



Operation Support System (OSS)

First contact

1. Sigfox Cloud GUI
2. Organization & Hierarchy
3. Devices & Device type
4. Service prediction
5. APIs & Callbacks

1

Cloud GUI

Pre-requisites

User checklist:

- ✓ Internet access
- ✓ Computer with correct date/time
- ✓ Recent web browser (Chrome, Firefox & Safari preferred)
- ✓ Account creation email received

Production cloud:
backend.sigfox.com

Support contact:
support.sigfox.com

Cloud GUI



DEVICE DEVICE TYPE USER GROUP



NEWS

SERVICE MAPS

KNOWN ISSUES

SIGFOX_France

Welcome to
sigfox portal

OPERATOR CLIENT

Release 8.9

SIGFOX Cloud 8.9 release implements internal system improvement to enhance user experience and satisfaction. This release also contains the following:

COLLECTIVE IMPROVEMENT

[APIv2 General availability](#)

Disclaimers about APIv1 deprecation and APIv2 general availability have been changed on the different pages. New Terms and Conditions for APIv2 are displayed at first login on backend after the release is done.

BASE STATIONS AND SITES

09
SEPTEMBER
2019



Cloud GUI

sigfox

DEVICE DEVICE TYPE USER GROUP

NEWS
SERVICE MAPS
KNOWN ISSUES

SIGFOX_France

Welcome to
sigfox portal

OPERATOR CLIENT

Release 8.9

SIGFOX Cloud 8.9 release implements internal system improvement to enhance user experience and satisfaction. This release also contains the following:

COLLECTIVE IMPROVEMENT

[APIv2 General availability](#)

Disclaimers about APIv1 deprecation and APIv2 general availability have been changed on the different pages. New Terms and Conditions for APIv2 are displayed at first login on backend after the release is done.

BASE STATIONS AND SITES

09
SEPTEMBER
2019

- Main page shortcut (News, service maps, known issues)
- Category selection (Device, Device type, User, Group)
- Global entries (profile, online help, logout)

Cloud GUI

The screenshot displays the Sigfox Cloud GUI interface for managing devices. The top navigation bar includes tabs for 'DEVICE', 'DEVICE TYPE', 'USER', and 'GROUP'. The 'DEVICE' tab is selected. On the left, a sidebar shows 'DEVICES' and 'DELETED DEVICES'. The main content area is titled 'Device - List' and features a search bar with filters for 'id', 'State', 'Last seen from date', and 'Last seen to date'. Below the search bar, there are fields for 'Group' and 'Include sub groups'. A toolbar at the bottom right contains icons for 'New', 'New series', 'Edit series', 'Transfer series', 'Replace series', 'Delete series', 'RESET', and 'FILTER'. A 'Count' indicator shows 'Count: 1000/1000' and a 'page 1' indicator is visible at the bottom center.

- Selected category
- Action buttons
- Filter conditions
- Displayed items/total

- Filter operations
- Export list to CSV
- Page switch
- Column display customization

Cloud GUI

The screenshot displays the Sigfox Cloud GUI interface. At the top, there are navigation tabs for 'DEVICE', 'DEVICE TYPE', 'USER', and 'GROUP'. Below these, a 'Device - List' section contains search filters for 'Id', 'State', 'Last seen from date', and 'Last seen to date'. A yellow highlight is placed over the 'Group' filter area, which includes a 'Select a group' button and an 'Include sub groups' checkbox. To the right, a 'Field selection' sidebar lists various columns available for display, such as 'Activation date', 'Automatic token renewal', 'Com status', 'Device type', 'Group', 'Id', 'Last purge', 'Last seen', 'Modem certificate', 'Name', 'PAC', 'Product certificate', 'Protocol', and 'Token state'. A green 'Apply' button is at the bottom of this sidebar. The main content area shows a table of devices with columns for 'Operational status', 'Device type', 'Group', 'Id', 'Last seen', 'Name', and 'Token state'. The table contains several rows of device data.

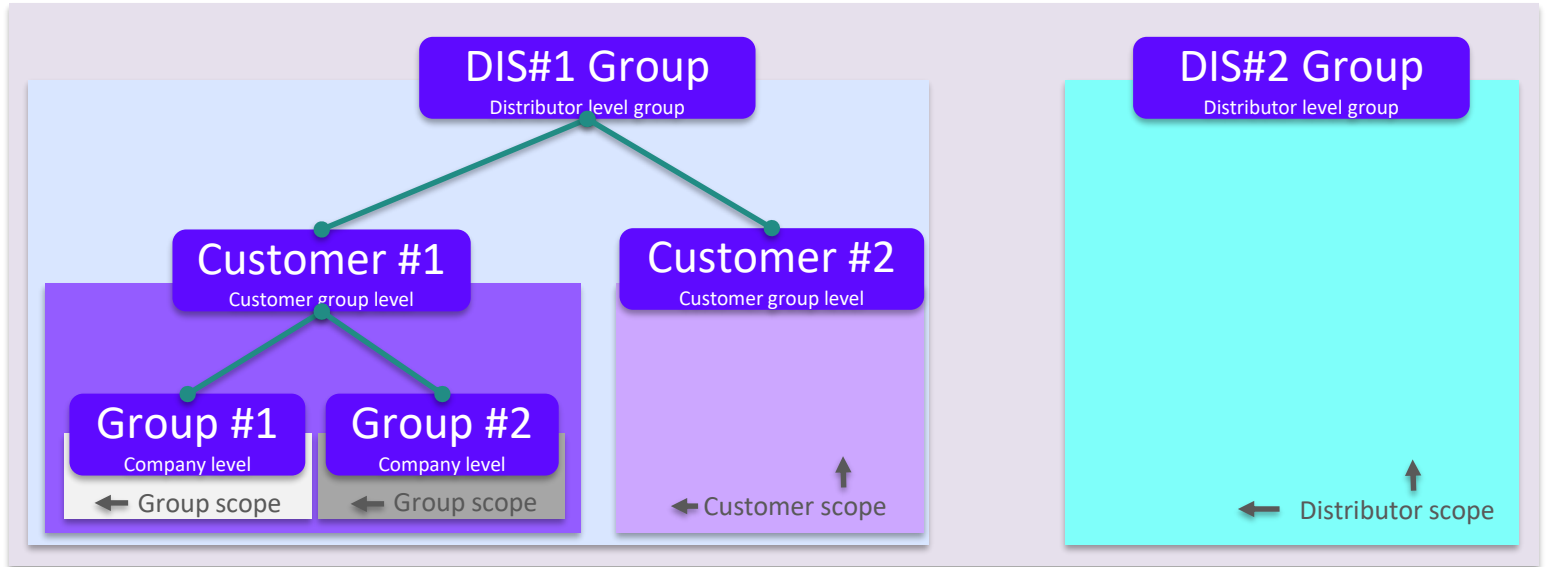
- Additional Filter conditions
- Column edition menu (max. 8 columns)
- Column edition validation

2

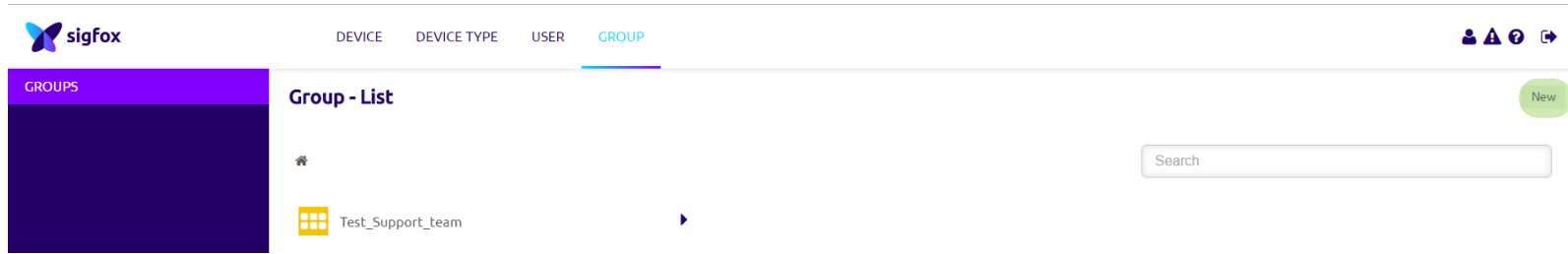
Organization & hierarchy

Group & Subgroups

Cloud organization is hierarchically structured

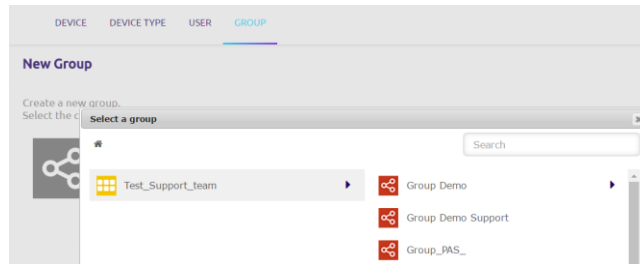


Group creation in detail



Step 1 : Click on New button in Group tab

Step 2 : Select the Parent Group



Group creation in detail

Step 3 : Enter Group information

Group - New

Group information

Name

Description

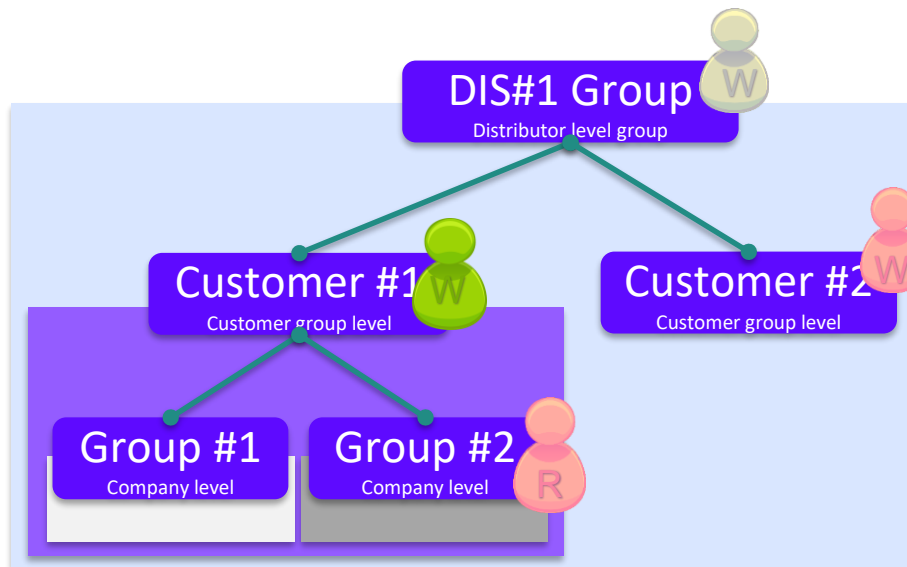
Parent group Group Demo

Timezone

User roles

User creation is linked to **rights allocation on groups**. A specific attention shall be given to the user role allocation:

- SIGFOX CORP predefines user roles
- DIS/Customers are granted with the rights that correspond to their needs
- User rights can be fine tuned (R/W)



User creation in detail

The screenshot shows the Sigfox user management interface. At the top, there are navigation tabs for 'DEVICE', 'DEVICE TYPE', 'USER', and 'GROUP', with 'USER' selected. A 'New' button is visible in the top right corner. Below the tabs, there is a search and filter section with input fields for 'Name/Email', 'Group', and a 'Profile' dropdown menu. To the right of these fields are several action buttons: a folder icon, a document icon, a trash icon, a document with a plus icon, 'RESET', and 'FILTER'. Below the search section, there is a 'page 1' indicator and a table header with columns for 'Name', 'Email', 'Last login', and 'Groups'.

Step 1: Click on New button in User tab

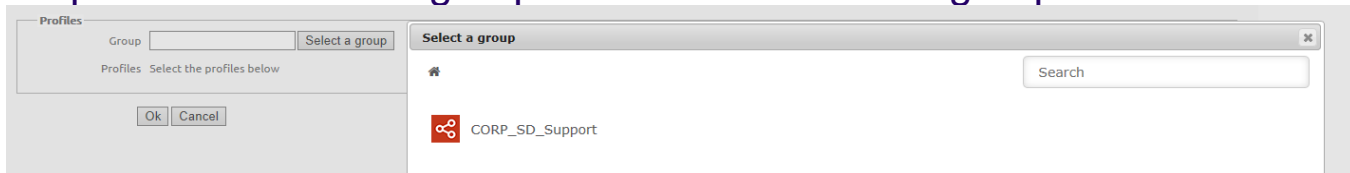
Step 2 : Enter user information

The 'New user' form is displayed, showing the 'User information' section. It contains the following fields:

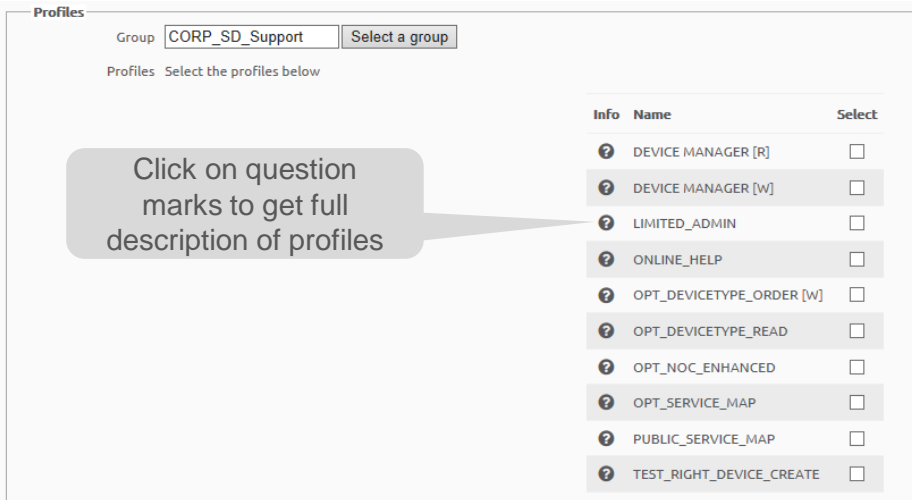
- First name
- Last Name
- Email
- Position
- Timezone

User creation in detail

Step 3 : Click select a group button and choose a group



Step 4 : Choose profiles



3

Devices & Device types

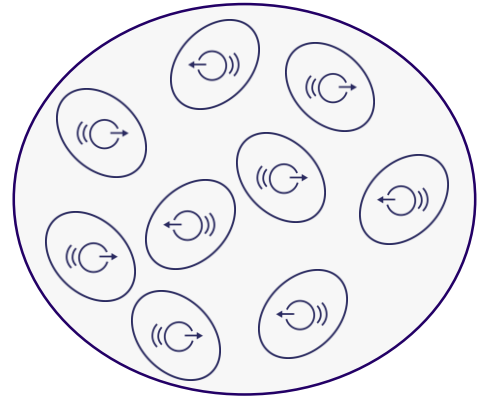
Device & Device type

Device notions:

- Unique ID per device
- One property title for each one: PAC (Porting Authorization Code)

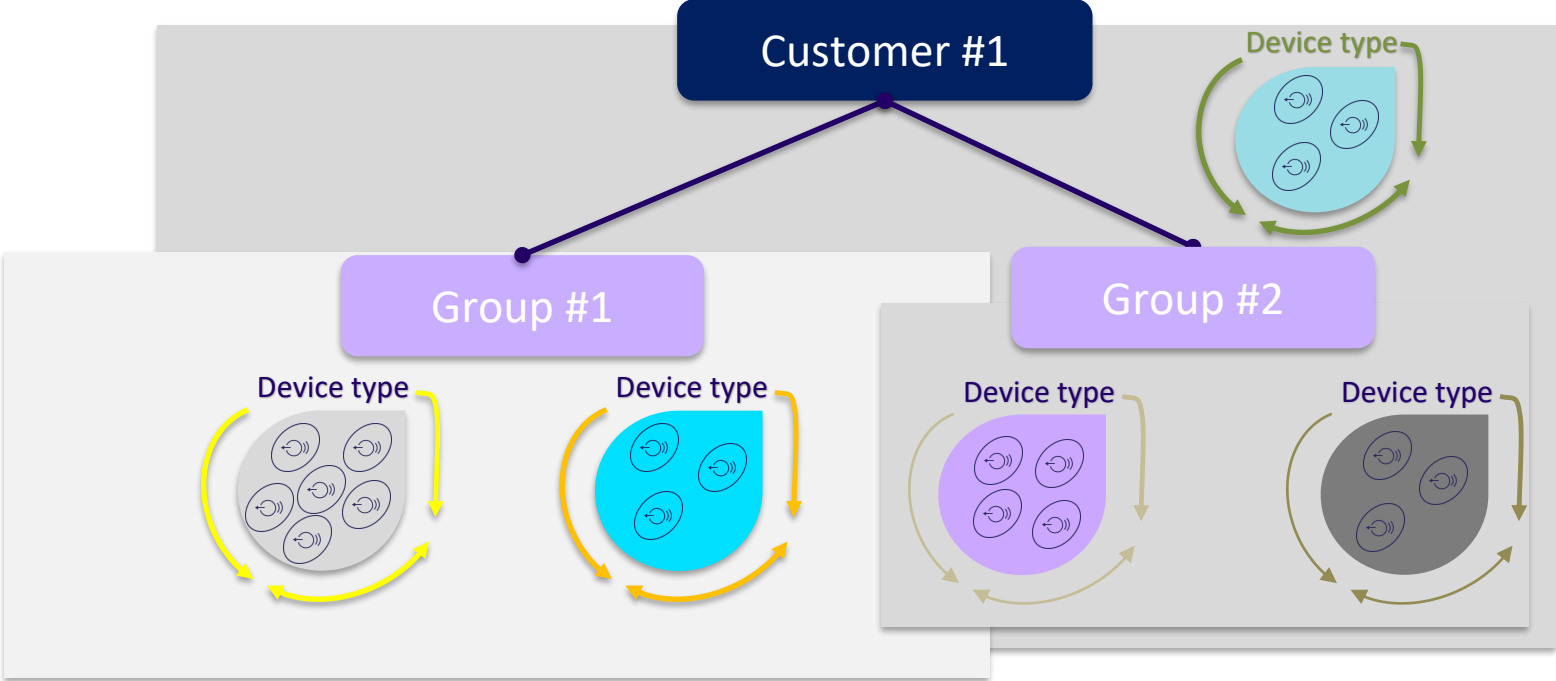
Device type notion:

- Set of devices with the same behavior
- Linked to a single order (same subscription levels and duration)
- Belongs to a unique group
- Callback availability to retrieve messages



⇒ To ease device management

Device & Device type

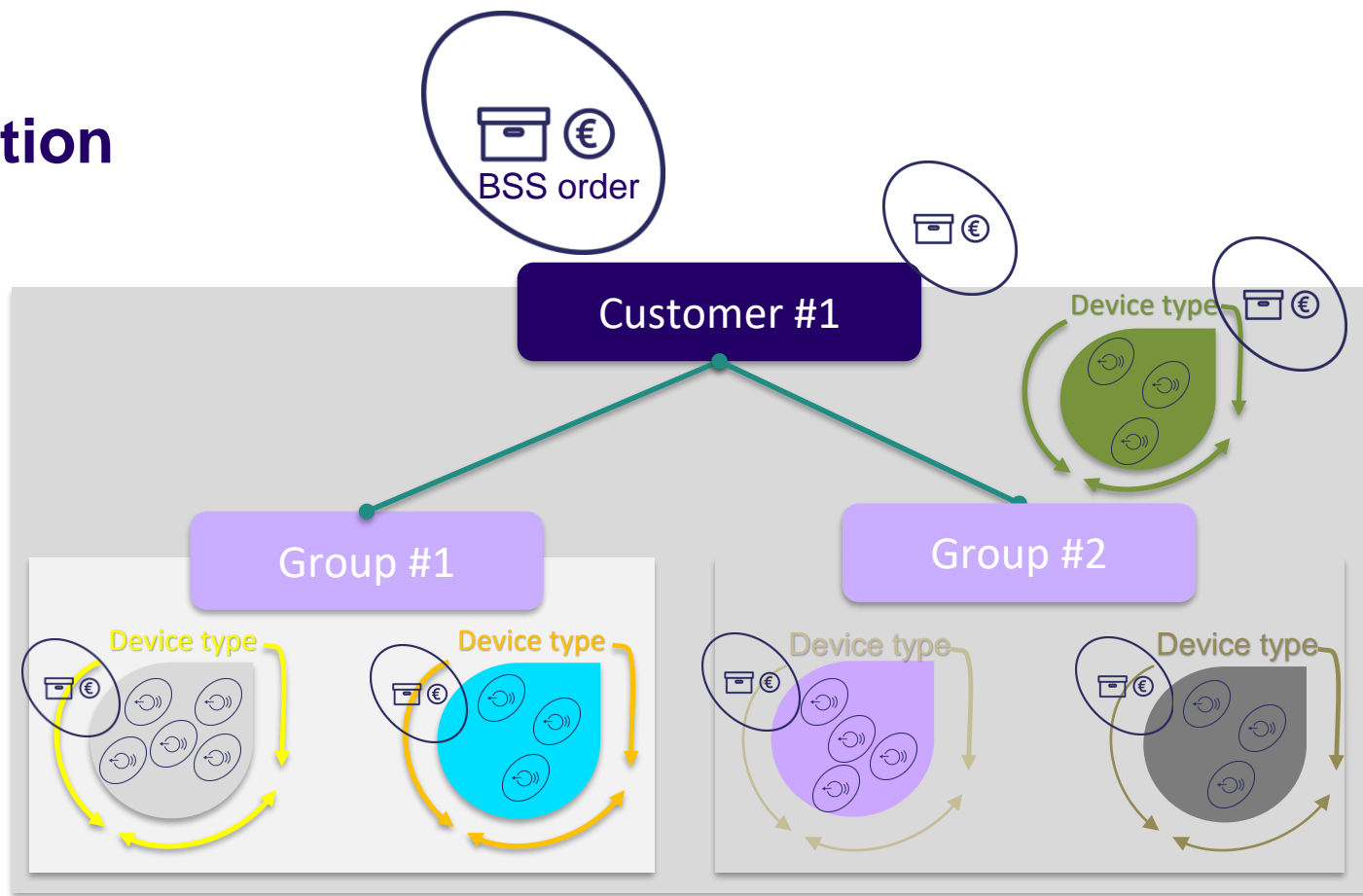


Order allocation

Customers
validate order:

Customers allocate
contracts to Groups
and possibly to
device type.

Groups allocates
contracts to device
type in their group or
in sub-groups below



Device type creation in detail

sigfox

DEVICE DEVICE TYPE USER GROUP

LIST

DEVICES BEING REGISTERED

GEOLOCATION PAYLOAD

BULK OPERATIONS

Device type - List

New

Name

Group Select a group

Include sub groups

Display type All

RESET FILTER

Count: 1 / 1

page 1

Description Display type Group Keep alive Name

Step 1 : Click on New button in Device type tab

Step 2 : Select a group

Device type - List

Select a group

Search

CORP_SD_Support

Count: 4 / 4

RESET FILTER

Device type creation in detail

Step 3 : Enter device type information

Device type - New

Device type information

- Device type name:** Name field
- Enable/disable Subscription automatic renewal for all devices:** Subscription automatic renewal checkbox (checked)
- Keep-alive configuration:** Keep-alive (in minutes) field (0)
- Select a contract:** Contract dropdown menu (Contract_ServiceDesk (9 tokens left, geoloc: yes))
- Email address configured for callback failure:** Alert email field

Downlink data

- Direct => Downlink data sent by the backend**
Callback => Downlink data sent through callback
- Downlink data sent in DIRECT mode:** Downlink mode dropdown menu (DIRECT)
- Display customization (Data decoding):** Downlink data in hexa field ({tapld}0000{rss})
- Payload display:** Payload parsing dropdown menu (Regular (raw payload))



Contracts can not be changed once device type is created

Device management in detail

Device - List

New New series Edit series Transfer series Replace series Delete series

Id

State

Average SNR (all)

Last seen from date

Count : 1 / 1

page 1

RESET FILTER

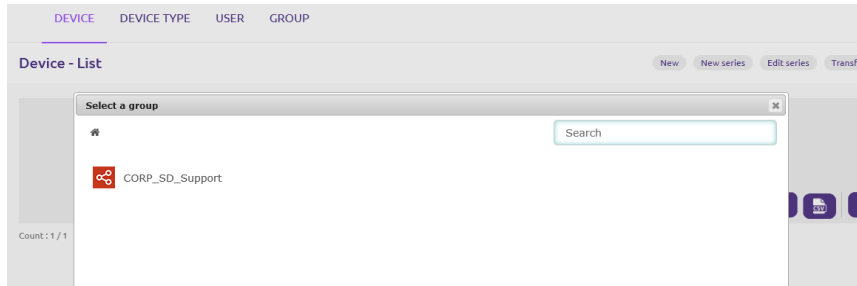
Average Rssi Average SNR Communication status Device type Group Id Last seen Name Token state

Step 1 : Select a way to register devices

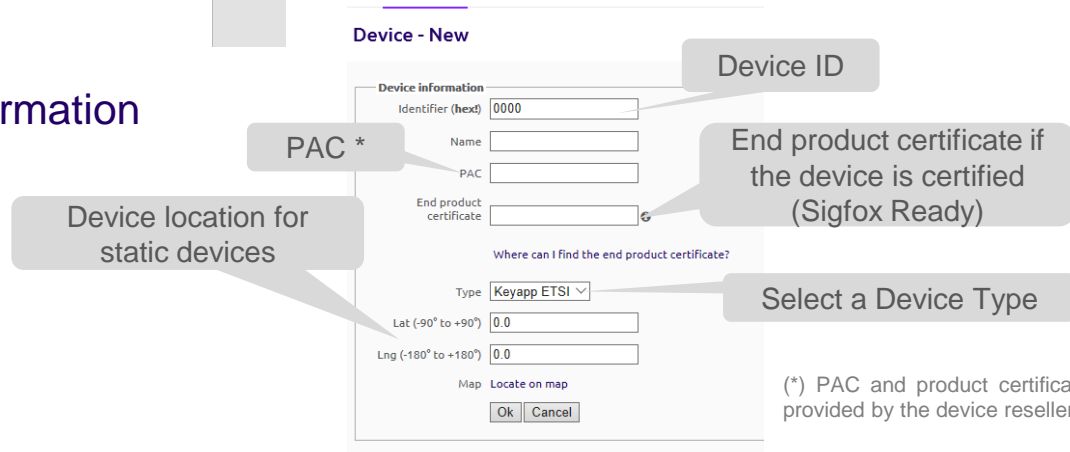
- New : register devices one by one
- New series : register batch of devices
- Edit series : edit device information
- Transfer series : move devices from device types (same or different contract)
- Replace series : replace a broken device by a new
- Delete series : batch of devices deletion

Device management in detail

Step 2a : If New has been chosen, select a group to register the device



Step 3a : Enter device information



Device - New

Device information

Identifier (hexd) **Device ID**

Name

PAC **PAC ***

End product certificate **End product certificate if the device is certified (Sigfox Ready)**

Where can I find the end product certificate?

Type **Select a Device Type**

Lat (-90° to +90°)

Lng (-180° to +180°)

Map [Locate on map](#)

Device location for static devices

(*) PAC and product certificate are provided by the device reseller

Device management in detail

Step 2b : If New series has been chosen

Device - Bulk creation

Use this feature to create several devices simultaneously

Device information

Batch name

Batch description

Devices names prefix

Group: **test_onboarding**

Type:

Identifiers Aucun fichier sélectionné.

End product certificate

Where can I find the end product certificate?

Subscription automatic renewal

Activable

Device name = prefix + increment

Device type

csv or txt file with ID/PAC

End product certificate if the devices are certified (Sigfox Ready)

Allows device to keep communicate after the end of its subscription date

Allows device to take a token after the end of its subscription date



Step 2c : If Edit series has been chosen

Device - Bulk edition

Use this feature to edit several devices simultaneously

Device information

Devices

csv or txt file with devices information

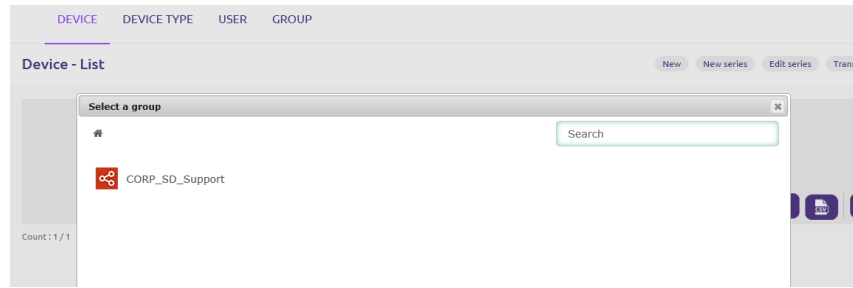
Helpful resources:



- [Register a batch of devices in one go](#)
- [Edit multiple devices simultaneously](#)

Device management in detail

Step 2d : If Transfer series has been chosen, select the destination group



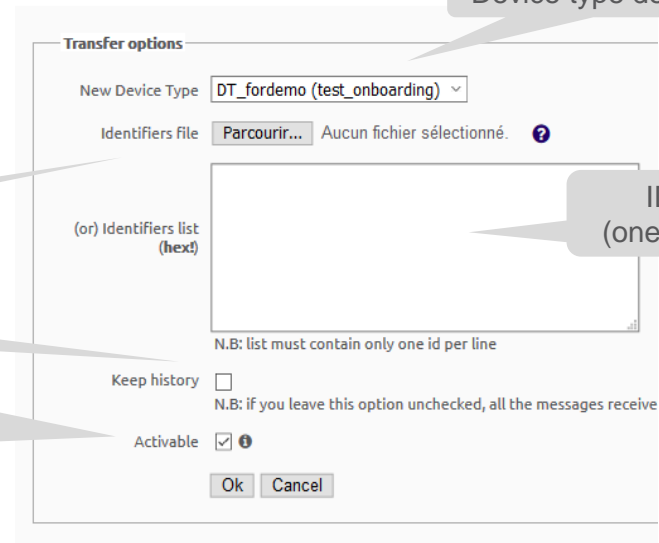
Step 3d : Enter transfer information

csv or txt file with ID

Keep message history

Allows device to take a token after the end of its subscription date

Device - Bulk Transfer



Transfer options

New Device Type:

Identifiers file: Aucun fichier sélectionné. ?

(or) Identifiers list (hex):

N.B: list must contain only one id per line

Keep history:

N.B: if you leave this option unchecked, all the messages receive

Activable: !

Device type destination

ID list
(one by line)


Device management in detail

Step 2e : If Replace series has been chosen

Device - Bulk replacement

Use this feature to transfer information from several devices to others simultaneously

Device information

Devices 

csv or txt file with old devices (faulty or stolen for instance) and new devices

Device management in detail

Step 2f : If Delete series has been chosen

Device - Bulk Delete

Delete options

Identifiers file Aucun fichier choisi ⓘ

Format of the file to choose

The format of the file has to be `.txt` or `.csv`
per line : one device identifier (hex)

Example file :

```
0017B46C
0017B46D
0017B46E
0017B46F
0017B470
0017B471
0017B472
0017B473
```

(or) Identifiers list (hex)

N.B: list must contain only one id per line

csv or txt file with identifier only (hex) to be deleted

OR enter the list of devices you want to delete in this field

- [Communication status indicator](#)
- [Link Quality: general knowledge](#)

Check device messages


1. Go to Device tab



DEVICE DEVICE TYPE USER GROUP

Device - List

2. Select a device by clicking on the ID

Communication status	Device type	Id	Last seen	Name	Token state
	Test5D	77FFF	2019-02-20 12:54:33	00077FFF	<input checked="" type="checkbox"/>

3. Go to the message tab



- INFORMATION
- LOCATION
- MESSAGES
- EVENTS
- STATISTICS
- EVENT CONFIGURATION

4. Send a message and check that the message has been received by the backend



2019-02-15 12:07:00

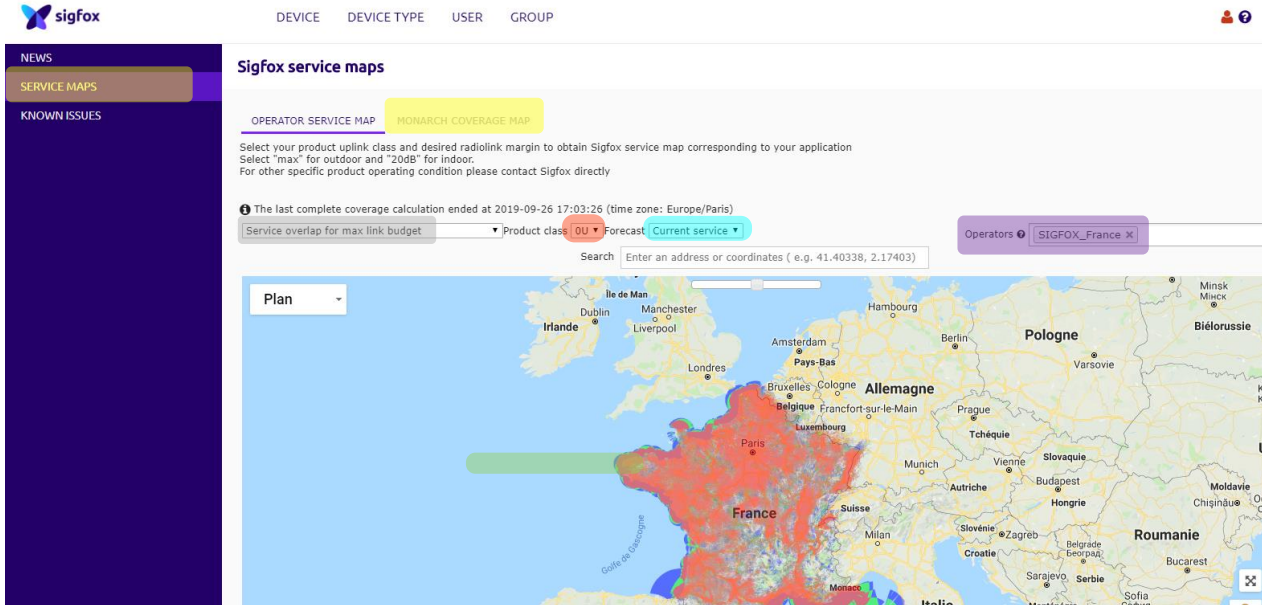
00000000000000000000150c62



4

Service map

Service map



● Service map

● Territory selection

● Installation type

● Monarch service map

● SIGFOX Ready Device class

● Forecast

● Specific place selection

5

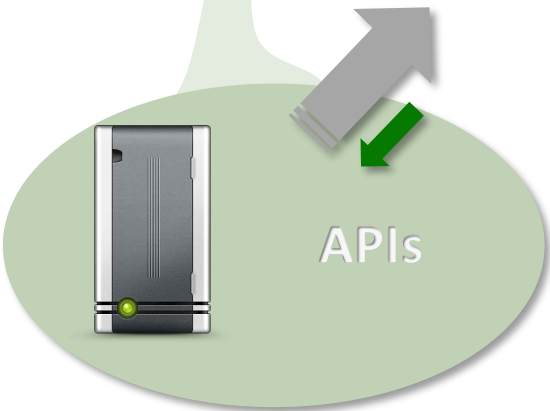
APIs & callbacks

Cloud interfaces

Web portal access to all cloud functions (admin, billing, service map, messages...)

API (pull mechanism) access to all cloud functions interconnecting customer servers with SIGFOX.
Usage: customer IT platform integration

Callbacks (push mechanism) for automatic, instantaneous notifications (new messages, alarms...)



API creation

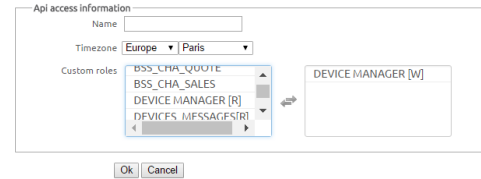
Step 1 : Click on New in API access tab (in Group tab)



The screenshot shows the Sigfox web interface. The top navigation bar includes 'DEVICE', 'DEVICE TYPE', 'USER', and 'GROUP'. The left sidebar has a menu with 'API ACCESS' highlighted. The main content area shows 'Api access 'Test_Support_team' - List' with a 'New' button in the top right corner.

Step 2 : Grant API the appropriate rights

Api access - Creation

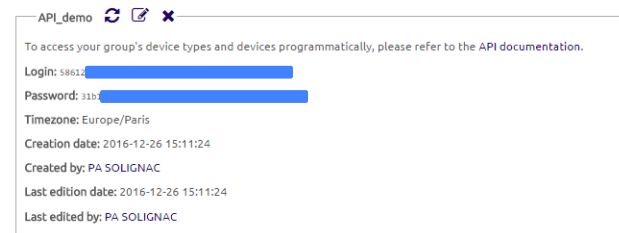


The 'Api access - Creation' dialog box contains the following fields and actions:

- Name:** An empty text input field.
- Timezone:** A dropdown menu currently showing 'Europe | Paris'.
- Custom roles:** A list of roles including 'BSS_CHA_QUOTE', 'BSS_CHA_SALES', 'DEVICE MANAGER [R]', and 'DEVICES_MESSAGING'. The 'DEVICE MANAGER [R]' role is selected and is being moved to the 'DEVICE MANAGER [W]' box.
- DEVICE MANAGER [W]:** An empty box for the selected role.
- Buttons:** 'Ok' and 'Cancel' buttons at the bottom.

Step 3 : Retrieve credentials to use API

Api access 'Test_Support_team' - List






The 'API_demo' window displays the following information:

- To access your group's device types and devices programmatically, please refer to the API documentation.
- Login: 5861: [redacted]
- Password: 315: [redacted]
- Timezone: Europe/Paris
- Creation date: 2016-12-26 15:11:24
- Created by: PA SOLIGNAC
- Last edition date: 2016-12-26 15:11:24
- Last edited by: PA SOLIGNAC


API creation

Step 4 : API documentation is generated according to API access rights





Api access 'Test_Support_team' - List

API_demo   

To access your group's device types and devices programmatically, please refer to the [API documentation](#).



DEVICE DEVICE TYPE USER GROUP

- General protocol information
 - Authentication and security
 - HTTP status codes
 - Using the API from the command line with curl
- Groups API
 - Group list
 - Group information
 - Group creation
 - Group edition
 - Group deletion
- Device types API
 - Device types list
 - Device type information
 - Device type creation
 - Device type edition
 - Device type deletion
 - Callback creation

REST API

[Export](#)

REST API

Some of the features found on the Sigfox backend's website can also be accessed programmatically using a webservice API. This API uses the HTTP protocol, following the REST principles.

All API endpoints return data in the JSON format, with the corresponding «application/json» content type header.
If a property has no value, it won't appear in the result.

General protocol information

Authentication and security

The API is accessible only using HTTPS, and all API endpoints require authentication credentials (login and password). Since all communication is encrypted, the authentication is performed using the simple HTTP Basic scheme.

Authentication credentials are associated to a **group**, which define the available device types that will be available with those credentials. A group initially does not have associated credentials, and it's up to you to create them using the «Generate credentials» buttons in the group edition screen.

If the API credentials ever get compromised, new ones can be regenerated at any moment, invalidating the previous ones.

CORS and JSONP are intentionally unsupported
Your credentials must remain private: exposing them to third parties is a bad practice.
CORS and JSONP JavaScript techniques tends to expose your credentials to your users.
If you really need to call Sigfox API from JavaScript in the browser, you must set a reverse proxy on your website.
Be careful not to blindly proxyify all request to Sigfox backend but select only needed ones.

HTTP status codes

Callback creation

Step 1 : Click on New in Callback tab (for a given device type)

The screenshot displays the Sigfox web interface. On the left, a dark purple sidebar lists navigation options: INFORMATION, LOCATION, ASSOCIATED DEVICES, DEVICES BEING REGISTERED, STATISTICS, EVENT CONFIGURATION, CALLBACKS (highlighted in a lighter purple), and BULK OPERATIONS. The top navigation bar features tabs for DEVICE, DEVICE TYPE (selected with a purple underline), USER, and GROUP. In the top right corner, there are icons for user profile, alert, help, and share. The main content area is titled 'Device type 'Keyapp ETSI' - Callbacks' and includes a green 'New' button. Below the title, a text block states: 'These callbacks transfer data received from the devices associated to this device type to your infrastructure. For more informations, please refer to the [Callback documentation](#)'.

Callback creation

Step 2 : Choose a callback type (e.g. Custom callback)

The screenshot shows the Sigfox web interface for creating a new callback. The left sidebar contains navigation options: INFORMATION, LOCATION, ASSOCIATED DEVICES, DEVICES BEING REGISTERED, STATISTICS, EVENT CONFIGURATION, CALLBACKS (highlighted), and BULK OPERATIONS. The top navigation bar includes tabs for DEVICE, DEVICE TYPE (selected), USER, and GROUP, along with user, alert, and help icons. The main content area is titled 'Device type 'Keyapp ETSI' - New Callback' and contains the following text: 'Create callbacks to connect Sigfox cloud to your server/platform. A callback is a custom http request containing your device(s) data, along with other variables, sent to a given server/platform when the aforesaid device(s) message is received by Sigfox cloud.'

The available callback types are:

- Custom callback**: Creates a new callback from **Sigfox cloud to your own server**. This is the "default" callback type. You can create a full custom request (http method, content type, headers, etc).
- AWS IoT**: AWS IoT is a managed cloud platform that lets connected devices easily and securely interact with cloud applications and other devices. AWS IoT can support billions of devices and trillions of messages, and can process and route those messages to AWS endpoints and to other devices reliably and securely.
- AWS Kinesis**: Amazon Kinesis is a platform for streaming data on AWS, offering powerful services to make it easy to load and analyze streaming data, and also providing the ability for you to build custom streaming data applications for specialized needs.
- Microsoft Azure™ Event hub**: Event Hubs is an event processing service that provides event and telemetry ingress to the cloud at massive scale, with low latency and high reliability. This service is especially useful for: application instrumentation, user experience or workflow processing, Internet of Things (IoT) scenarios.
- Microsoft Azure™ IoT hub**: Azure IoT Hub is a fully managed service that enables reliable and secure communications between millions of IoT devices and a solution back end. Azure IoT Hub enables secure communications using per-device security credentials and access control. Note that the devices are **automatically created** on the IoT hub if needed.

- [Custom Callback Creation](#)
- [Downlink acknowledgement](#)
 - [Geocallback creation](#)
- [Callbacks and connectors](#)

Callback creation

Step 3a : Enter callback information

Device type Keyapp_PA_BSS - Callback new

Callbacks

Type

Channel

Send duplicate

Callback type (DATA, SERVICE, ERROR)

Communication channel (URL, Batch_URL, email)

- **DATA – Uplink** : send uplink messages to customer platform – **BIDIR** : send uplink messages to customer platform and wait for DOWNLINK messages from the same platform;
- **ERROR** : in case of communication failure, it allows to know if it is a device (based on keepalive value defined in the device type edition page) or a network issue
- **SERVICE** : provide additional services based on service messages or network information
 - **STATUS**: device battery and temperature information provided by service messages (e.g. keepalive messages)
 - **ACKNOWLEDGE**: status about the downlink emission. This does not ensure that the device received the message
 - **REPEATER**: service messages (battery, number of repeated messages,...) from repeater devices
 - **DATA_ADVANCED**: Some variables are computed over the different BS which received the messages and thus this callback is delayed by approximately 30s. The list of available variables is displayed on the backend upon creation.

Using batch_URL is strongly recommended to limit the number of request when retrieving messages. Batch_URL gathers messages within 1 seconds prior to sending the HTTP request.

Callback creation

Step 3b : Enter callback information

Device type Keyapp_PA_BSS - Callback new

Callbacks

Type:

Channel:

Send duplicate:

Custom payload config:

URL syntax: `http://host/path?id={device}&time={time}&key1={var1}&key2={var2}...`
Available variables: device, time, duplicate, snr, station, data, avg5nr, lat, lng, rssi, seqNumber
Custom variables:

Url pattern:

Use HTTP Method:

Send SNI: (Server Name Indication) for SSL/TLS connections

Headers: header value

Content type:

Body:

Callback mode (Uplink or BIDIR)

Callback is triggered for each duplicate (same message received by a different BS)

Customized payload decoding

HTTP method (GET, POST, PUT)

Variables to be used in callback

HTTP body (if applicable)

Callback creation

Step 4 : Check that Callback is ENABLED and downlink (if BIDIR callback configured)

Device type 'Keyapp_PA_BSS' - Callbacks

New

These callbacks transfer data received from the devices associated to this device type to your infrastructure. For more informations, please refer to the [Callback documentation](#)

DATA callbacks

Downlink	Enable	Channel	Subtype	Duplicate	Batch	Information	Edit	Errors	Delete
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		UPLINK	<input type="checkbox"/>	<input type="checkbox"/>	Test (john.doe@sigfox.com) Test {device}			

Link to Callback documentation

Thank you!