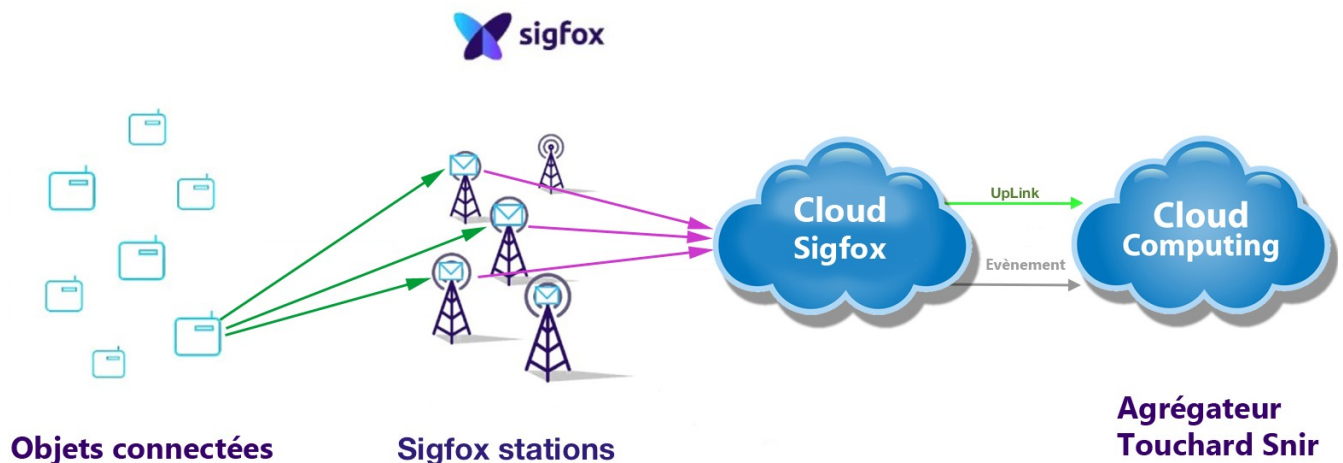


Gérez les rappels du backend Sigfox vers l'agrégateur

1 Vue d'ensemble

Sigfox a été créée en 2010 et est un pionnier mondial reconnu et un leader dans le secteur de la connectivité des appareils. L'ambition de Sigfox, incarnée par le slogan « Make Things Come Alive », est de donner une voix au monde physique qui nous entoure et de permettre à ces millions et un jour milliards d'objets de jouer un rôle dans le développement économique et social.



La Ruche connectée lit les capteurs et envoie les valeurs (dans un *message*) aux stations de base Sigfox pour être retransmis au Cloud Sigfox. À ce stade, le message est dans le cloud Sigfox, il est donc nécessaire de transférer le précieux message vers la plate-forme Cloud Computing en utilisant un Callback (c'est une fonction de rappel qui décode et transfère le message reçu).

Sur le cloud Sigfox, une entreprise est représentée par un «Groupe», qui contient au moins un «Type d'appareil». Chaque type d'appareil peut être attribué à une «famille» d'appareils comme par exemple les ruches connectées. Un type d'appareil regroupe toutes les unités d'un même produit (Ruche Oizé, Ruche Muguet ...), pour leur permettre de se comporter exactement de la même manière lorsque le réseau Sigfox reçoit un message. C'est ce qu'on appelle un comportement de type appareil. De cette façon il est très simple d'ajouter une nouvelle ruche sans avoir à redéfinir son comportement.

Une intégration basée sur le rappel est définie par type d'appareil.

Les callbacks Sigfox utilisent des requêtes HTTP pour transmettre des données de manière bidirectionnelle vers n'importe quel service cloud. Dans cet exemple, nous allons montrer dans une première partie comment configurer le Callback Sigfox pour transmettre les données décodées à l'agrégateur de la section snir. Puis dans une deuxième partie comment configurer un rappel d'événement pour faire envoyer un SMS par l'agrégateur.

Lorsque le cloud Sigfox reçoit un message d'un appareil émetteur, il génère instantanément un message de rappel et l'envoie au serveur agrégateur Touchard Snir. De cette façon, il n'y a pas à vérifier régulièrement que des nouvelles données soient présentes.

Les rappels peuvent être de trois types: données, service et erreur.

Type de rappel		La description
DONNÉES	UPLINK	Envoie les messages de liaison montante à la plateforme Agrégateur.
	BIDIR	Envoie les messages de liaison montante à la plate-forme Agrégateur et attend un message de liaison descendante de la même plate-forme.
	STATUT	Pour récupérer les informations sur la batterie et la température de l'appareil fournies par les messages de service (par exemple, les messages de maintien en vie).
SERVICE	RECONNAÎTRE	Pour récupérer l'accusé de réception du réseau, confirmer la transmission du message de liaison descendante.
	DATA_ADVANCED	Pour récupérer des données facultatives, telles que la géolocalisation , ainsi que des informations de métadonnées. La liste des variables disponibles s'affiche sur le backend lors de la création.
		Certaines variables sont calculées sur les différentes stations de base qui ont reçu les messages. Ce rappel est retardé d'environ 30 s.
ERREUR		En cas d'échec de la communication, il indique s'il s'agit d'un problème de périphérique (en fonction de la valeur de maintien en vie définie dans la page d'édition de l'appareil) ou d'un problème de réseau
UN ÉVÉNEMENT		Être alerté lors de l'occurrence d'un événement sur l'appareil. Cela peut être configuré au niveau du groupe, du type de périphérique ou du périphérique.

Pour optimiser l'utilisation des ressources, il est recommandé que les clients intègrent uniquement avec le service de rappel de données avancé pour collecter les données, notamment pour les cas commerciaux où le délai de livraison n'est pas critique, où un délai de 30 secondes est acceptable ET lorsqu'un mécanisme de liaison descendante est non requis.

Ce guide suppose que l'appareil est déjà enregistré et envoie déjà des données au backend Sigfox.

Ce faisant, vous serez capable de traiter les données envoyées de la ruche connectée et de rendre les informations collectées de manière intelligible pour l'utilisateur final - après le décodage de la trame hexadécimale.

2 Les messages de données Sigfox

Comme vous le savez peut-être déjà, le réseau Sigfox permet d'envoyer sur la liaison montante jusqu'à 12 octets par message avec un nombre maximum de 140 messages par jour.

À ce stade, il est important de préciser que les appareils Sigfox transmettent des données dans différentes structures de charge utile. Chaque type d'objets définit sa propre structure de données.

Dans notre cas la structure de données envoyées via Sigfox pour la surveillance de la batterie et les mesures internes à la ruche aura la forme suivante :

```
typedef struct {  
    short field1;  
    short field2;  
    short field3;  
    unsigned short field5;  
    short field6;  
    unsigned char field4;  
    unsigned char type;  
} trame;
```

Le contenu du message en hexadécimal est par exemple pour la batterie :

b304f2ff53ff3e0b00005202

field1	: 0xb304	→ 0x04B3	→ 1203	octet0 (2 octets)	Tension batterie 12,03V
field2	: 0xf2ff	→ 0xffff	→ -14	octet2 (2 octets)	Courant batterie -0,14 A
field3	: 0x53ff	→ 0xff53	→ -173	octet4 (2 octets)	Puissance -0,173 W
field5	: 0x3e0b	→ 0x0b3e	→ 2878	octet6 (2 octets)	Charge 2,878 Ah
field6	: 0x0000	→ 0x0000	→ 0	octet8 (2 octets)	NU
field4	: 0x52	→ 0x52	→ 82	octet10 (1 octet)	Taux de charge 82/2 = 41 %
type	: 0x02	→ 0x02	→ 02	octet10 (1 octet)	Type de trame (2 batterie)

Autre exemple de message pour les mesures internes à la Ruche

3d11fb0d9e277080ba082e01

field1	: 0x3d11	→ 0x113d	→ 4413	octet0 (2 octets)	Poids 44,13 Kg
field2	: 0xfb0d	→ 0x0dfb	→ 3579	octet2 (2 octets)	Température 35,79°C
field3	: 0x9e27	→ 0x279e	→ 10142	octet4 (2 octets)	Pression 1014,2 hPa
field5	: 0x081f	→ 0x1f08	→ 7944	octet6 (2 octets)	Eclairement 7944 Lux
field6	: 0xba08	→ 0x08ba	→ 2234	octet8 (2 octets)	Point de rosée 22,44°C
field4	: 0x2e	→ 0x2e	→ 46	octet10 (1 octet)	humidité = 46 %
type	: 0x01	→ 0x01	→ 01	octet10 (1 octet)	Type de trame (1 mesures)

Ces données sont interprétées, comprises et décodées par le backend Sigfox.

Le Callback Sigfox est la fonctionnalité qui, lorsqu'elle est utilisée correctement, permet de relayer des données vers agrégateur en utilisant le protocole HTTP.

3 Configuration d'un rappel data uplink vers Agrégateur

Pour commencer, accédez au backend Sigfox

The screenshot shows the Sigfox backend interface. The top navigation bar has tabs: **DEVICE** (highlighted), DEVICE TYPE, USER, and GROUP. The left sidebar has a purple header with 'DEVICES' and 'DELETED DEVICES'. The main content area is titled 'Device' and includes buttons: New, New series, Edit series, Transfer series, Replace series, Delete series, and a dropdown menu set to 'SIGFOX_France'. Below these are search filters: Id, State (All), Last seen from date, and Last seen to date. There are also icons for folder, save, delete, CSV, and buttons for RESET and FILTER. A status bar shows 'Count: 2 / 2' and 'page 1'. The table below lists two devices:

Communication status	Device type	Group	Id	Last seen	Name	Token state	Activation date
<input type="radio"/>	SNOC_DevKit_2	Touchard Informatique Réseaux	C32B3A	2020-08-11 11:00:04	Ruche Beaufay	<input checked="" type="checkbox"/>	2020-08-03 21:23:26
<input type="radio"/>	Type Ruche	Touchard Informatique Réseaux	C32B57	2020-08-11 11:00:04	Ruche_Olize	<input checked="" type="checkbox"/>	2020-07-18 23:21:16

Accédez à la section **Device** et sélectionnez le **Device type** de l'appareil enregistré:

Ensuite, vérifiez les informations de votre appareil et sélectionnez " Callbacks " dans le menu sur le côté gauche de la page, comme indiqué ci-dessous:

[INFORMATION](#)
[LOCATION](#)
[ASSOCIATED DEVICES](#)
[DEVICES BEING REGISTERED](#)
[STATISTICS](#)
[EVENT CONFIGURATION](#)
[CALLBACKS](#)
[BULK OPERATIONS](#)

Device type 'Type Ruche' - Information

Id:

Name: Type Ruche

Description: DevKit 1 (SNOC)

Keep alive: N/A

Subscription automatic renewal: ☒ ⓘ

Group: Touchard Informatique Réseaux

Payload display: None

Downlink mode: CALLBACK

Contracts:

1.

Alert Email:

Creation date: 2020-07-18 23:11:25

Created by: buy.sigfox.com

Last edition date: 2020-08-10 05:59:50

Last edited by: PHILIPPE SIMIER

Utilisez les paramètres suivants:

Type: DATA -UPLINK

Canal: URL

Configuration de la charge utile personnalisée:

field1::int:16:little-endian field2::int:16:little-endian field3::int:16:little-endian field5::uint:16:little-endian field6::int:16:little-endian field4::uint:8 type::uint:8

methode : POST

Modèle d'URL: http://195.221.61.190/Ruche/sigfox

La configuration de charge utile permet de décoder les différents champs de la structure.

Le 3 premiers champs sont de type short équivalent à int:16:little-endian

Le quatrième champ est un unsigned short équivalent à uint:16:little-endian

Le 5eme champ est de type short équivalent à int:16:little-endian

le 6 et 7 eme champs sont de type unsigned char équivalent à uint:8

les valeurs décodées customData#field1 à 6 et customData#type sont renvoyée dans le corps de la requête POST avec id du transmetteur le timestamp et le sequence number.

id={device}&time={time}&data={data}&seqNumber={seqNumber}&field1={customData#field1}&field2={customData#field2}&field3={customData#field3}&field4={customData#field4}&field5={customData#field5}&field6={customData#field6}&type={customData#type}

Après avoir configuré le rappel, cela devrait ressembler à ceci:

Device type Type Ruche - Callback edition

Callbacks

Type

DATA

UPLINK

Channel

URL

Custom payload config

field1::int:16:little-endian field2::int:16:little-endian field3::int:16:little-endian field5::uir

URL syntax: http://host/path?id={device}&time={time}&key1={var1}&key2={var2}...

Available variables: device, time, data, seqNumber, deviceTypeId

Custom variables: customData#field1, customData#field2, customData#field3, customData#field5, customData#field6, customData#field4, customData#type

Url pattern

http://195.221.61.190/Ruche/sigfox

Use HTTP Method

POST

Send SNI

☒ (Server Name Indication) for SSL/TLS connections

Headers

header

value

Content type

application/x-www-form-urlencoded

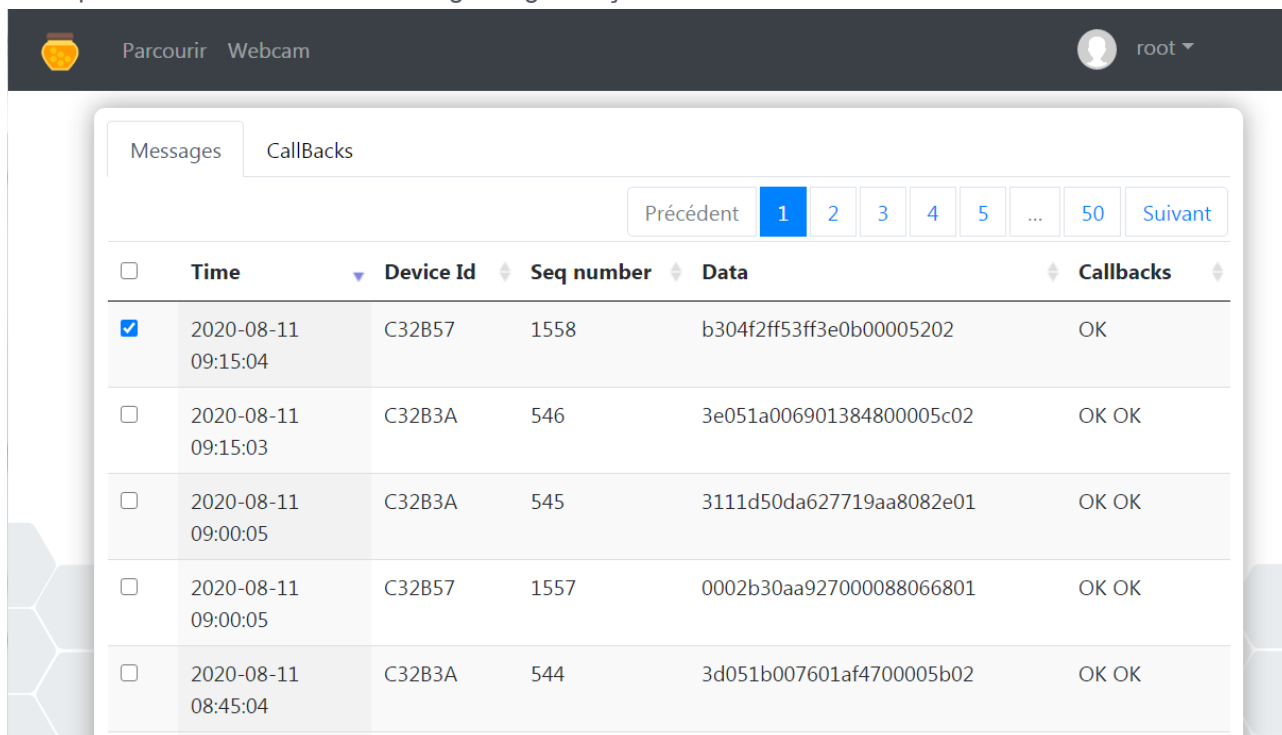
Body

id={device}&time={time}&data={data}&seqNumber={seqNumber}&field1={customData#field1}&field2={customData#field2}&field3={customData#field3}&field4={customData#field4}&field5={customData#field5}&field6={customData#field6}&type={customData#type}

À ce stade, les données vont être décodées en fonction du **custom payload config** configuré

4 Vérification des données reçus dans L'agrégateur

Après l'envoi matériel d'une nouvelle valeur, vous remarquerez qu'une nouvelle ligne a été ajoutée automatiquement au tableau des messages Sigfox reçus.



The screenshot shows the 'Messages' tab in the Sigfox aggregator interface. At the top, there are navigation links 'Parcourir' and 'Webcam', and a user profile 'root'. Below the tabs, there is a pagination bar with 'Précédent', '1' (selected), '2', '3', '4', '5', '...', '50', and 'Suivant'. The table has columns: 'Time', 'Device Id', 'Seq number', 'Data', and 'Callbacks'. The first row is selected with a checkbox.

<input type="checkbox"/>	Time	Device Id	Seq number	Data	Callbacks
<input checked="" type="checkbox"/>	2020-08-11 09:15:04	C32B57	1558	b304f2ff53ff3e0b00005202	OK
<input type="checkbox"/>	2020-08-11 09:15:03	C32B3A	546	3e051a006901384800005c02	OK OK
<input type="checkbox"/>	2020-08-11 09:00:05	C32B3A	545	3111d50da627719aa8082e01	OK OK
<input type="checkbox"/>	2020-08-11 09:00:05	C32B57	1557	0002b30aa927000088066801	OK OK
<input type="checkbox"/>	2020-08-11 08:45:04	C32B3A	544	3d051b007601af4700005b02	OK OK

Pour chaque message les informations suivantes sont disponibles :

la date et l'heure de la reception par le backend sigfox;

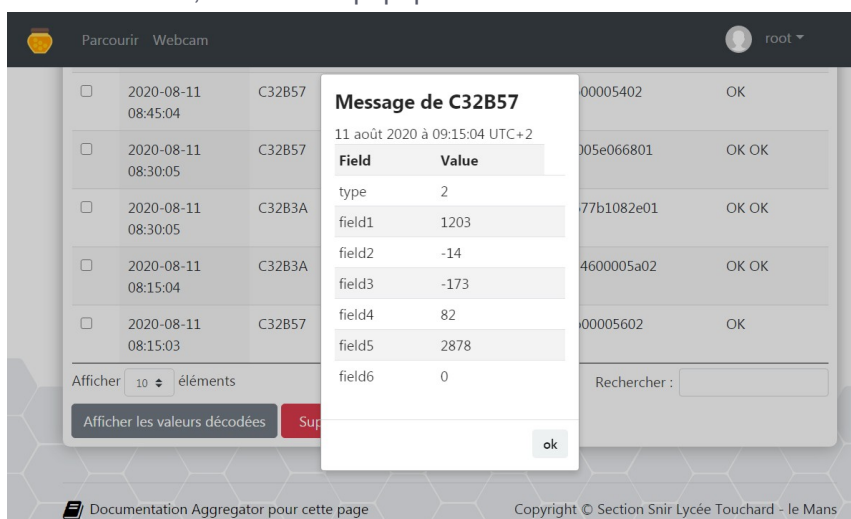
Le **device id** est l'identifiant Sigfox indiqué sur l'étiquette du transmetteur. Il s'agit généralement de 6 caractères (0-9 et AF, par exemple 21F657), mais peut comporter jusqu'à 8 caractères ;

Le **Seq number** la valeur s'incrémente d'une unité entre deux messages consécutifs ;

Les **data** au format brut en hexadécimal ;

le résultat de l'exécution des fonctions callbacks de l'agrégateur.

Vous pouvez visualiser les valeurs des données décodées en cochant un message puis en cliquant sur le bouton Afficher les valeurs décodées, une fenêtre popup s'ouvre.



The screenshot shows the same interface as before, but with a popup window titled 'Message de C32B57' open. The popup displays the date and time '11 août 2020 à 09:15:04 UTC+2' and a table of decoded fields.

Field	Value
type	2
field1	1203
field2	-14
field3	-173
field4	82
field5	2878
field6	0

At the bottom of the popup is an 'ok' button. In the background, the table of messages is visible, and the first row is still selected. Below the table, there is a search bar and a button 'Afficher les valeurs décodées'.

Les callbacks de l'agrégateur appliquent un traitement aux données reçues avant de les enregistrer dans un canal ou de les retransmettre vers un autre agrégateur. Le traitement permet par exemple de changer l'unité des champs comme par exemple passer du déci-degré en degré.

5 Configuration d'un rappel type événement

La configuration d'un rappel permet d'être alerté lors de l'occurrence d'un événement sur l'appareil.

Nous présenterons ici un rappel pour notifier par SMS d'un écart dans la continuité du numéro de séquence.

Le numéro de séquence fait partie de la surcharge du protocole Sigfox et est incrémenté par l'appareil lui-même à chaque émission de message. Un numéro de séquence cohérent reçu avec un message est enregistré comme référence. Un écart dans la continuité du numéro de séquence peut être dû à l'appareil envoyant des messages à partir d'une zone non couverte.

Événement d'avertissement

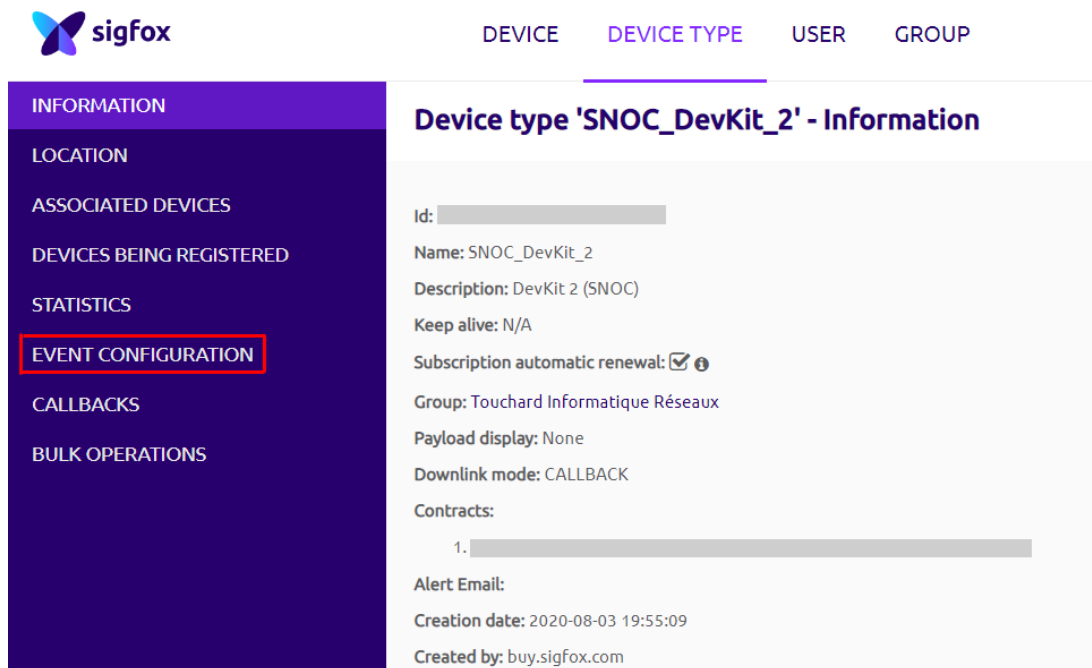
L'événement d'avertissement "**rupture de séquence de messages**" est purement informatif, il permet d'être informé d'une rupture de continuité des messages.

Événement d'erreur

L'événement d'erreur "hors séquence de messages" provoquera une interruption de la livraison du message, le cloud n'a pas pu évaluer que le message provenait bien de l'appareil lui-même. Les messages entrants seront rejetés jusqu'à ce que la fonction de numéro de séquence de désengagement soit utilisée.

Le fait que l'écart du numéro de séquence déclenche un avertissement ou un événement d'erreur dépend de l'ampleur de l'écart, et des jours écoulés.

Accédez à la section **Device** et sélectionnez le **Device type** de l'appareil enregistré:



The screenshot shows the Sigfox web interface. At the top, there's a navigation bar with the Sigfox logo and tabs for 'DEVICE', 'DEVICE TYPE', 'USER', and 'GROUP'. The 'DEVICE TYPE' tab is selected. On the left, a sidebar menu lists various options: 'INFORMATION', 'LOCATION', 'ASSOCIATED DEVICES', 'DEVICES BEING REGISTERED', 'STATISTICS', 'EVENT CONFIGURATION' (highlighted with a red box), 'CALLBACKS', and 'BULK OPERATIONS'. The main content area is titled 'Device type 'SNOC_DevKit_2' - Information' and displays the following details:

- Id:** [Redacted]
- Name:** SNOC_DevKit_2
- Description:** DevKit 2 (SNOC)
- Keep alive:** N/A
- Subscription automatic renewal:** ☒ ⓘ
- Group:** Touchard Informatique Réseaux
- Payload display:** None
- Downlink mode:** CALLBACK
- Contracts:**
 - 1. [Redacted]
- Alert Email:**
- Creation date:** 2020-08-03 19:55:09
- Created by:** buy.sigfox.com

Utilisez les paramètres suivants:

Event type : Break in message sequence

Channel : URL

Url pattern : <http://touchardinforesseau.servehttp.com/Ruche/api/sendSMS>

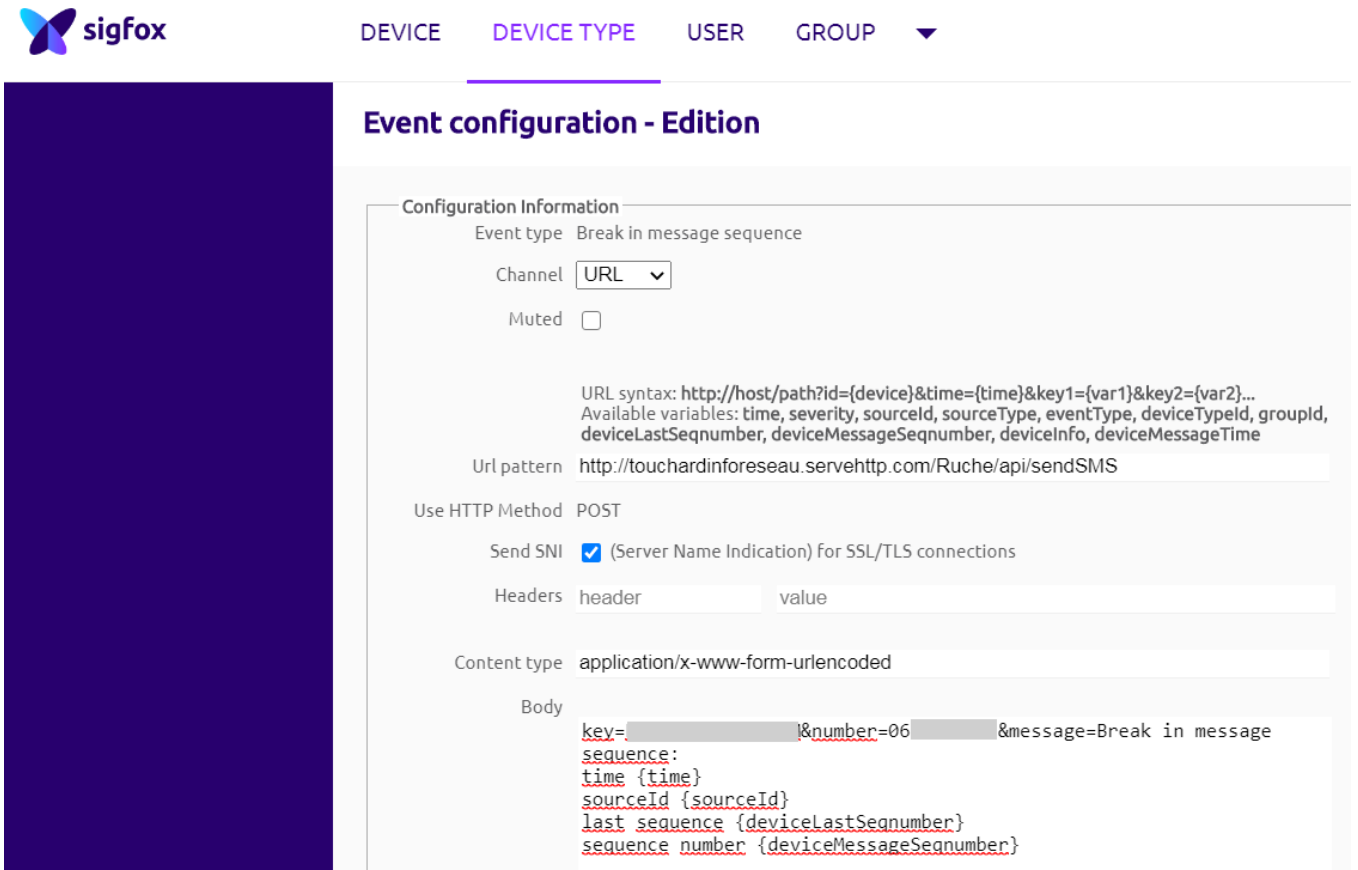
Use HTTP Method : POST

Content type : application/x-www-form-urlencoded

Body :

```
key=XXXXXXXXXX&number=0612345678&message=Break in message sequence:
time {time}
sourceId {sourceId}
last sequence {deviceLastSeqnumber}
sequence number {deviceMessageSeqnumber}
```

Après avoir configuré le rappel, cela devrait ressembler à ceci:



Event configuration - Edition

Configuration Information

Event type Break in message sequence

Channel

Muted ☐

URL syntax: `http://host/path?id={device}&time={time}&key1={var1}&key2={var2}...`
Available variables: `time, severity, sourceId, sourceType, eventType, deviceType, groupId, deviceLastSeqnumber, deviceMessageSeqnumber, deviceInfo, deviceMessageTime`

Url pattern

Use HTTP Method POST

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

Headers

header	value
--------	-------

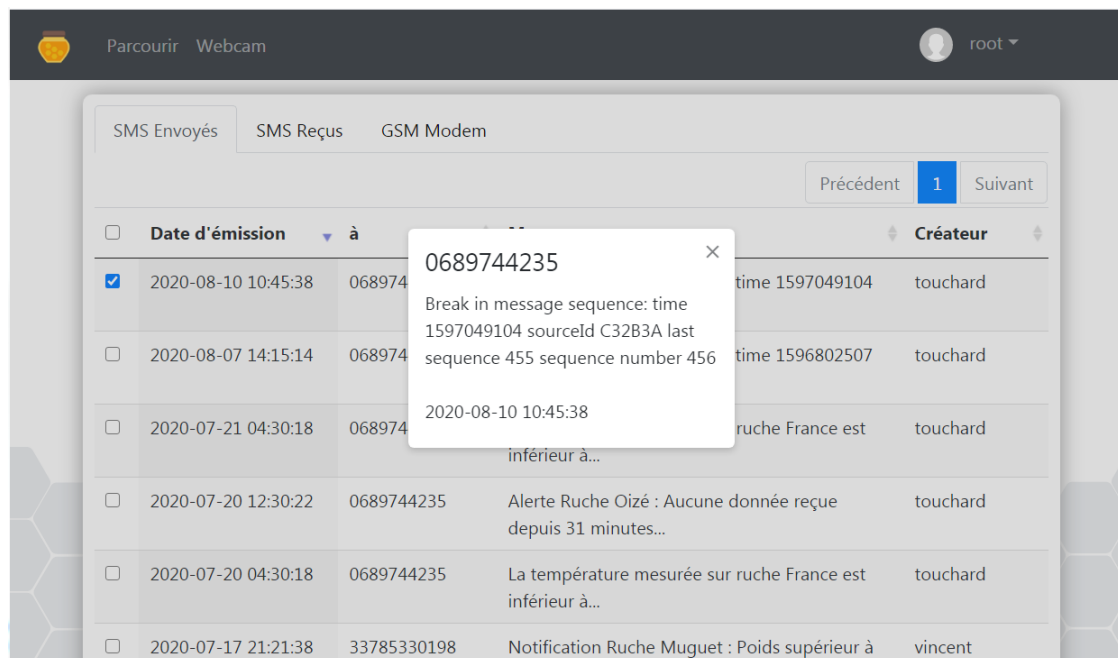
Content type

Body

```
key={device}&number=06{device}&message=Break in message  
sequence:  
time {time}  
sourceId {sourceId}  
last sequence {deviceLastSeqnumber}  
sequence number {deviceMessageSeqnumber}
```

6 Vérification des SMS envoyés

Chaque événement provoque l'envoi d'une notification par SMS. Le backend de l'agrégateur permet de visualiser les SMS archivés .



SMS Envoyés		SMS Reçus	GSM Modem
Précédent		1	Suivant
<input type="checkbox"/>	Date d'émission	à	Créateur
<input checked="" type="checkbox"/>	2020-08-10 10:45:38	0689744235	touchard
<input type="checkbox"/>	2020-08-07 14:15:14	0689744235	touchard
<input type="checkbox"/>	2020-07-21 04:30:18	0689744235	touchard
<input type="checkbox"/>	2020-07-20 12:30:22	0689744235	touchard
<input type="checkbox"/>	2020-07-20 04:30:18	0689744235	touchard
<input type="checkbox"/>	2020-07-17 21:21:38	33785330198	vincent

0689744235

Break in message sequence: time 1597049104 sourceId C32B3A last sequence 455 sequence number 456

2020-08-10 10:45:38

7 Créer un rappel DATA_ADVANCED

Le rappel data-advanced permet de retransmettre les métadonnées. Il est déclenché environ 30 secondes après le rappel DATA. Ce délai permet de garantir que toutes les stations ont signalé la réception des trames. Ces informations sont vraiment importantes pour la gestion des appareils

1 les métadonnées liées au signal

Les informations de métadonnées sont parfois aussi importantes que les données elles-mêmes. Dans le cas des métadonnées Sigfox, nous avons l'ID de l'appareil, l'horodatage du message. Nous avons également des informations sur le signal radio: la force du signal (RSSI), le rapport signal sur bruit (SNR), la station Sigfox recevant ce signal.

Vous pouvez choisir d'ignorer ces informations ou de les traiter. À mon avis, ignorer ces informations est une erreur pour les raisons suivantes:

- Le nombre de stations et la qualité du signal nous donnent une bonne idée de la qualité de réception et du risque de perte de trame.
- L'identification de la station nous permet de savoir si l'appareil est mobile ou fixe, Il peut également être utilisé pour positionner un appareil.

L'utilisation de la méthode POST ou PUT est obligatoire.

Le corps de la requête est obligatoirement au format JSON.

```
{
  "device" : "{device}",
  "time" : "{time}",
  "data" : "{data}",
  "seqNumber" : "{seqNumber}",
  "lqi" : "{lqi}",
  "operatorName" : "{operatorName}",
  "countryCode": {countryCode},
  "computedLocation" : {computedLocation}
}
```

- lqi: Qualité du lien - une chaîne de texte brut comme "Excellent"
- operatorName - une chaîne de texte brut comme "SIGFOX_France"
- countryCode - le code du pays selon la [norme numérique ISO 3166-1](#).

La structure **computedLocation** est une structure prédéfinie JSON.

```
{
  "lat":48.00511506226378,
  "lng":0.19602204603779083,
  "radius":5700,
  "source":2,
  "status":1
}
```

Cette structure peut être ajoutée au rappel DATA_ADVANCED lorsque vous avez une option de localisation Atlas activée dans votre contrat.

Les champs **lat** et **lng** donnent la position du centre d'un cercle.

Le champ **radius** donne le rayon en mètres dans lequel potentiellement l'appareil se trouve réellement. Comprenez le rayon comme l'erreur sur l'emplacement détecté.

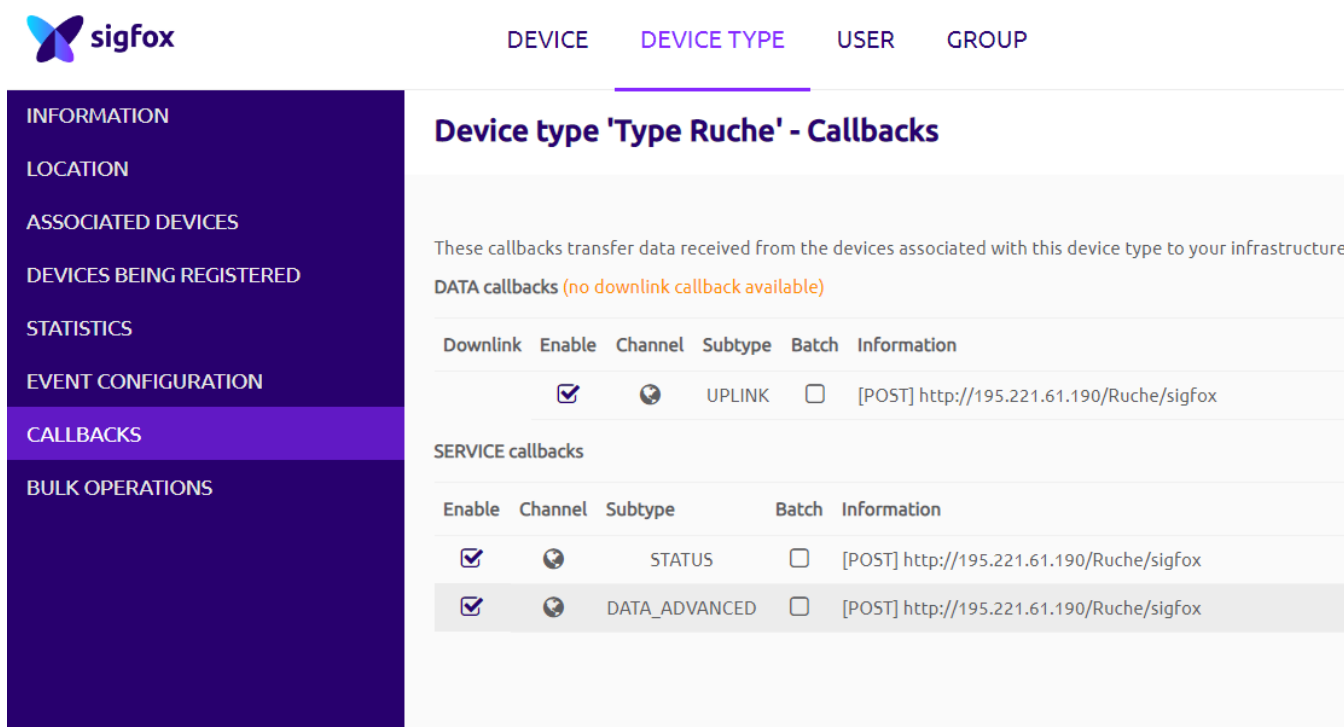
le champ **status** est mis à 1 lorsqu'une position a été calculée. 0 lorsqu'aucune position n'a été calculée. Dans ce cas, seul le champ «status» est renvoyé avec la valeur 0. Rien de plus.

Le champ **source** indique la méthode utilisée pour calculer de la localisation.

Si la source est définie sur :

- 1, l'emplacement est calculé à l'aide du GPS à l'intérieur de la charge utile de données sur la liaison montante.
- 2, l'emplacement est calculé par Atlas Network.
- 3, l'emplacement est calculé par Atlas POI
- 4, l'emplacement est calculé par Atlas HD.

Accédez à la section **Device** et sélectionnez le **Device type** de l'appareil enregistré:
Cliquez sur CALLBACKS



Device type 'Type Ruche' - Callbacks

These callbacks transfer data received from the devices associated with this device type to your infrastructure.

DATA callbacks (no downlink callback available)

Downlink	Enable	Channel	Subtype	Batch	Information
	<input checked="" type="checkbox"/>		UPLINK	<input type="checkbox"/>	[POST] http://195.221.61.190/Ruche/sigfox

SERVICE callbacks

Enable	Channel	Subtype	Batch	Information
<input checked="" type="checkbox"/>		STATUS	<input type="checkbox"/>	[POST] http://195.221.61.190/Ruche/sigfox
<input checked="" type="checkbox"/>		DATA_ADVANCED	<input type="checkbox"/>	[POST] http://195.221.61.190/Ruche/sigfox

Cliquez sur New et utilisez les paramètres suivants :

Channel : URL

Url pattern : **http://195.221.61.190/Ruche/sigfox**

Content type : **application/json**

body :

```
{
  "device" : "{device}",
  "time" : "{time}",
  "data" : "{data}",
  "seqNumber" : "{seqNumber}",
  "lqi" : "{lqi}",
  "operatorName" : "{operatorName}",
  "countryCode": {countryCode},
  "computedLocation" : {computedLocation}
}
```

Vous pouvez vérifier le bon fonctionnement en allant sur la section DEVICE puis MESSAGES

Cliquez sur un callback vous obtiendrez le contenu de la requête envoyé à l'agrégateur comme le montre l'écran suivant :

Device C32B57 - Messages

From date

To date

Callback - OK

[OK] - Base station 1EB3 - 22 seconds

200 - - #1

```
POST http://195.221.61.190/Ruche/sigfox HTTP/1.0
content-length : 314
accept-encoding : gzip,deflate
accept-language : fr
host : 195.221.61.190
accept-charset : UTF-8;q=0.9,*;q=0.7
user-agent : SIGFOX
content-type : application/json
```

```
{
  "device" : "C32B57",
  "time" : "1597339805",
  "data" : "1600a70a96270000bb066c01",
  "seqNumber" : "1775",
  "lqi" : "Good",
  "operatorName" : "SIGFOX_France",
  "countryCode" : 250,
  "computedLocation" :
  {"lat":48.00507367251108,"lng":0.19511333480176982,"radius":5500,"source":2,"status":1}
}
```

2020-08-13 18:30:05 1771 1700c70a962700009c066801

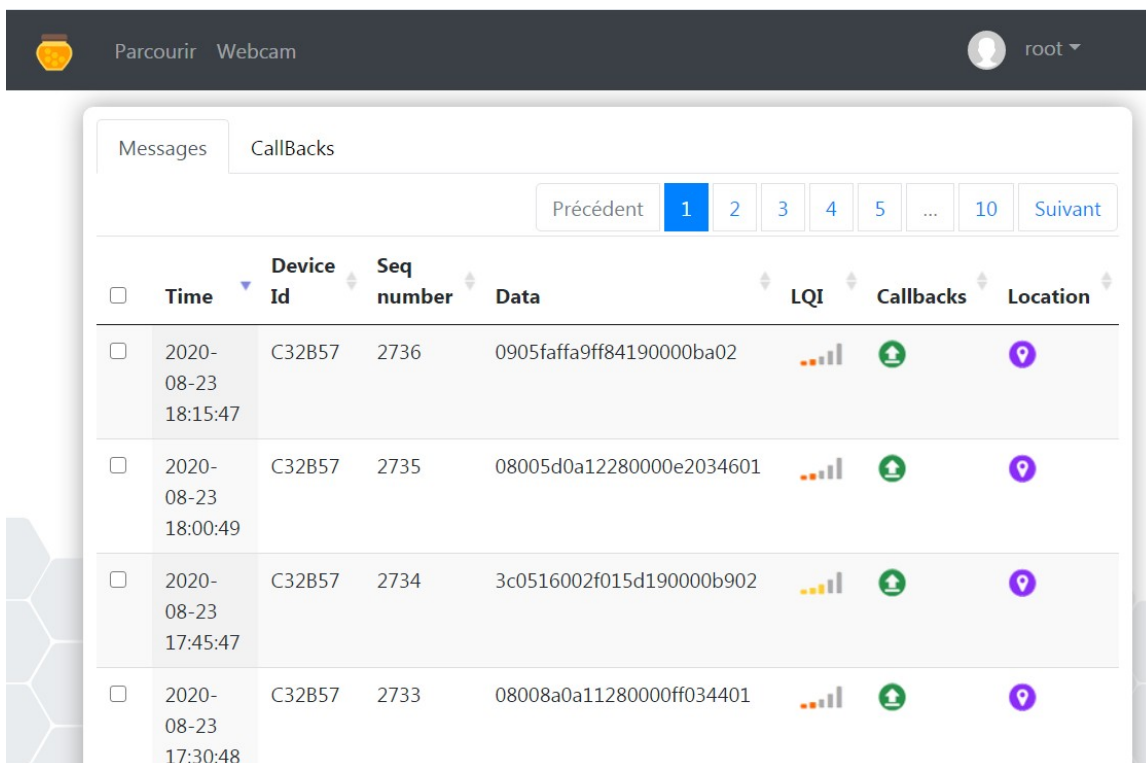


Location



8 Vérification des métadonnées reçues

Sur l'agrégateur, les métadonnées peuvent être visualisées. A partir du tableau des messages reçus pour un objet. Dans le menu choisir Objet puis cliquer sur l'identifiant Sigfox.



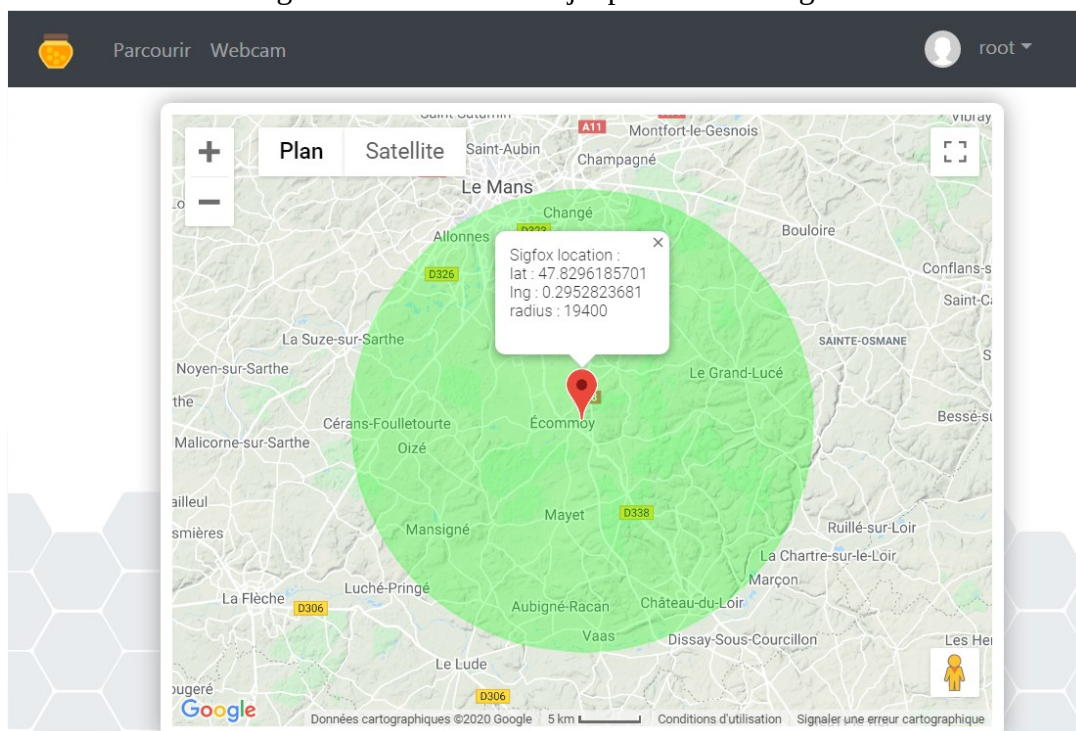
The screenshot shows the 'Messages' tab in the Sigfox aggregator interface. At the top, there are navigation links 'Parcourir' and 'Webcam', and a user profile 'root'. Below the tabs, there is a pagination bar with 'Précédent', '1' (selected), '2', '3', '4', '5', '...', '10', and 'Suivant'. The table below has columns: 'Time', 'Device Id', 'Seq number', 'Data', 'LQI', 'Callbacks', and 'Location'. Each row represents a received message with a checkbox on the left.

<input type="checkbox"/>	Time	Device Id	Seq number	Data	LQI	Callbacks	Location
<input type="checkbox"/>	2020-08-23 18:15:47	C32B57	2736	0905faffa9ff84190000ba02
<input type="checkbox"/>	2020-08-23 18:00:49	C32B57	2735	08005d0a12280000e2034601
<input type="checkbox"/>	2020-08-23 17:45:47	C32B57	2734	3c0516002f015d190000b902
<input type="checkbox"/>	2020-08-23 17:30:48	C32B57	2733	08008a0a11280000ff034401

La colonne LQI donne le niveau de réception.

La colonne Callbacks donne le retour de l'exécution des fonctions de rappel.

La colonne Location donne la géolocalisation de l'objet par le réseau sigfox.



PAC est une clé pour garantir que l'appareil ne peut pas être ajouté de manière malveillante à un nouveau compte. Il est imprimé sur l'étiquette. Il s'agit d'un code alphanumérique de 16 caractères. Notez qu'une fois que vous avez ajouté un appareil à un compte, le PAC changera. Vous pouvez obtenir le nouveau PAC à partir du compte auquel il est actuellement ajouté.