

Grandstream Networks, Inc.

Wave Administrator Guide



Wave Administrator Guide

Wave Desktop supports the functions of Wave administrator.

Wave administrator can configure Wave related functions as an enterprise administrator (UCM administrator). Currently, you can configure LiveChat's Chatbot, view and message records, and enable the third-party integration interface for sending messages.

PREREQUISITES

The UCM administrator should enable the Wave Administrator privileges for the specific extension by accessing the administrator web UI of the UCM, then navigate to **Extension/Trunk > Extensions**, then on "User Portal/Wave Privileges", select "Wave Administrator" as shown in the figure below.

Extensions > Edit Extension: 1000

< Basic Settings Media Features Voicemail Custom Time Wave Client Follow Me Advanced Se >

General

* Extension

1000

CallerID Number

* Call Privileges

Local

* SIP/IAX Password

AuthID

* Concurrent Registrations

3

Disable This Extension

☐

User Settings

* User Portal/Wave Privileges

Wave Administrator

Add / Edit Privileges

First Name

Arthur

Last Name

Morgan

Email Address

morgan.arthur@email.com

* User/Wave Password

Mobile Number

+1

Department

Enterprise Root Directory

Job Title

Cancel

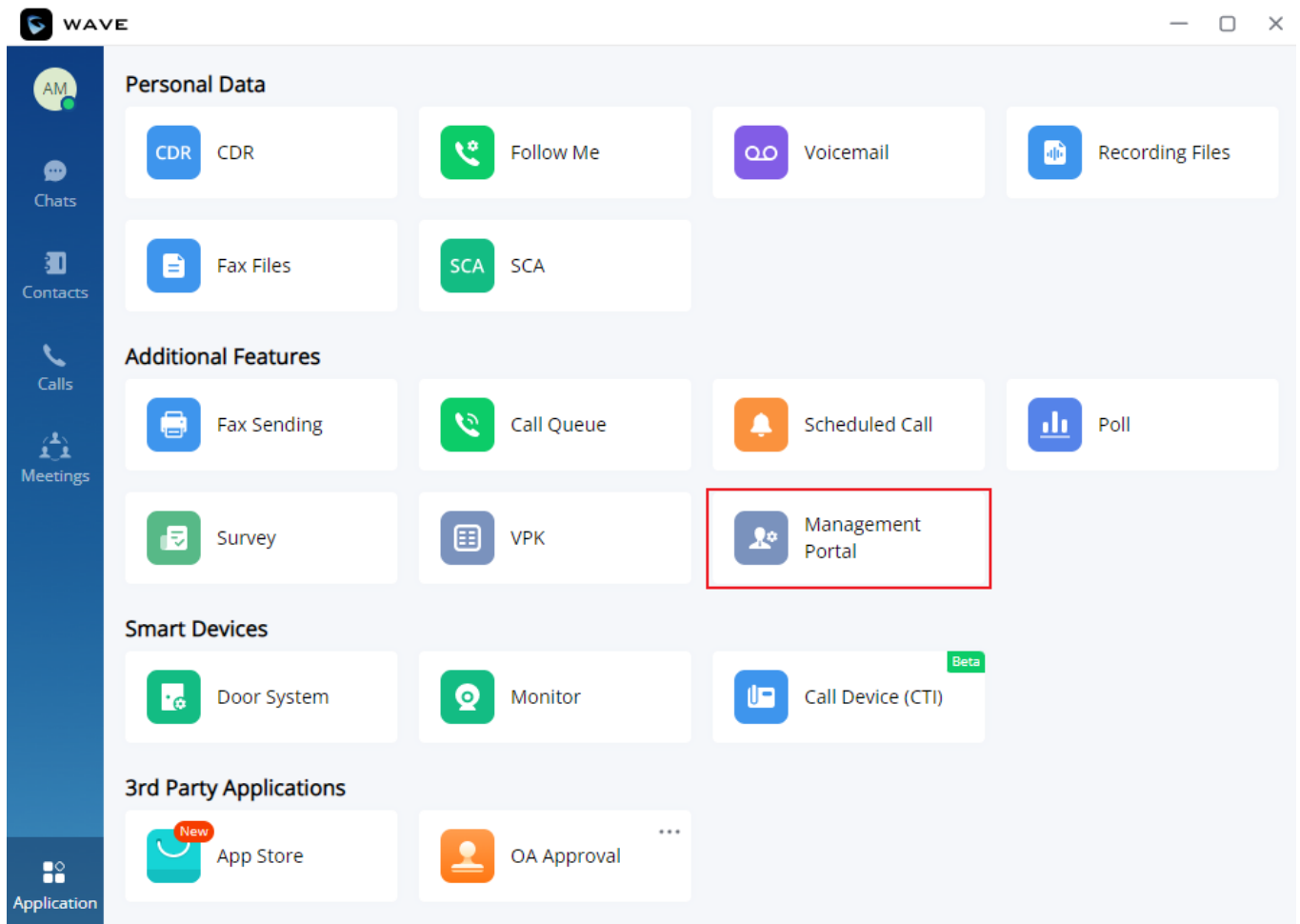
Save

Extension Basic Settings

In order to use this feature, please respect the software versions mentioned below:

- Software version for Wave Desktop must be 1.27.X or higher.
- Software version for the UCM6300 Series device must be 1.0.27.X or higher.

OPEN MANAGEMENT PORTAL



Wave Application

Chatbot

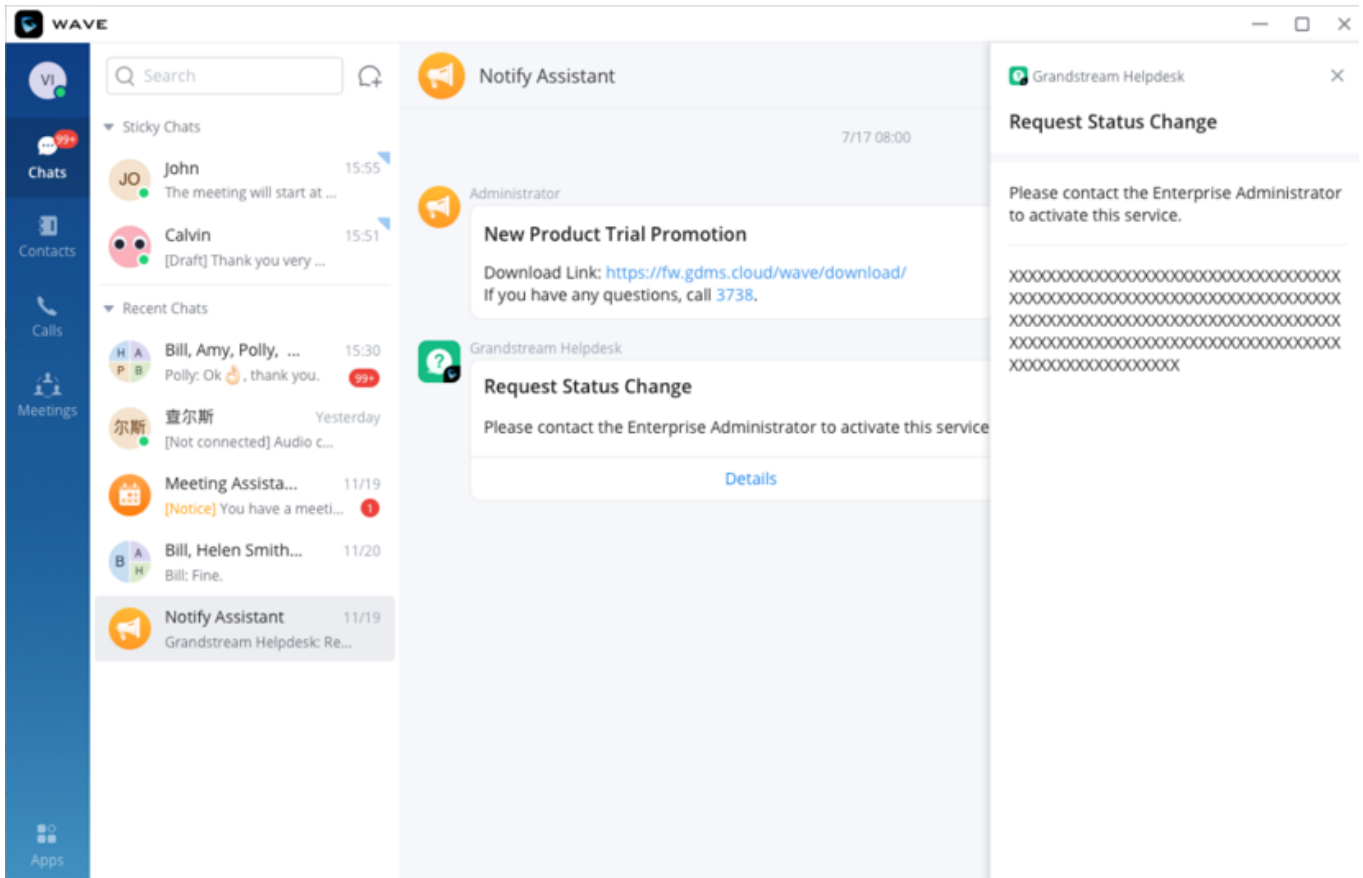
To configure the Chatbot feature please refer to the Chatbot Configuration Guide:
<https://documentation.grandstream.com/knowledge-base/live-chat-chatbot-admin-guide/>

Integration

Third-party systems can send notification messages to Wave users through the following interfaces.

For example, the third-party system "Helpdesk" sends notifications to Wave users when there is a new ticket or the ticket status changes, and then the "Helpdesk" system uses the interface to send notification messages to the specified extension or email address.

These extension users can receive notification messages when they log in to the Wave desktop or mobile client, as follows:

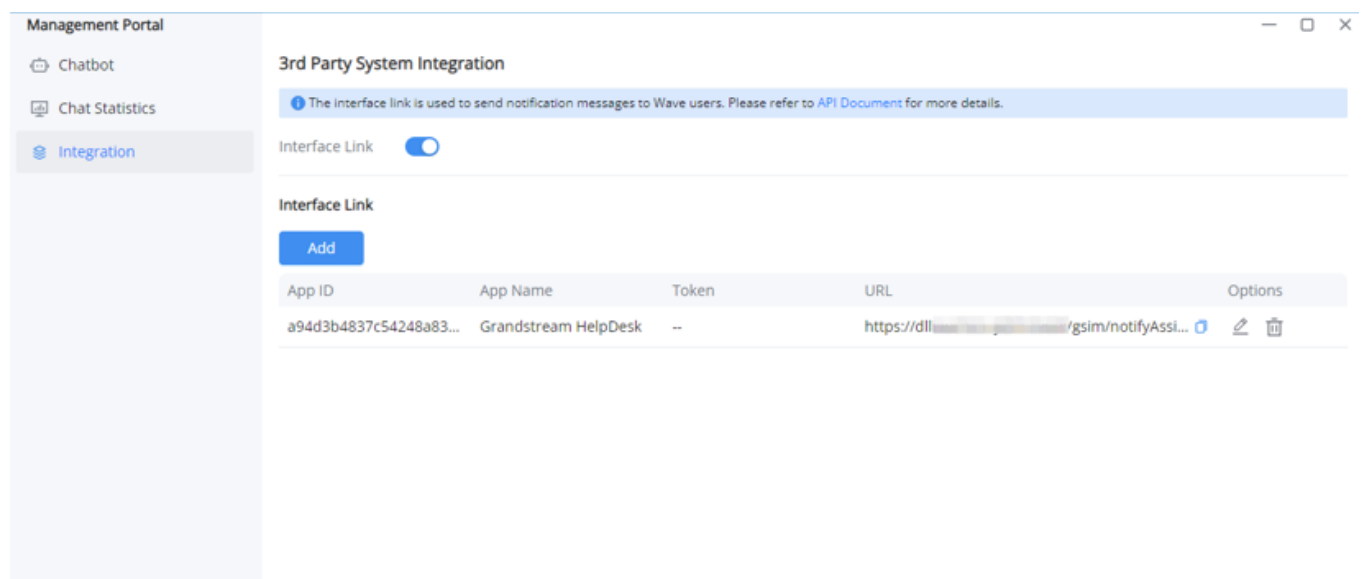


Notify Assistant

Enable Interface Link

To set an Interface Link please follow the steps below:

1. Log in to the Wave client, and the Wave administrator opens the management page in **Application > Management Portal > Integration**.
2. Click to enable interface link. After enabling, all the following interface links will be valid. Otherwise, all interface links will be invalid.



Interface Link Enabled

1. If your UCMRC/CloudUCM plan expires, all the links will also expire.
2. If the UCMRC/CloudUCM domain name changes, the interface link address will automatically change to the new domain name. Your third-party system needs to use the new link address.

Add Interface Link

Users can create different interface links for different third-party systems to facilitate the management of the use of third-party systems.

1. Log in to the Wave client, and the Wave administrator opens the management page in **Application > Management Portal > Integration > Wave Notification Messages**.
2. Click "Add" to add a new interface link.

Management Portal

Chatbot

Integration

Pre-installed Add-in

Wave Notification Messages SSO Authentication

The interface link is used to send notification messages to Wave users. Please refer to [API Document](#) for more details.

Interface Link ☒

Interface Link

Add

App ID	App Name	Token	URL	Options
No data				

Add Integration

APP ID	Enter your APP ID for unique identification. It can be a random string or your custom Wave add-in app ID. If you want to send a notification message to a Wave user, who can start the Wave add-in to parse the detailed data, you can enter your custom add-in APP ID in the Wave app store. Note: To learn how to upload a custom add-in to Wave's application store, please refer to the following link: https://doc.grandstream.dev/WAVE/EN/#api-Overview-Introduction
App Name	Enter a name, which will be displayed when the Wave client receives notification messages.
Token	Optional, you can enter a random string for authentication when using the interface. Parameters for calling the interface: secret="my_token";
Interface Link	An interface link corresponding to this APP ID will be generated for use by third-party systems. Parameters for calling the interface: curl -X POST -d "\$body" -k "https://c074ad2abcd.a.gdms.cloud/gsim/notifyAssistant/0-aae3e891849641069cfc37ea3d677d98-c074ad2a4b5c" -H "X-Hub-Signature-256: sha256=\$signature";

How to Call the Interface

Third-party systems can call the interface link through the following method to send notification messages to Wave users.

1. If your UCMRC/CloudUCM plan expires, all the links will also expire.
2. If the UCMRC/CloudUCM domain name changes, the interface link address will automatically change to the new domain name. Your third-party system needs to use the new link address.

Post:

```
/gsim/notifyingAssistant/:urlToken
```

Request example:

```
curl -i /gsim/notifyingAssistant/:urlToken
```

Header:

Field Name	Type	Description
X-Hub-Signature-256	string	Optional. This parameter is required only when "token" is set when adding a new interface link. Use Token to calculate the header field X-Hub-Signature-256 (please refer to the Github interface for usage).

Parameter:

The total length of this message body cannot exceed 5000 characters.

Field	Type	Required	Length	Description
to	array<string>	yes	1~500	The extension or email address that receives the notification message
title	string	yes	1~128	The title of the notification message, which will be displayed in the Wave notification assistant
description	string	yes	1~500	The message content will be directly displayed as a message in the Wave notification assistant
target	string	no	0~36	Optional. If you want the new notification message of the same object (such as the same ticket) to overwrite the old notification message, you can fill in the ID of this object (such as the ticket ID). For example, if you fill in the ticket ID here, when you send a new notification message with the same ticket ID again, the Wave client will overwrite the original notification message of the ticket ID with the new notification content.

messageType	string	yes	1~128	The message content format can be: "json", "text", "url", "sso_url", the default is "text".
data	string	no	0~4000	The details of the notification will not be directly displayed in the notification assistant. If you click the "Details" button, it will be parsed. If the messageType is "json" format, the Wave plug-in of this APP ID will be started to parse the data after clicking the details. If the messageType is "text" format, the text of the data will be displayed in a pop-up window after clicking the details. If the messageType is "url" format, this URL will be opened in a browser. If the messageType is "sso_url" format, this URL must include the App ID of the SSO Authentication API. Once clicked by the user, the re-assembled URL will be opened in a browser.

Request Parameter Example:

```
#!/bin/bash

body='{ "to": ["1001"], "title": "my title...", "description": "my description...", "data": "{}", "target": "20880603134400088888" }';
secret="my_token";
signature=$(echo -n "$body" | openssl -sha256 -hmac "$secret" | awk '{print $2}');
curl -X POST -d "$body" -k
"https://c074ad2abcd.a.gdms.cloud/gsim/notifyAssistant/0-aae3e891849641069cfc37ea3d677d98-c074ad2a4b5c" -H "X-Hub-Signature-256: sha256=$signature";
```

Body Example:

```
{
  "to": [
    "support@test.com",
    "1001"
  ],
  "title": "my title...",
  "description": "my description...",
  "messageType": "json",
  "target": "20880603134400088888",
  "data": "{ \"TicketID\": \"123456\", \"TicketNumber\": \"654321\", \"ProductMode\": \"xxx\", \"TicketTitle\": \"[title]\", \"TicketDescription\": \"[description]\", \"MacAddress\": \"[MAC]\", \"Bug\": \"[bug]\", \"TicketURL\": \"[...]\" }"
}
```

Response:

Field Name	Type	Description
im	object	
action	string	interface name

type	string	resp
tid	number	response id
errorCode	number	error code
errorInfo	string	error detail

Response Parameter Example:

```
{
  "im": {
    "action": "notifyAssistant",
    "type": "resp",
    "tid": 1709082631725,
    "errorCode": 0,
    "errorInfo": "success"
  }
}
```

Error Code List:

Error Code	Description
600	Webhook disabled
599	Data invalid length
598	Description invalid length
597	Title invalid length
548	Request body too long
508	Redis operation unknown error
506	Invalid key
503	Server abnormality
501	Server internal DB error
468	UCM device info is invalid
431	Stream read failed, failed to read webhook request body
426	User permission denied, the current user does not have permission
420	User not found
411	Error data type
402	Unknown action, it may be because the current UCM firmware version does not support this function
400	Illegal request parse json fail, the json format is incorrect and json parsing failed

Edit/Delete Interface Link

1. Log into Wave Desktop client and open the management page in **Application > Management Portal Integration**
2. Click the button to edit the interface link. Only the APP Name and Token can be edited, and the others cannot be edited.

After changing the Token, you also need to modify the Token synchronously when calling the interface.

< Edit

App ID a94d3b483... 72c0d9

*App Name

Token

Interface Link <https://...gsim/notifyAssistant/6-0d44942bd5e143128dd0aa61d6999edd-c074ad2a4b36>

3. Click on the button to delete the interface. After deletion, the link to this interface will be invalid.

App ID	App Name	Token	URL	Options
a94d3b...	Grandstream HelpDesk	--	https://dl...gsim/notifyAssi...	 

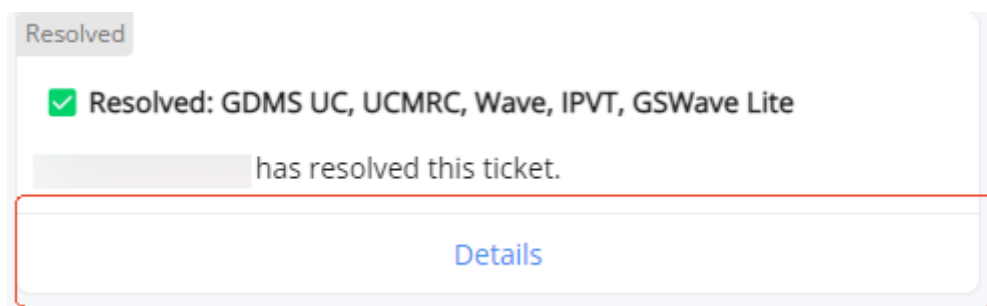
App Configuration Entry

WAVE SSO AUTHENTICATION

Third-party systems can integrate this SSO API (which recognizes Wave accounts) to enable users to sign in to third-party systems using their Wave account simply by clicking a link inside Wave notification messages ("Sign in with Wave" links).

Example:

When a user receives a notification message from a third-party system on Wave and clicks the "Details" button, the URL will open in a browser (e.g., an OA system). The URL carries the Wave account metadata, allowing the third-party system to decrypt the Wave account information using a secret key, achieving a seamless Single Sign-On (SSO) experience.



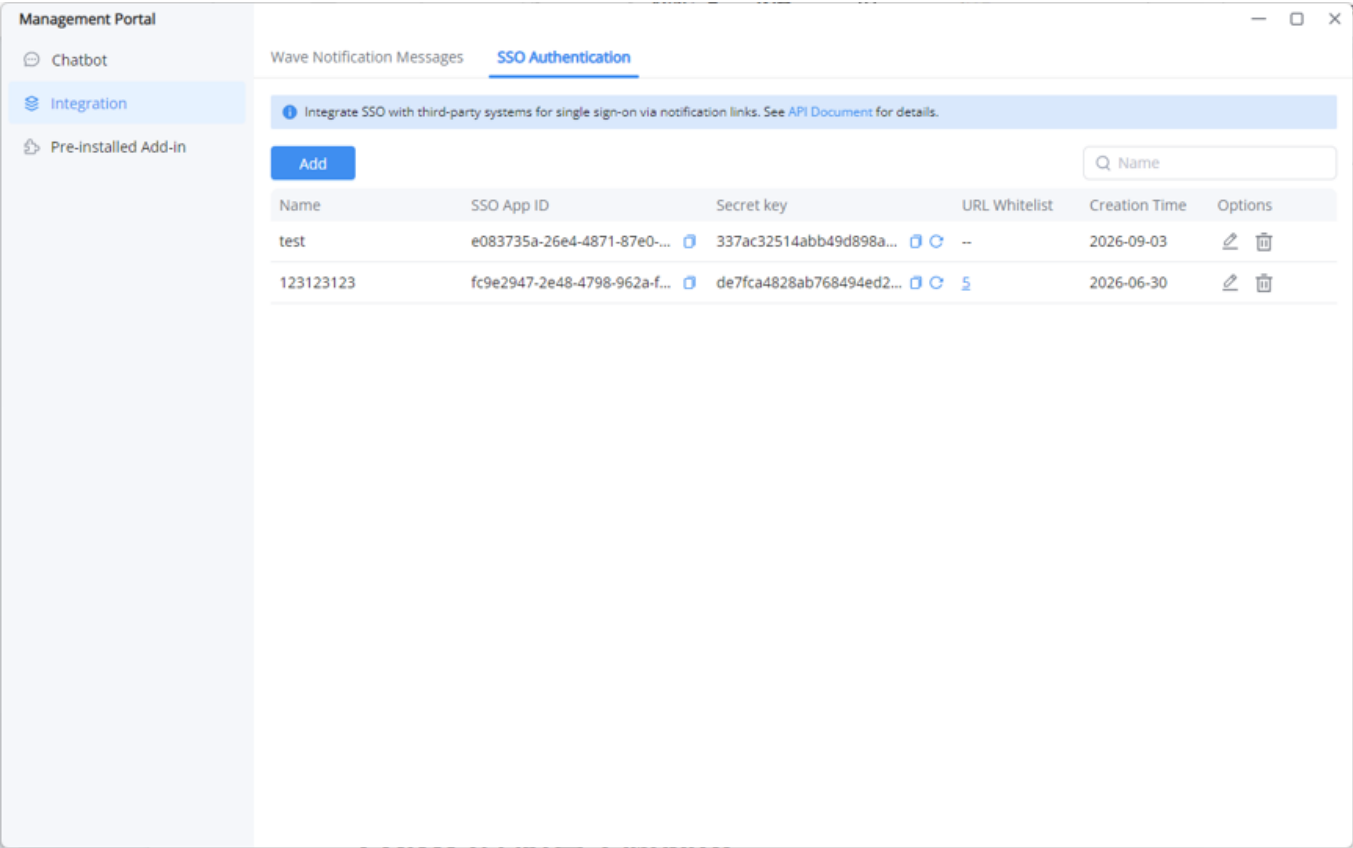
Wave Notification Message

- Go to Management Portal -> SSO Authentication and create a new SSO Authentication API. During creation, configure the input parameters as follows:

Parameter Name	Description
Name	Required. A descriptive/remark name for this SSO authentication, such as its intended purpose.
URL Whitelist	Optional. Specifies which URLs are eligible for this SSO authentication parsing. If left blank, any URL using this App ID will automatically be re-assembled to carry Wave account authentication metadata.

Once created successfully, the generated parameters are as follows:

Parameter Name	Description
App ID	Can be used as a URL parameter for the notification message SSO interface link (see steps below).
Secret Key	Used to authenticate the Wave account information within the newly re-assembled URL. The data is stored in JWT format (see steps below).



SSO Authentication

2. Inject the SSO interface link through the [Add Interface Link]. The link assembly rules are defined below, requiring the inclusion of the SSO appid:

```
https://xxx.yourcompany.cloud:9443/servlet/sso_login?appid=xxxxxxxxxx&redirect_url=https://gxxx.yourcompany.cloud:9443/v2/indax%3Fembed_url=sys/file/my/index"
```

3. When a user clicks the details button in the notification message, Wave will construct a new URL and launch it in a web browser:

```
https://xxx.yourcompany.cloud:9443/servlet/sso_login?redirect_url=https%3A%2F%2Fxxx.yourcompany.cloud%3A9443%2Fv2%2Findax%3Fembed_url%3Dsys%2Ffile%2Fmy%2Findex\u0026login_hint=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhcHBpZCI6ImdkbXNfZGV2IiwiaWF0IjoxNzc0OTQyMTk4LCJleHRlbnNpb24iOiIxMDA1IiwiaWZmlzc3RfYmFtZSI6ImRknzQ0N3lhMGQ4Zjh5IiwiaWF0IjoxNzc0OTQyMTk4LCJpc3MiOiJnc2ltIiwibGFzZdF9uYW1lIjoizGQzMGUzeSI6InVzZXJfawQiOiJkbGxpduBncmFuZHN0cmVhbS5jb2IiInVzZXJfdHlwZSI6ImVtYWlsIn0.J-nE18DDOdBNmFZ-xYu9HtIi-uvh0o3zNmIeTCN9PDY
```

4. Third-party systems shall verify and parse user metadata leveraging the login_hint parameter within the URL payload.

JWT follows the standard protocol, and signature verification can be performed using standard procedures.

The commands under Linux are as follows:

```
JWT="eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhcHBpZCI6ImdkbXNfZGV2IiwiaXhwIjoxNzc0OTQyMTk4LCJleHRlb  
nNpb24iOiIxMDA1IiwiaWZmlzc3RfbmFtZSI6ImRkNzQ0N3lhMGQ4Zjh5IiwiaWF0IjoxNzc0OTQyMTk4LCJpc3MiOiJnc2ltIiw  
bGFzdF9uYW11IjoizGQzMGUzeSIsInVzZXJfawQiOiJkbGxpdUBncmFuZHN0cmVhbS5jb2IiInVzZXJfdHlwZSI6ImVtYWlsIn0  
.J-nE18DD0dBNmFZ-xYu9HtIi-uvh0o3zNmIeTCN9PDY"
```

```
SECRET="x10KCayFXAMoEMWx0cek57alqyhRk0TD"
```

```
HEADER=$(echo $JWT | cut -d '.' -f1)  
PAYLOAD=$(echo $JWT | cut -d '.' -f2)  
SIGNATURE=$(echo $JWT | cut -d '.' -f3)
```

```
LOCAL_SIGN=$(echo -n "$HEADER.$PAYLOAD" | openssl dgst -sha256 -hmac "$SECRET" -binary | base64 |  
tr -d '=' | tr '/+' '_-' | tr -d '\n')  
echo "Original Signature:$SIGNATURE"echo "Calculated Signature:$LOCAL_SIGN"  
if [ "$SIGNATURE" = "$LOCAL_SIGN" ]; then  
echo -e "\n✅ Signature verification successful: JWT is valid and authentic"else  
echo -e "\n? Signature verification failed: Signatures do not match."  
fi
```

```
Parsed payload example:echo $PAYLOAD | base64 -d 2>/dev/null | jq .  
{  
  "appid": "gdms_dev",  
  "exp": 1774942198,  
  "extension": "1005",  
  "first_name": "dd7447ya0d8f8y",  
  "iat": 1774942138,  
  "iss": "gsim",  
  "last_name": "dd30e3y",  
  "user_id": "user123@company123.com",  
  "user_type": "email"  
}
```

REFERENCES

[Live Chat Chatbot Admin Guide](#)

[Wave Desktop - User Guide](#)

[Live Chat - User Guide](#)