

The normal module channel becomes a Standby Channel when attached to a LINE Official Account. After that, the module channel acquires initiative (Chat Control) using the Acquire Control API as needed (triggered by user operation, etc.) and becomes an Active Channel.

Module channels that have been given the "Default Active" feature will automatically become Active Channels when attached to a LINE Official Account. So, there is no need to call the Acquire Control API.

### Exclusive control

---

Only one module channel with the "Default Active" function can be attached to a LINE Official Account.

If a module channel with the "Default Active" functionality has already been attached to a LINE Official Account, you won't be able to attach any other module channels with the "Default Active" functionality to that account.

You can attach multiple module channels that don't have the "Default Active" function, but the current doesn't provide them.

## Using the Messaging API from the module channel

Module channels, like Messaging API channels, can use the Messaging API to send messages and switch between rich menus.

## Using the Messaging API with the module channel's channel access token

### User ID used in the module channel

In the module channel provided by the , the identifier for each user, which is the [user ID](#), is different from the conventional format and is an identifier consisting of a 68-digit character string starting with the letter "L".

This identifier will be different between LINE Official Accounts, even if they are the same user.

#### Example 68-digit identifier starting with L

```
LUb577ef3cbe786a8da85ff8e902a03fc6-U5fac33f633e72c192759f09afc41fa28
```

### Module channel's channel access token

Once the module channel has switched to Active Channel, you can call the Messaging API or Module Channel API using the module channel's channel access token.

You can use one of these channel access tokens for module channels. You can find the information you need to issue a channel access token in the [LINE Developers Console](#) > Module Channels > Channel Preferences.

- [Short-lived channel access token](#)
- [Channel access token with a user-specified expiration \(Channel access token v2.1\)](#)

#### Long-lived channel access tokens can't be used

Long-lived channel access tokens are not available for module channels.

#### **LINE Developers documentation**

- [Messaging API > Messaging API reference > Issue channel access token](#)

### Call the Messaging API endpoint

You can use the Messaging API by using the channel access token of the module channel.

However, pay attention to the scope and request headers.

#### scope

To use the Messaging API, you must have a scope defined for each endpoint.

The scope must be specified when attaching the module channel and permission to use it must be obtained from the LINE Official Account admin. For details, see [Request authorization from the LINE Official Account admin](#).

#### Request header

When calling the Messaging API endpoint from a module channel, specify the module channel's channel access token in the `Authorization` header. Also, be sure to specify "private header".

<b>Authorization</b>	<p>Bearer {channel access token}</p> <ul style="list-style-type: none"> <li>For {channel access token} , specify the <a href="#">module channel's channel access token</a>.</li> </ul>
<b>Private header</b>	<p>User ID of the LINE Official Account bot attached to the module channel</p> <ul style="list-style-type: none"> <li>You can get the user ID of the bot from the response of POST <a href="https://manager.line.biz/module/auth/v1/token">https://manager.line.biz/module/auth/v1/token</a> or the <a href="#">Attached event</a>.</li> <li><b>Specify "private header" for all API requests that require you to specify a LINE Official Account, not just for sending messages.</b></li> </ul> <div style="border: 1px solid #ffc107; padding: 10px; margin-top: 10px;"> <p><b>ⓘ Do not omit "private header"</b></p> <p>Module channel is a service designed to attach to multiple LINE Official Accounts. Be sure to specify "private header".</p> </div> <div style="border: 1px solid #dc3545; padding: 10px; margin-top: 10px;"> <p><b>⚠ The specific header name isn't disclosed</b></p> <p>The name (parameter name) of this header is open only to customers who participate in the <a href="#">LINE Marketplace</a>.</p> </div>

#### Example

```
curl -v -X POST https://api.line.me/v2/bot/message/push \
-H 'Content-Type:application/json' \
-H 'Authorization: Bearer {channel access token}' \
-H 'Private header: xxxxxxxxxxxxxxxxxxxxxxxxx' \      // NEED THIS HEADER
-d '{
  "to": "LUb577ef3cbe...",
  "messages": [
    {
      "type":"text",
      "text":"Hello, world1"
    }
  ]
}'
```

## Receiving a webhook

When you receive a webhook event on the Webhook URL server registered in the module channel, check the values of the `mode` and `destination` properties.

#### ⓘ Note

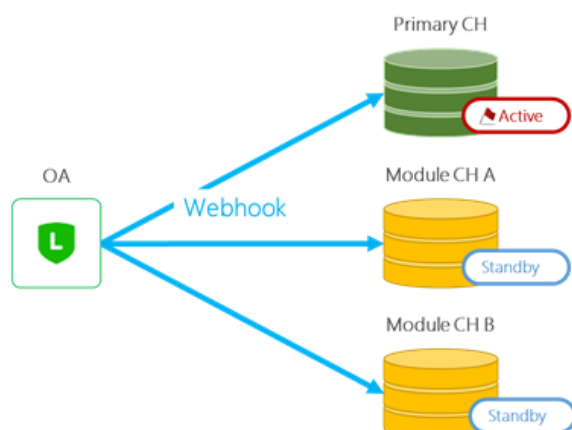
If the module channel webhook URL server doesn't receive the webhook event, check the following:

- The module channel must be attached to a LINE Official Account.  
Make sure you can send push messages from the module channel to users who have added the LINE Official Account as a friend.

- When requesting authorization from the LINE Official Account admin to attach the module channel, `message%3Areceive` (message:receive) must be specified in the `scope` query parameter of the authorization URL.  
For more information on scope, see [Request authorization from the LINE Official Account admin](#).

## mode property

Webhook events such as messages from users and adding friends will be sent to all channels connected to the LINE Official Account (Primary Channel and Module Channels attached to the LINE Official Account) at the same time.



Before processing the webhook event, make sure that each channel has the initiative (Chat Control) to respond to the end user.

To check the initiative (Chat Control), use the `mode` property of the Webhook event.

Value of <code>mode</code> property	Description
<code>active</code>	The channel that received the webhook event is the Active Channel.  The webhook URL server that receives this webhook event <b>can send reply messages, push messages, etc.</b>
<code>standby</code>	The channel that received the webhook event is the Standby Channel.  The webhook URL server that receives this webhook event <b>should refrain from sending messages.</b>

Of the channels attached to the LINE Official Account, only one channel has the `mode` property set to `active`. For all other channels, the `mode` property is set to `standby`.

### Example of a webhook event sent to Active Channel

```

{
  "replyToken": "0f3779fba3b349968c5d07db31eab56f", // NOTICE THIS PROPERTY
  "type": "message",
  "mode": "active", // NOTICE THIS PROPERTY
  "timestamp": 1462629479859,
  "source": {
    "type": "user",
    "userId": "LUb577ef3cbe..."
  }
},

```

```

"message": {
  "id": "325708",
  "type": "text",
  "text": "Hello, world"
}
}

```

### **Response messages aren't available on Standby Channel**

Webhook events arriving at the Standby Channel won't include the `replyToken` property.

Therefore, the response message isn't available.

#### **Example of a webhook event sent to the Standby Channel**

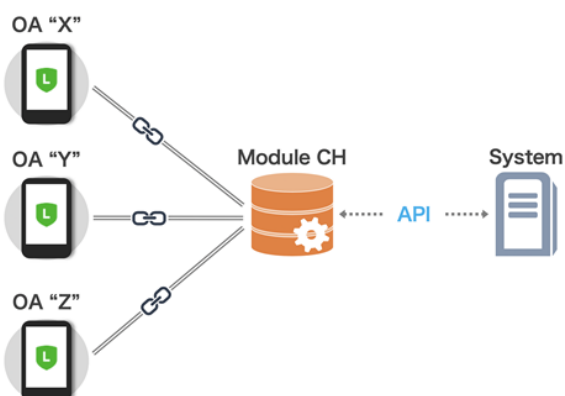
```

{
  // replyToken PROPERTY DOES NOT EXIST
  "type": "message",
  "mode": "standby", // NOTICE THIS PROPERTY
  "timestamp": 1462629479859,
  "source": {
    "type": "user",
    "userId": "U4af4980629..."
  },
  "message": {
    "id": "325708",
    "type": "text",
    "text": "Hello, world!"
  }
}

```

## **destination property**

Module channels may be attached to multiple LINE Official Accounts (OA "X", OA "Y", OA "Z", ...) as shown in the figure below.



Therefore, use the `destination` property to determine which LINE Official Account the webhook was sent from.



<b>destination</b>	User ID of the bot of the LINE Official Account that sent the webhook event
<b>String</b>	The value of the bot's user ID is a string that matches the regular expression <code>U[0-9a-f]{32}</code> .

**Webhook example**

```
{  
  "destination": "U53387d548170020e6cedef5f41d1e01d",  // CHECK THIS PROPERTY  
  "events": [...]  
}
```

## Receive module channel-specific Webhook events

These webhook events are sent to the module channel's webhook URL server.

Webhook event	Description
Attached event	This event indicates that the module channel has been linked to the LINE Official Account.
Detached event	This event indicates that the module channel has been unlinked from the LINE Official Account.
Activated event	This event indicates that the module channel has been switched to Active Channel by calling the Acquire Control API.
Deactivated event	This event indicates that the module channel has been switched to Standby Channel by calling Acquire Control API or Release Control API.
botSuspend event	This event indicates that the LINE Official Account has been suspended (Suspend).
botResumed event	This event indicates that the LINE Official Account has returned from the suspended state.

### ✓ How to detect a change in initiative (Chat Control)

When the module channel is set to Active Channel, the initiative (Chat Control) may change automatically without calling the Release Control API. You can detect a change in initiative (Chat Control) in the following ways:

Webhook event	Description
<ul style="list-style-type: none"> <li>Activated event</li> <li>Deactivated event</li> </ul>	If you call the Acquire Control API or Release Control API on a module channel that is attached to a LINE Official Account, the initiative (Chat Control) of the module channel will be switched.
<ul style="list-style-type: none"> <li><a href="#">Follow event</a></li> <li><a href="#">Unfollow event</a></li> </ul>	<p>When an end user blocks the LINE Official Account and adds it as a friend again, the initiative (Chat Control) will be automatically reset to the default state.</p> <ul style="list-style-type: none"> <li>If the module channel has the <b>Default Active</b> function, it will automatically become the Active Channel.</li> </ul>

### ✓ About the suspended state (Suspend) of the LINE Official Account

Regardless of the module channel settings or service availability, the LINE Official Account may be suspended (Suspend) at the convenience of the LINE Official Account operator. Specifically, the LINE Official Account will be suspended in these situations:

- When you delete your LINE Official Account
- When the use of your LINE Official Account is suspended for any reason.

In some cases, the LINE Official Account may be restored from its suspended status. A webhook event will be sent when your LINE Official Account is suspended or restored.

Implement the module so that there's no conflict in the information managed by the module channel side.

## Attached event

This event indicates that the module channel has been attached to the LINE Official Account. Sent to the webhook URL server of the module channel.

<b>timestamp, etc.</b>	See <a href="#">Common properties</a> . However, <b>mode</b> is fixed to <code>active</code> .
<b>type</b> <i>String</i>	<code>module</code>
<b>module.type</b> <i>String</i>	<code>attached</code>
<b>module.botId</b> <i>String</i>	User ID of the bot on the attached LINE Official Account
<b>module.scopes</b> <i>String array</i>	Scope permitted by the admin of the LINE Official Account

## Detached event

This event indicates that the module channel has been detached from the LINE Official Account. Sent to the webhook URL server of the module channel.

<b>timestamp, source, etc.</b>	See <a href="#">Common properties</a> . However, <b>mode</b> is fixed to <code>active</code> .
<b>type</b> <i>String</i>	<code>module</code>
<b>module.type</b> <i>String</i>	<code>detached</code>
<b>module.botId</b> <i>String</i>	Unlinked LINE Official Account bot user ID
<b>module.reason</b> <i>String</i>	Reason for unlinking <ul style="list-style-type: none"> <li><code>bot_deleted</code> : All information, including analysis data for the LINE Official Account, has been completely deleted.</li> </ul>

### Detach isn't done when you delete the LINE Official Account

The module channel won't be unlinked when the LINE Official Account Manager is used to delete the LINE Official Account.

After three months have passed since the operation to delete the account, and all information including the LINE Official Account's analysis data has been completely deleted, the account will automatically be detached.

## Activated event

This event indicates that the module channel has been switched to Active Channel by calling the Acquire Control API. Sent to the webhook URL server of the module channel.

<b>timestamp, source, etc.</b>	See <a href="#">Common properties</a> . However, <b>mode</b> is fixed to <code>active</code> .
<b>type</b> <b>String</b>	<code>activated</code>
<b>chatControl.expireAt</b> <b>Number</b>	The time limit for maintaining "active".

### Note

The Activated event won't be sent if the validity period specified in the Acquire Control API has expired and the initiative (Chat Control) has been switched.

## Deactivated event

This event indicates that the module channel has been switched to Standby Channel by calling Acquire Control API or Release Control API. Sent to the webhook URL server of the module channel.

<b>timestamp, source, etc.</b>	See <a href="#">Common properties</a> . However, <b>mode</b> is fixed to <code>active</code> .
<b>type</b> <b>String</b>	<code>deactivated</code>

### Note

The deactivated event won't be sent if the validity period specified in the Acquire Control API has expired and the initiative (Chat Control) has been switched.

## botSuspend event

This event indicates that the LINE Official Account has been suspended (Suspend). Sent to the webhook URL server of the module channel.

When you receive this event, it's recommended that you do the following:

- Display a message such as "This admin screen can't be used because the LINE Official Account is unavailable" on the module channel admin screen, and stop using the admin screen.
- Even if it goes into the paused state, it may return from the paused state (it may receive a botResume event). It is recommended that all information be kept.

<b>timestamp, etc.</b>	See <a href="#">Common properties</a> . However, <b>mode</b> is fixed to <code>active</code> .
------------------------	---

<b>type</b> String	botSuspended
-----------------------	--------------

```
{
  "events": [
    {
      "type": "botSuspended",
      "timestamp": 1616390574119,
      "mode": "active"
    }
  ],
  "destination": "U53387d548170020e6cedef5f41d1e01d"
}
```

#### **Note**

- The botSuspend event isn't sent to the Primary Channel.
- If you receive the Detached event after receiving the botSuspend event, it means that the LINE Official Account has stopped using the module channel and canceled the contract.

## botResumed event

This event indicates that the LINE Official Account has returned from the suspended state. Sent to the webhook URL server of the module channel.

When you receive this event, it's recommended that you do the following:

- Hide the message "This admin page is unavailable due to the LINE Official Account being unavailable" from the module channel admin page and resume using the admin page.

<b>timestamp, etc.</b>	See <a href="#">Common properties</a> . However, <b>mode</b> is fixed to <code>active</code> .
<b>type</b> String	botResumed

```
{
  "events": [
    {
      "type": "botResumed",
      "timestamp": 1616390634211,
      "mode": "active"
    }
  ],
  "destination": "U53387d548170020e6cedef5f41d1e01d"
}
```

**Note**

The botResumed event isn't sent to the Primary Channel.

## Get the LINE Official Account information from the module channel


### Get bot information

Gets basic information about the bots of LINE Official Accounts that have attached module channels.

See the LINE Developers documentation> Messaging API> Messaging API reference> "[Get bot info](#)" for details.

Also, specify this content in the request header:

#### Request header

<b>Authorization</b>	Bearer {Channel Access Token} <ul style="list-style-type: none"> <li>For {channel access token} , specify the module channel's channel access token.</li> </ul>
<b>Private header</b>	User ID of the LINE Official Account bot attached to the module channel <ul style="list-style-type: none"> <li>You can get the user ID of the bot from the POST <a href="https://manager.line.biz/module/auth/v1/token">https://manager.line.biz/module/auth/v1/token</a> response or the Attached event.</li> </ul> <div>  <b>Specific header names aren't disclosed</b>            The name (parameter name) of this header is open only to customers who participate in the <a href="#">LINE Marketplace</a>.         </div>

### Get information on multiple bots

Gets basic information about the bots of multiple LINE Official Accounts that have attached module channels.

#### HTTP request

```
GET https://api.line.me/v2/bot/list?limit={limit}&start={continuationToken}
```

#### Request header

<b>Authorization</b>	Bearer {Channel Access Token} <ul style="list-style-type: none"> <li>For {channel access token} , specify the module channel's channel access token.</li> </ul>
----------------------	---

#### Query parameters

<b>limit</b>	Specifies the maximum number of bots to get basic information.
Optional	If omitted, or if a value greater than 100 is specified, the basic information of up to 100 bots can be obtained.

<b>start</b> Optional	<p>The value of the continuation token. Included in the <code>next</code> property of the JSON object returned in the response.</p> <p>If you can't get all the basic information about the bot in one request, specify this parameter to get the rest of the array.</p>
--------------------------	--

## Response

Returns a JSON object with status code `200` and this information on success.

<b>bots</b>  <b>Array of Bot list Item objects</b>	Array representing basic information about the bot
<b>bots[].userId</b>  <b>String</b>	Bot's user ID
<b>bots[].basicId</b>  <b>String</b>	Bot's user ID
<b>bots[].premiumId</b>  <b>String</b>	Premium ID of the bot. This value isn't included if the premium ID hasn't been set.
<b>bots[].displayName</b>  <b>String</b>	Bot's display name
<b>bots[].pictureUrl</b>  <b>String</b>	Profile image URL. Image URL starting with "https://". If you haven't set a profile image for your bot, it won't be included in the response.
<b>next</b>  <b>String</b>	Continuation token. Used to get the next array of basic bot information. This property is only returned if there is basic information about the bot that the previous response <code>bots</code> couldn't get.

## Notes

- When Detaching a module channel, there will be a time lag before the settings are reflected. Don't send requests after detaching.
- If you want to add a Scope to a target account, you can do so even for accounts that have already been attached.