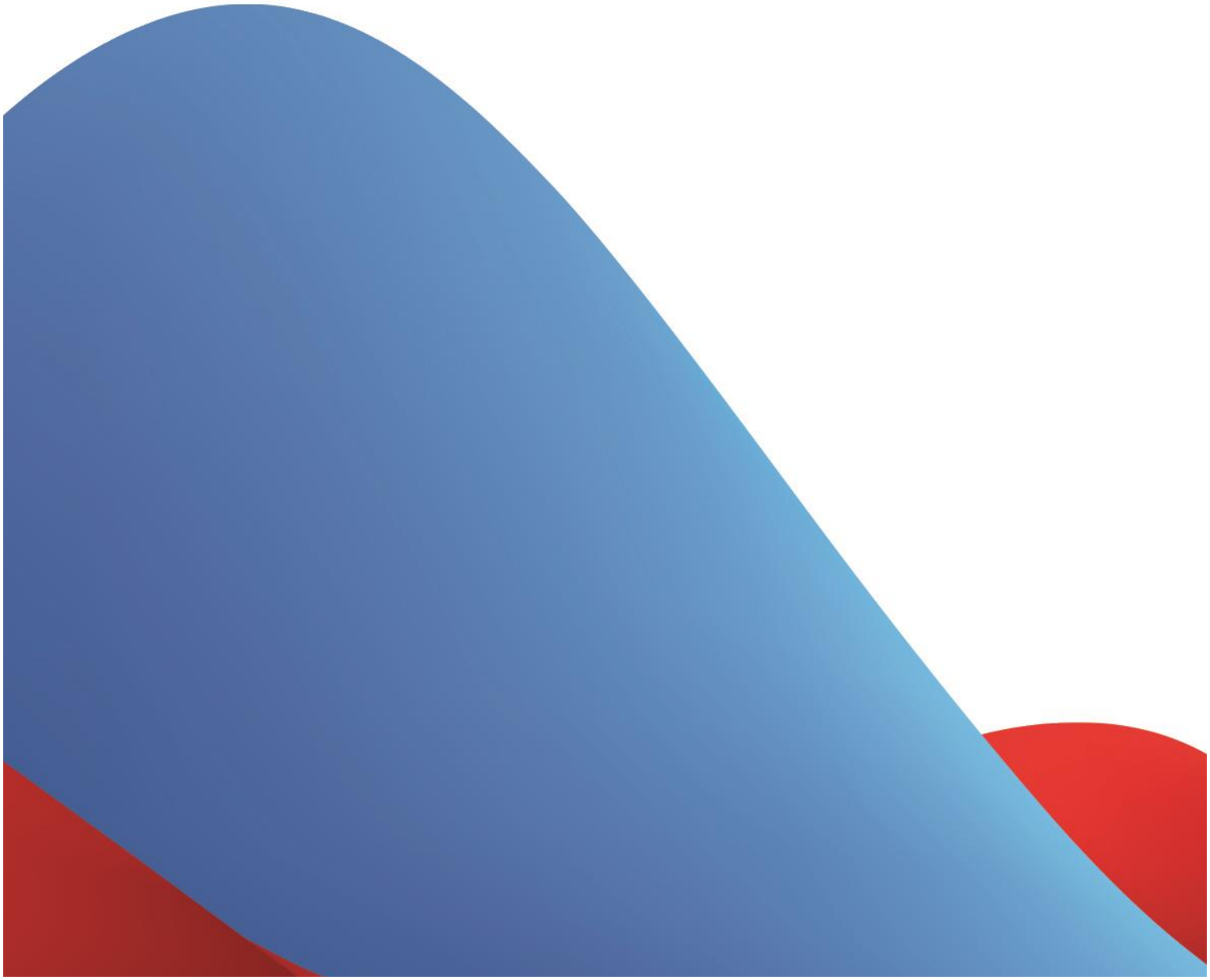
Swisscom LPN Portal

Device Developer Guide

1

Device Manager and Wireless Logger

February 2017



NOTICE

This document contains proprietary and confidential material of Swisscom (Schweiz) Ltd. This document is provided under and governed by either a license or confidentiality agreement. Any unauthorized reproduction, use, or disclosure of this material, or any part thereof, is strictly prohibited.

The material provided in this document is believed to be accurate and reliable. However, no responsibility is assumed by Swisscom (Schweiz) Ltd. for the use of this material. Swisscom (Schweiz) Ltd. reserves the right to make changes to the material at any time and without notice. This document is intended for information and operational purposes only. No part of this document shall constitute any contractual commitment by Swisscom (Schweiz) Ltd.

Portions of this documentation and of the software herein described are used by permission of their copyright owners.

Versions

Version	Date	Author	Details
1	31/08/16	Swisscom	Initial version
1.1	30/10/16	Swisscom	Revised initial version
1.2	31/01/17	Swisscom	New GUI features, new layout
1.3	04/05/17	Swisscom	DX API description

Table of Contents

Definitions and acronyms		6
1.	Scope	7
2.	LoRaWAN specification	7
2.1.	Globally unique EUI-64: DevEUI	8
2.2.	Over-The-Air Activation (OTAA)	9
2.2.1	128 bit application key: AppKey	9
2.2.2	64 bit application server identifier: AppEUI (joinEUI)	9
2.3.	Activation By Personalization (ABP)	10
2.3.1	Device address: DevAddr	10
2.3.2	128 bit network secret: NwkSkey	10
2.4.	Channel plans	11
2.4.1	ETSI EU 868	11
3.	Device Manager	12
3.1.	Devices Creation and Management	13
3.1.1	Device list	13
3.1.2	Device details	14
3.1.3	Device provisioning	17
3.2.	Connectivity plan	20
3.2.1	Connectivity Plan details	21
3.3.	Application servers	22
3.3.1	Create a new application server	23
3.3.2	Edit an Application server	24
3.3.3	Delete an Application server	25
3.4.	AS routing profiles	25
3.4.1	Create an AS Routing Profile	26
3.4.2	Modify or Delete an AS Routing Profile	27
3.4.3	Assigne or Remove an AS Routing Profile	27
4.	Wireless Logger	29
4.1.	Metadata	29
4.2.	Payload	30
5.	Application Server	31
5.1.	Uplink interface:	31

5.1.1	Query parameters:	31
5.1.2	XML payload	31
5.1.3	JSON payload	32
5.2.	Downlink interface	33
6.	DX API	35
6.1.	Authentication	35
6.2.	Get Started page	36

Definitions and acronyms

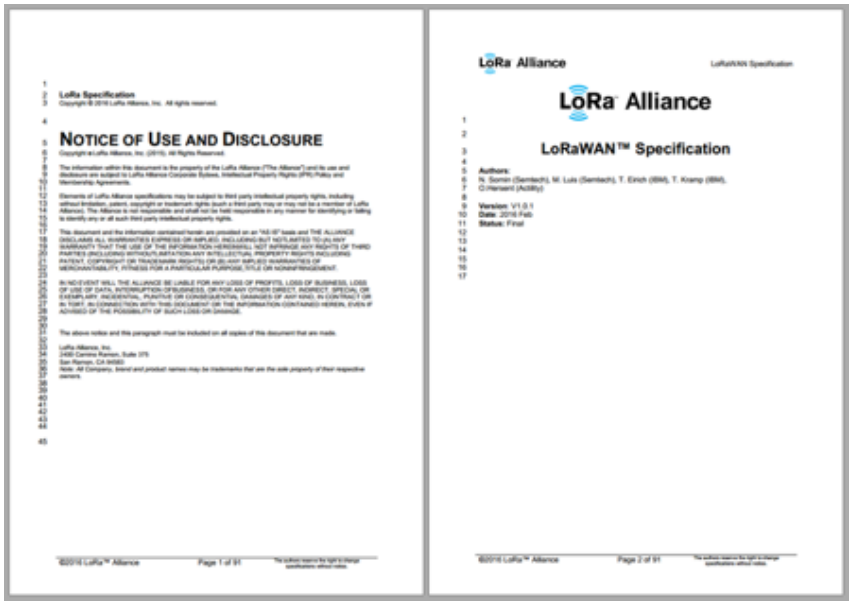
ABP	Activation by Personalization
ADR	Adaptive Data Rate
AES	Advanced Encryption Standard
AS	Application Server
ESP	Estimated Signal Power
LC	Logical Channel
LRC	Long-Range Controller: Network Server
LRR	Long-Range Relay: software inside the gateway
OTAA	Over The Air Activation
RF	Radio Frequency
RSSI	Received Signal Strength Indicator
SF	Spreading Factor
SNR	Signal to Noise Ratio
ABP	Activation by Personalization
ADR	Adaptive Data Rate
AES	Advanced Encryption Standard
AS	Application Server
ESP	Estimated Signal Power
LC	Logical Channel
LRC	Long-Range Controller: Network Server
LRR	Long-Range Relay: software inside the gateway
OTAA	Over The Air Activation
RF	Radio Frequency
RSSI	Received Signal Strength Indicator
SF	Spreading Factor
SNR	Signal to Noise Ratio

1. Scope

The Scope of this Device Developer Guide is to provide the guidelines to a developer during the Swisscom LPN Portal connectivity integration.

2. LoRaWAN specification

The LoRaWAN specification is publicly available from the LoRa™ alliance web site: lora-alliance.org

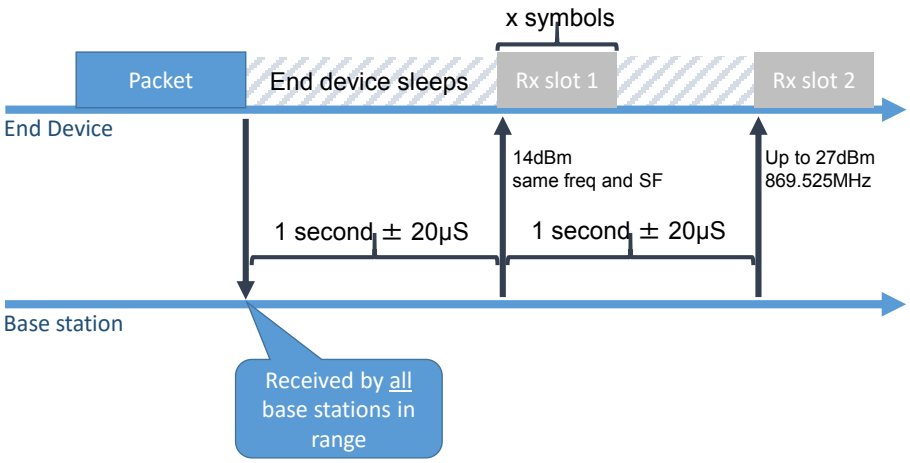


The LoRaWAN specification enables bidirectional communications, and currently offers three variants of the MAC layer: class A, optimized for battery usage and sensors; class B, optimized for battery powered actuators with low command latency requirements, and class C for mains powered devices including actuators.

Class name	Intended usage
A (« all »)	Battery powered sensors, or actuators with no latency constraint Most energy efficient communication class. Must be supported by all devices
B (« beacon »)	Battery powered actuators Energy efficient communication class for latency controlled downlink. Based on slotted communication synchronized with a network beacon.

C (« continuous »)	Mains powered actuators Devices which can afford to listen continuously. No latency for downlink communication.

LoRaWAN class A employs the well-known receiver initiated transmit strategy to enable communication bidirectionality.



The End-device, e.g. a temperature sensor, wakes-up to transmit its measurement. The LoRaWAN radio frame will be received by all nearby base stations of the RF network. The device then immediately goes to sleep for a specified amount of time, by default 1 second, in order to preserve the battery.

After the exact sleep time, the End-device must wake up to receive potential downlink communication from the network. The downlink communication may be an ACK from the network if the End-device had sent a confirmed LoRaWAN frame, or it may be a command from the MAC layer network controller or from an application server.

The Core network will use, at its choice, this first receive window (RX1), or the second receive window (RX2) do send downlink frames. So if no frame has been received during the RX1 slot, the device must go to sleep again and wake up another time for the RX2 slot.

2.1. Globally unique EUI-64: DevEUI

Each LoRaWAN end device has a globally unique IEEE EUI-64 address, the DevEUI. These addresses are allocated by manufacturers within address blocks that must be purchased from IEEE, three blocks are available:

- Organizational Unique Identifier (OUI) / MAC Address Block Large (MA-L)
- MAC Address Block Medium (MA-M)
- MAC Address Block Small (MA-S)

These 3 different blocks of addresses can be purchased from IEEE here:

- http://standards.ieee.org/develop/regauth/oui/
- http://standards.ieee.org/develop/regauth/oui28/index.html
- http://standards.ieee.org/develop/regauth/oui36/index.html

2.2. Over-The-Air Activation (OTAA)

OTAA device derives its NwkSkey and AppSkey using the Join key negotiation procedure as they first attach to a network. This procedure uses a master AppKey secret that must be personalized at production in the device.

A single application server, identified by its AppEUI (JoinEUI), is supported per device.

OTAA	Who?	What is it?
DevEUI	IEEE/Device manufacturer	The DevEUI identifies the device on the LoRaWAN network during the JOIN request
AppEUI (joinEUI)	Operator	The AppEUI identifies the join server during the JOIN request
AppKey	Device manufacturer	The AppKey encrypts the data during the JOIN request

2.2.1 128 bit application key: AppKey

The 128 bit AppKey must be personalized in each device during production. It may be distinct per device or unique per application depending on the use-case.

Typically, the AppKey must be part of the Excel file delivered with each production batch to the customer, so that the customer may be able to provision the AppKey to the application server associated with the Device.

2.2.2 64 bit application server identifier: AppEUI (joinEUI)

The AppEUI is a globally unique identifier of the target application server that will process all exchanges with the device.

The network forwards the join message to the application server identified by the AppEUI. This application server is supposed to have been provisioned with the Device AppKey. Based on the AppKey and the content of the Join message sent by the device, the Application server:

1. Generates a NwkSkey and AppSkey and sends the NwkSkey information to the Core Network
2. Forms a cryptographic Join response payload that will allow the device to compute a NwkSkey and AppSkey

As part of the Join procedure, the network also allocates a DevAddr address to the LoRaWAN device.

The Swisscom AppEUI is F0:3D:29:AC:71:00:00:01

2.3. Activation By Personalization (ABP)

ABP	Who?	What is it?
DevEUI	IEEE/Device manufacturer	DevEUI is not used in LoRa communication in ABP but is used to identify the device at the Network Server side
DevAddr	Operator	The DevAddr is the Device Address on the LPN Network
NwkSKey	Device manufacturer	The NwkSKey encrypts the data during the transmission. Gateways from other networks cannot see the content of messages. The NwkSKey authenticates the device on a LoRa network
AppSKeys	Device manufacturer	The AppSKey encrypts the payload data

2.3.1 Device address: DevAddr

The DevAddr identifies the device on the network, together with the Network secret for the sensor. The group (DevAddr, NwkSkey) must be globally unique.

If the end device is not using the JOIN LoRaWAN procedure, it must also be personalized with the DevAddr.

The DevAddr will be provided by your LoRaWAN network operator.

2.3.2 128 bit network secret: NwkSkey

The 128 bit NwkSkey is used by the Core network to verify the authenticity and integrity of each message. Use a random NwkSkey for each device.

Allocating a random NwkSKey per device is very important for security, but also to ensure that the short address collision resolution algorithm will work appropriately. The pair (DevAddr, NwkSkey) must be globally unique.

128 bit application secret: AppSkey

The 128 bit AppSkey is used to encrypt the payload of messages. You may decide to use a unique AppSkey for all LoRaWAN ports used by your device, or to allocate one AppSKey for each port.

AppSkeys must be known to the Application Server. Commonly AppSkeys are part of a production Excel file providing the associations between DevEUI, DevAddr, NwkSkey, AppSkey(s) of the devices part of the production batch.

When adding the device to your account using the Device Manager application:

> It is not mandatory to provision the AppSkey. The Swisscom LPN Portal will then forward the payload in encrypted form to the application servers and has no access to the payload clear-form content.
> If you provision the AppSkey(s), then the Core Network will decode the payload before forwarding it to the application server(s).

2.4. Channel plans

2.4.1 ETSI EU 868

The implementation in Europe is the following default and mandatory channel plan:

- LC1: 868.1 MHz
- LC2: 868.3 MHz
- LC3: 868.5 MHz
- RX2: 869.525 MHz / SF12

Then the network will configure the device with the operator settings (add new channels, change RX2 configuration).

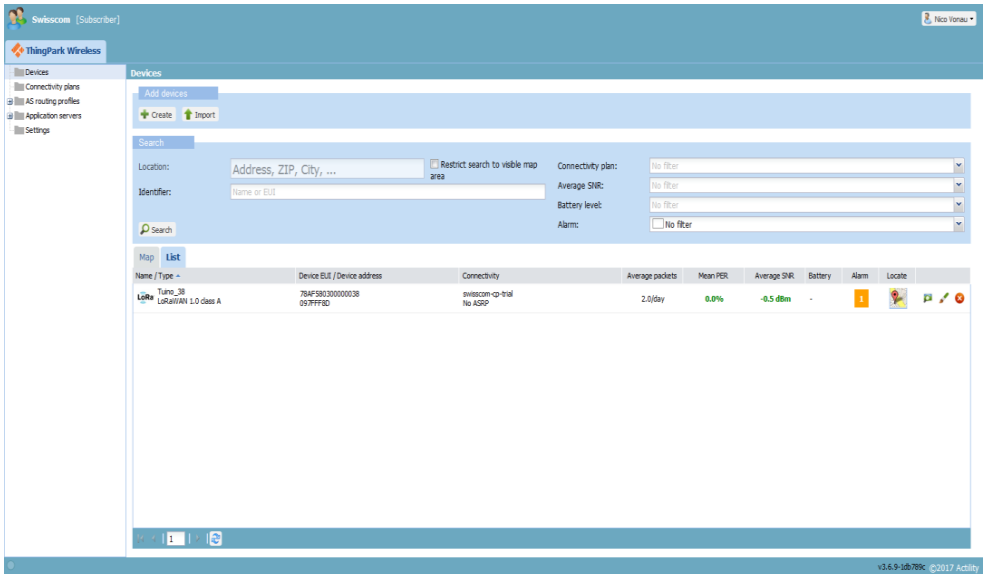
The channels used by the Swisscom LPN:

- LC1: 868.1 MHz
- LC2: 868.3 MHz
- LC3: 868.5 MHz
- LC4: 867.1 MHz
- LC5: 867.3 MHz
- LC6: 867.5 MHz
- LC7: 867.7 MHz
- LC8: 867.9 MHz
- RX2: 869.525 MHz / SF12

3. Device Manager

> Launch your Device Manager via the following link:
https://portal.lpn.swisscom.ch/deviceManager/

Once you login to the Device Management portal, you will get an overview of all the devices of your account. You can easily shift between the Map and List view of devices, by clicking the corresponding tabs.



The interface is based on 2 frames, a left sidebar menu and a main application frame showing device data.

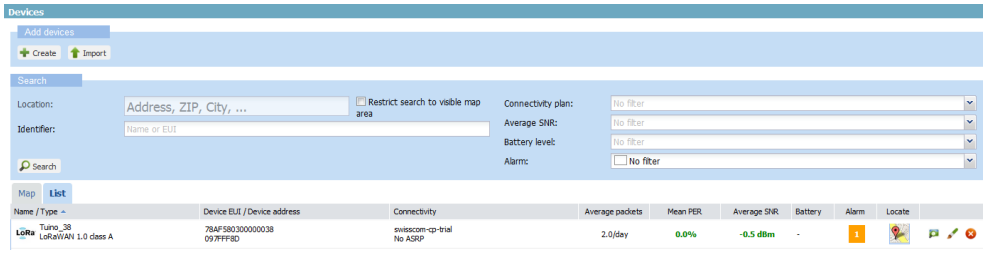
The left sidebar menu gives access to devices, connectivity plans, AS routing profiles, application servers and Settings.

The first main frame contains a Search bar, allowing users to search devices by Location, device Identifier or other filtering criterias.

3.1. Devices Creation and Management

3.1.1 Device list

Device List displays all the filtered devices in a list.

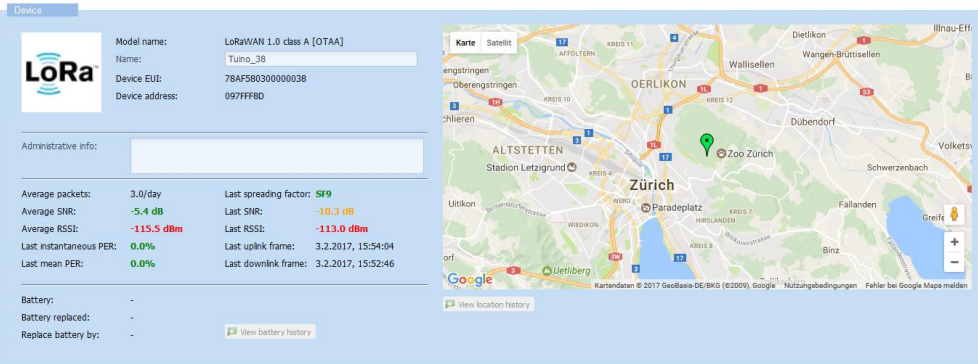


The displayed field are:

- Name / Type: name and device profile
- MAC IEEE address / Device Address: DevEUI and DevAddr of the device
- Connectivity plan / application routing profile
- Mean packet error rate
- Average amount of packets per day
- Average SNR: based on the last 5 packets received
- Battery status
- Alarm: number of alarms not acknowledged
- Locate: open a pop-up and display the device on a map
- Button ( ) to view more info of the device
- Button ( ) to edit the settings of the device
- Button ( ) to delete the device

The filtering/sorting can be done on most of the above mentioned properties. Users can therefore easily display all devices for which e.g. the battery level is low, that have raised critical alarms, etc.

3.1.2 Device details



Device frame

This frame displays the basic information of the device such as the model (device profile), name, DevEUI, DevAddr.

The interface also provides information on the LoRaWAN traffic such as the average number of packets, average RSSI and SNR, last instantaneous/mean PERs, last RSSI/SNR/SF, date/time of the last message received/sent.

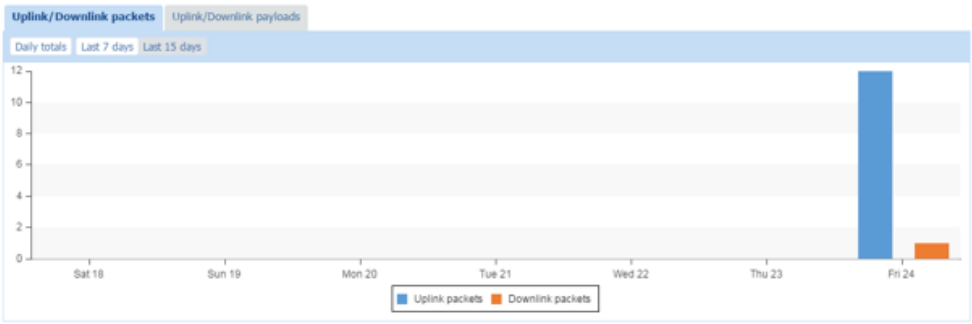
For devices supporting the feature, the battery status information can also be displayed.

The location of the device on the map could be provided in 3 different ways:

- Last GPS position reported by the device
- Manual location
- Gateway position which has received the last message

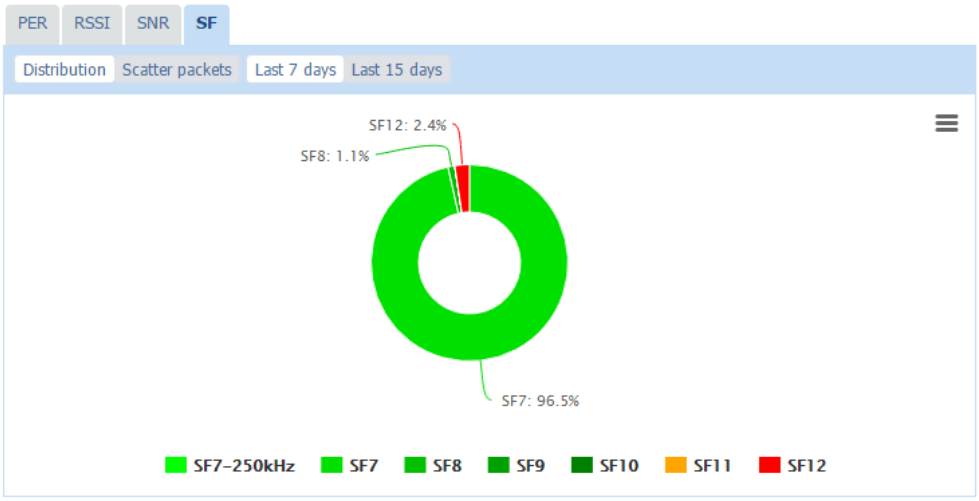
The View location history button displays a map with markers showing where the devices are located.

Uplink/Downlink frame



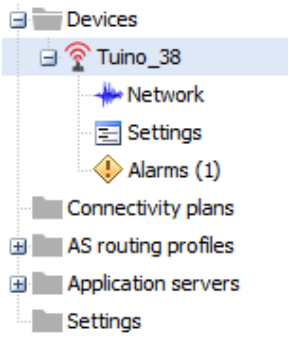
The graphic above displays the number of uplink/downlink packets and payloads (bytes) over the selected period (Daily totals, last 7 days, last 15 days).

PER/RSSI/SNR/SF graph



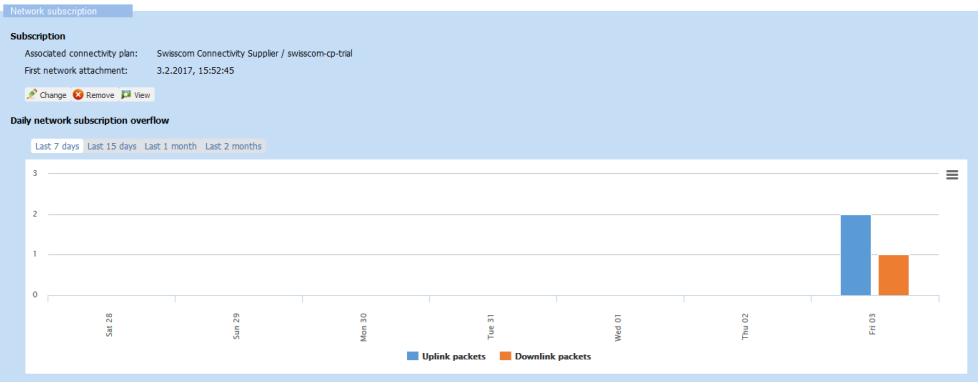
The graphic above displays the PER, RSSI, SNR or SF distribution over the selected period.

Device Network



The Devices/Network section provides information on the Network subscription, associated connectivity plan, routing plans associated to the device and the Network Coverage.

Network subscription



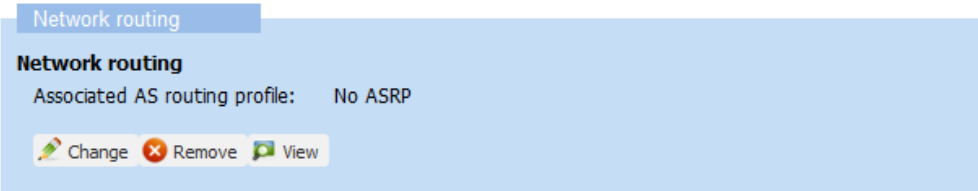
The Subscription section displays the current associated connectivity plan and the first date on which the device has communicated.

The graphic displays the usage of the device in the connectivity plan, showing the number of packets on the selected period.

HOW-TO change the Connectivity plan

> Go to the device Edit view
> Click on Change in the subscription section
> Select the desired Connectivity Plan from the drop-down menu
> Click Save

Network/cloud routing



The Network routing section displays the current associated AS routing profile. It is possible to view the details of this routing profile, or also change and remove it.

HOW-TO change or assign an AS Routing Profile

> Go to the device Edit view
> Click on Change in the Network routing section
> Select the preferred AS routing profile from the drop-down menu in the pop-up
> Click Save

3.1.3 Device provisioning

The device provisioning allows users to create devices and register them on the network through Activation By Personalization (ABP) or Over the Air Activation (OTAA).

There are two ways to create a device

- Manual creation: create devices one by one
- Batch creation: mass import from a csv file to create several devices in one go

Information required to create a new device:

ABP:

- DevEUI
- DevAddr
- NwkSKey

OTAA:

- DevEUI
- AppEUI
- AppKey

Optional information may be required:

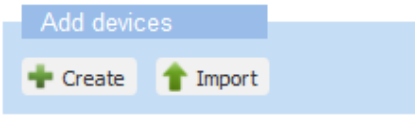
- AppSKey

ABP Manual Device creation

The manual creation is mainly for support or development purpose. Once the device is in production, refer to the Batch provisioning mode.

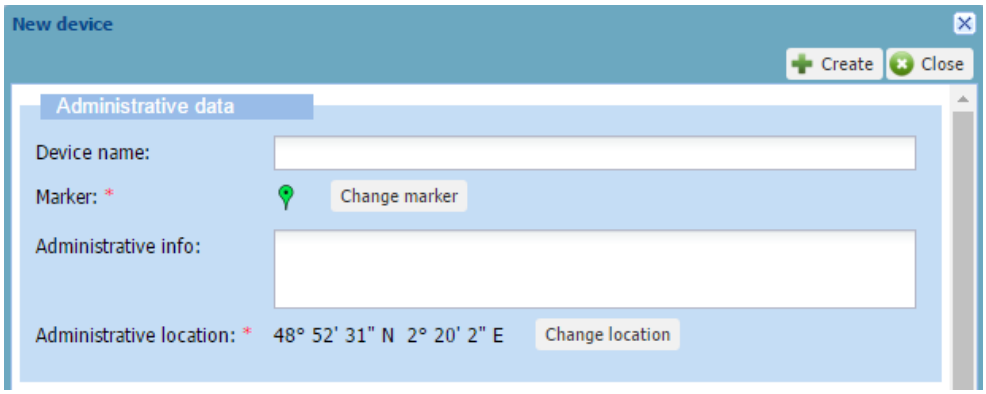
To create a new device manually:

- Click on "Create" in the Devices view



- A new pop-up appears with 4 sections: Administrative data, Device identification, Network parameters, Application layer handling

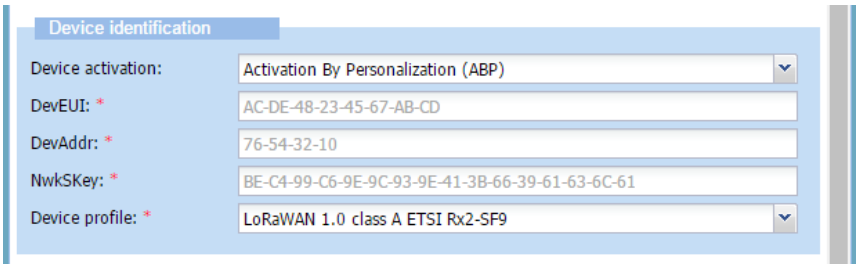
Fill out the Administrative data section



- Enter a name for the device

> Change the marker if you wish to customize the device marker
> Enter any relevant administrative information (Such information will be displayed in the e-mail generated by an alarm)
> Set a location
  - Network location
  - Manual location

Fill out the Device identification section. Please note that all fields are mandatory

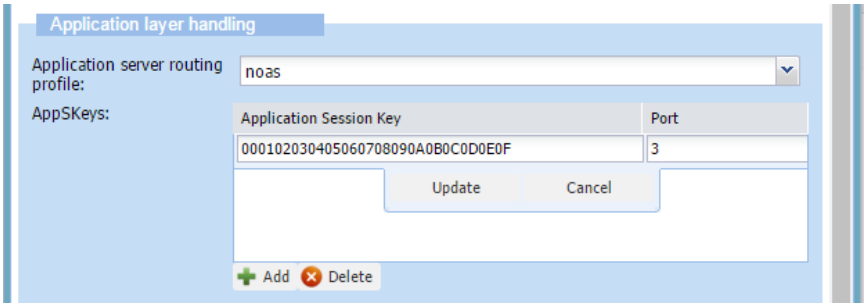


> Device EUI (hexa): DevEUI, device EUI, globally unique IEEE EUI-64 address
> Network address (hexa): DevAddr, device address
> Device profile: Bidirectional communication class
> Network key (hexa): NwkSKey, Network Session Key, 128-bit key

Fill the Network section

> Choose a Connectivity plan in the drop-down menu, the displayed count indicates the remaining connectivity plans available
> Choose an AS routing profile in the drop-down menu

Fill out the Application layer section



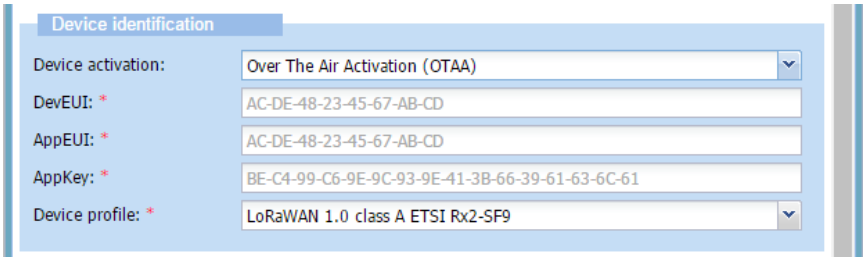
> To add a new Application Key
  - Click on Add
  - Fill the Key, 128-bit key
  - Enter a port number, " * " sends the data to all ports
  - Click on Update

The port 0 is encrypted by the NwkSKey

The optional 128-bit AppSkey is used to encrypt the payload of the messages, and has to be shared with the application server. You may decide to use a unique AppSkey for all LoRaWAN ports used by your device (* keyword), or to allocate one AppSkey for each port.

> If you do not provision the AppSKey, The Swisscom LPN Portal will forward the payload in encrypted form to the application servers and has no access to the payload clear-form content.
> If you provision the AppSkey(s), the LRC will decode the payload before forwarding it to the application server(s).

OTAA Manual Device creation



The Join device activation procedure is similar as the ABP provisioning, except the keys to inquire.

Fill out the Device identification section. Please note that all fields are mandatory

> Device EUI (hexa): devEUI, globally unique IEEE EUI-64 address
> Device profile: Bidirectional communication class
> Application EUI (hexa): AppEUI is a global application ID in IEEE EUI64 address space that uniquely identifies application provider of the end-device; The Swisscom LPN AppEUI is: F0:3D:29:AC:71:00:00:01
> Application key (hexa): is an AES-128 application key specific for the end-device that is assigned by the application owner to the end-device and is responsible to encrypt JOIN communication.

Modify a device

Modifying a device allows you to update device-related data such as the name, the manually entered location, define another connectivity plan or define a routing plan.

Start by opening a device in Edit view:

> Click on Edit to enter the Edit view

Modification allowed:

> Device name
> Administrative info
> Device location
> Device marker
> Change/remove a connectivity plan

> Change/remove a AS routing plan

Finally, to confirm the changes, go back to the device details by selecting the device in the column in the left sidebar, then click Save in the top-right corner of the screen.

Delete a device

Deleting a device is an action which cannot be undone and should be handled with care. All device details and device status information will be lost.

Below are the steps to delete a device:

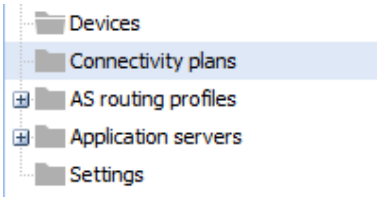
> Click on Delete to delete the device

> Confirm in the pop-up to delete the device
> Go back to the device list, click on Refresh to refresh the list

3.2. Connectivity plan

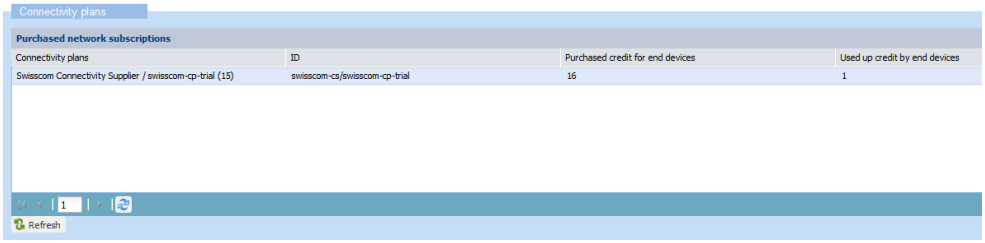
The Connectivity plan defines the network connectivity features (e.g. confirmed messages, downlink traffic), and traffic policing parameters (token bucket regulators for uplink and downlink traffic) and is associated to a given activation and recurring fee.

To access Connectivity plans, click on Connectivity plans in the left sidebar menu:



> Connectivity plan section

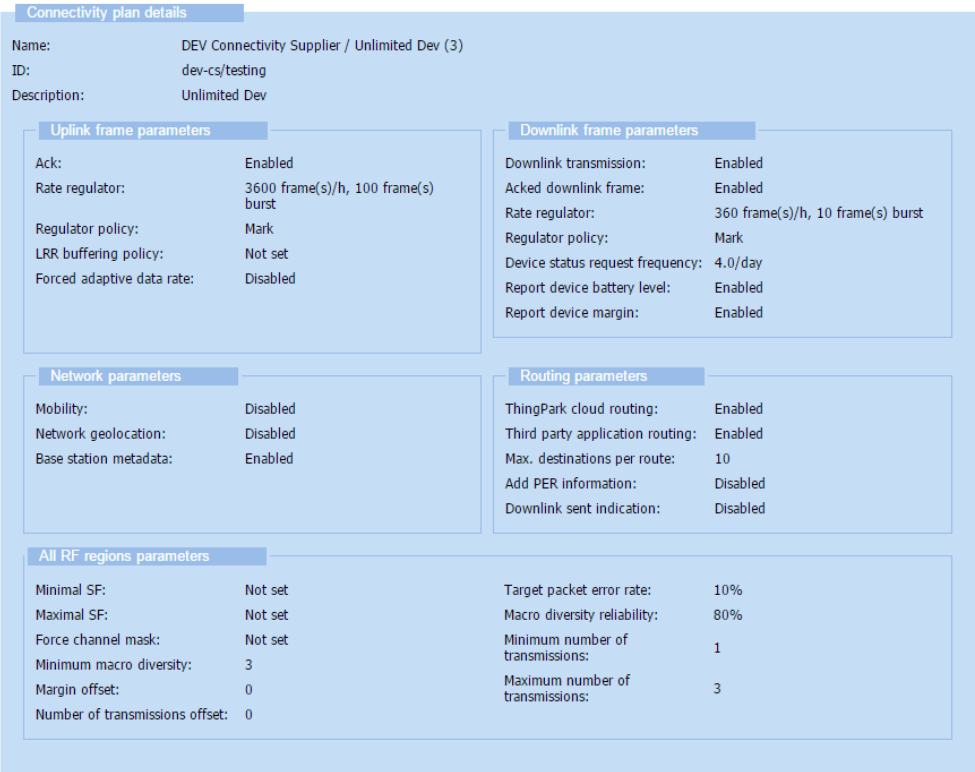
First Connectivity plan section displays the available plans in your account:



> Connectivity plans: name of the connectivity plan
> ID of the connectivity plan (required for batch provisioning)
> Purchased credit for end devices: number of maximum devices allowed in the plan
> Used up credit by end devices: number of devices registered on the selected plan

So the remaining devices that could be provisioned are “Purchased credit” – “Used up credit”.

Ask Swisscom to purchase additional Connectivity Plans.



3.2.1 Connectivity Plan details

In the Connectivity Plan details tab, the name, ID and description of the plan are displayed.

Furthermore, this tab provides a view on the following characteristics of the plan:

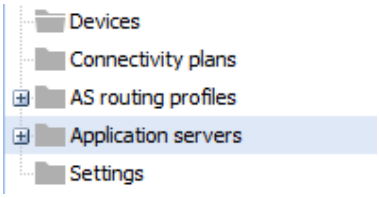
> Uplink frame parameters
  - Ack: Feature flag to allow the network to send an ACK when requested by a device operating in confirmed frame up mode.
  - Rate regulator: nb of frames allowed per hour, nb of frames allowed in burst
  - Regulator policy: uplink and downlink regulator policy
  - Mark: the system will keep track of devices exceeding the limits set
  - Drop: the system will drop packets that are beyond the limits set
  - LRR buffering policy: RFU
  - Forced adaptive data rate: Force a specific DataRate
> Downlink frame parameters
  - Downlink transmission: activation state of the downlink transmission
  - Acked downlink frame: Allow to send downlink confirmed
  - Rate regulator: number of frames allowed per hour, number of frames allowed in burst
  - Regulator policy: uplink and downlink regulator policy (see Uplink frame Regulatory policy above)
  - Device status request frequency: Number of DevStatusReq sent by the server in 24h
  - Report device battery level: Report Device battery usage level to the Device Manager application and to Third Party Application Servers
  - Report device margin: Report Device signal margin to the Device Manager application and to Third Party Application Servers

- Network parameters
  - Mobility: RFU
  - Network geolocation: RFU
  - Base station metadata: feature flag to provide LRR meta information (RSSI, SR, SNR, LRR, …) to third party application
- Routing parameters
  - Third party application routing: Feature flag to allow routing to third party servers
  - Max. destinations per route: maximum number of destinations per route
  - Add PER information: Forward PER to the Application Server
  - Downlink sent indication: Forward sent downlink indication to the Application Server
- All RF regions parameters
  - Minimal/Maximal SF: Lowest/highest spreading factor allowed for a device
  - Force channel mask: Force a specific channel mask
  - Minimum macro diversity: Minimum number of gateways which received the uplink
  - Margin offset: The offset to apply on top of the global SNR margin set at RFregion level to tune the global margin differently for different class of services to control uplink PER
  - Number of transmissions offset: Tune the number of transmissions for each uplink packet (aka redundancy) differently for different class of service to optimize uplink performance
  - Target packet error rate: "Packet Error Rate value targeted by the ADR algorithm
  - Macro diversity reliability: Minimum probability target of having N Base Stations receiving Device uplink packets
  - Minimum/Maximum number of transmissions: Minimum/Maximum number of Device uplink transmissions to ensure quality of service will not be degraded

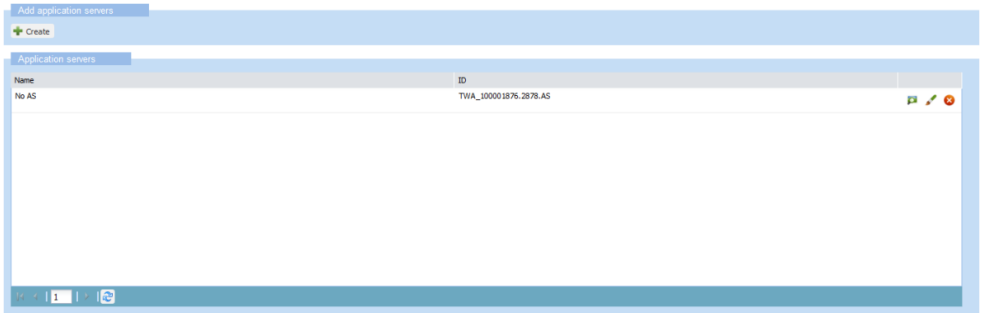
3.3. Application servers

The Application Server (AS) needs to be defined prior to adding a routing profile to it.

To access the Application Server settings, click on Application servers on the left sidebar menu:

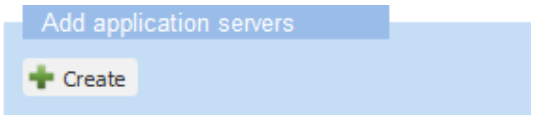


The right-hand side will then be populated with the following dialog:



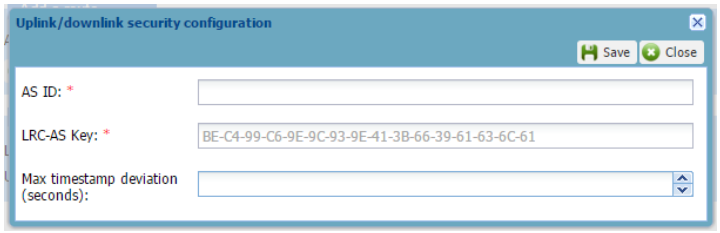
3.3.1 Create a new application server

> In the 'New application server' section, click on



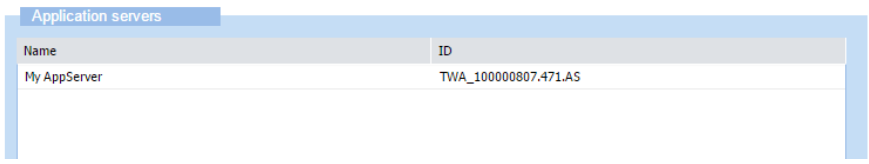
> In the name section of the new dialog, enter the name of your new Application server and click "Create":

> A new dialog appears.
> Select the type of content your Application Server can handle (xml or json) for the metadata, including a field for the payload
> If you want the security feature to be enabled, click on 'Activate' in the 'Uplink/downlink security' section



- Enter the Application Server unique ID, signature key and allowed timestamp deviation between the LRC and your Application Server. This can seriously affect communication if both sides are not properly synchronized.
- Click on 'Save' to validate the information

> Add a new route – URL or IP adress
> Click on 'Save' to save the new Application server, then close.

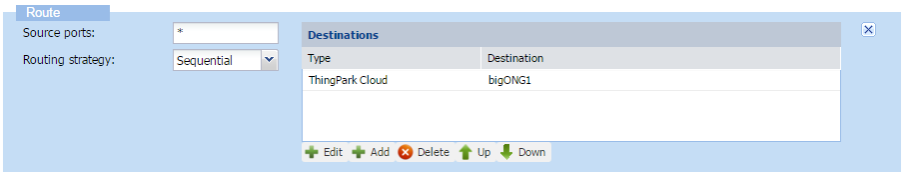


3.3.2 Edit an Application server

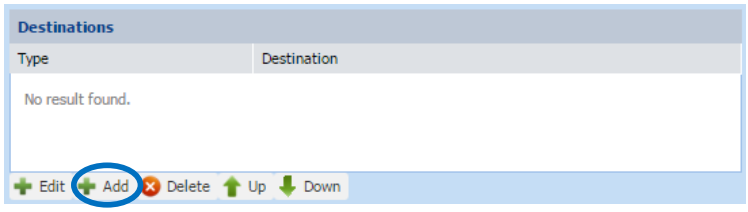
> In the list of application server, select the server you want to edit and click on "", you might have to confirm that you want to edit if not already done.

> In the 'Add a route' section, click on 'Add'

- A new layer "Route" appears



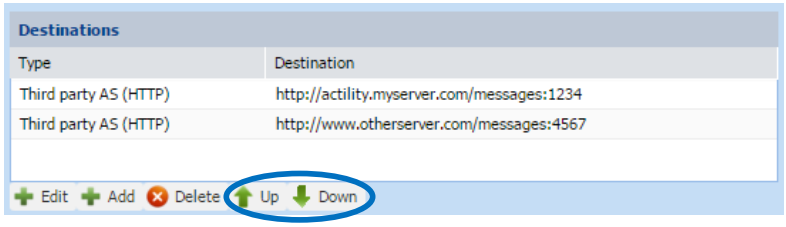
- Enter the Source ports to route
  The Source ports are the LoRaWAN ports, it could be only one port (1), a range of ports (1-4), or all ports ('*').
- Choose the Routing strategy
  In case multiple destinations are given, the routing strategy defines how the data will be sent to these destinations.
  If "Sequential" is selected, the data will be sent to the first destination and only be sent to the subsequent destination if the previous one is not available. If "Blast" is selected, the data will be sent to all destinations at the same time.
- Click on Add to create a new one



- Select the Destination of the new route

- Click on Add to add this new destination

> Sort between different destinations, for order selection if using Sequential strategy
  - You can change the order in which messages will be sent using the arrow buttons



> To edit a destination, select it and click on Edit
> To delete a destination, select it and click on Delete

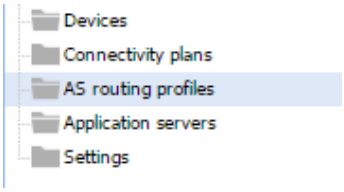
3.3.3 Delete an Application server

> In the list of application server, select the server you want to delete and click on ; then confirm that you want to delete the application server.

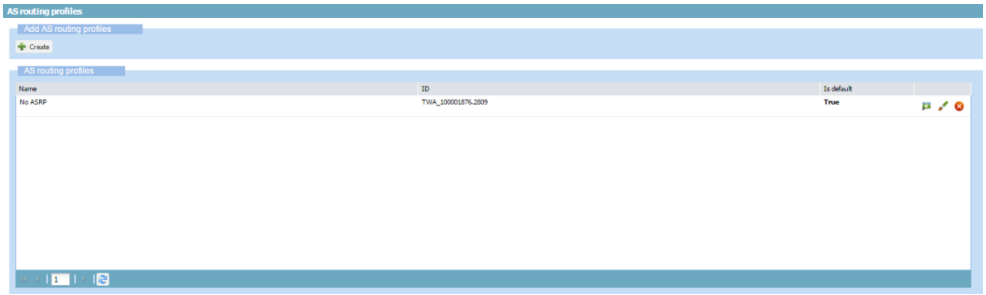
3.4. AS routing profiles

The AS (Application Server) routing profile defines how the sensor data is routed to a back-end application.

To access an AS routing profile, click on AS routing profiles on the left sidebar menu:



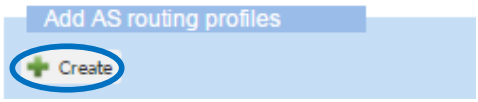
You will see the existing AS routing profiles in the list where you can View details or Edit an AS routing profile.



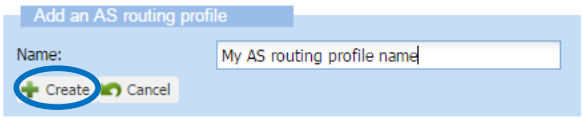
3.4.1 Create an AS Routing Profile

In order to create a new AS Routing Profile, go through the following steps:

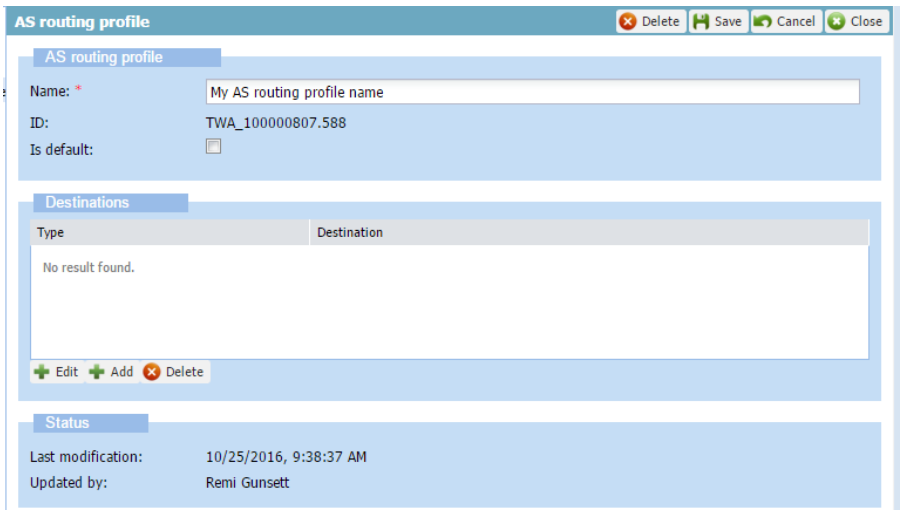
> Click on Create in the Add AS routing profiles section



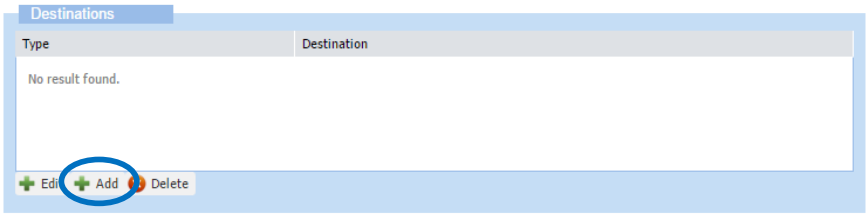
> Enter a name to the desired new AS routing profile



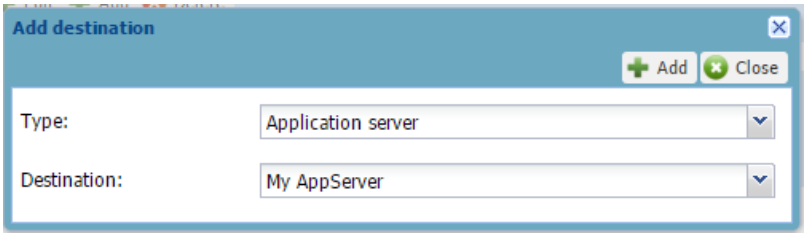
> Click on Create
> This new AS routing profile is opened and you can now edit it



> Set or unset the profile as default: check Is default checkbox
> Add a route
  - Click on Add in the "Destinations" section

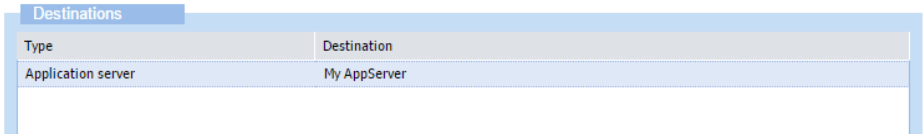


- A new pop up 'Add destination' appears



- Select the type of destination (Application server) and the destination
- Click on 'Add'

> The Application server is now in the list



3.4.2 Modify or Delete an AS Routing Profile

In order to modify an AS Routing Profile

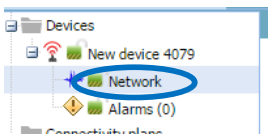
> Select the applicable Profile in the list
> Click on

Remember to click on Save after any modification made in case of edition.

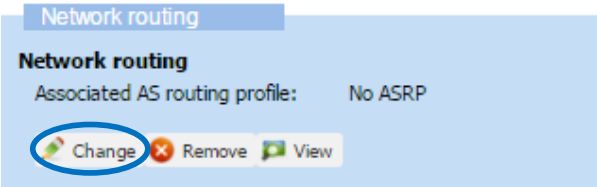
3.4.3 Assigne or Remove an AS Routing Profile

First, go to the Device Edit view in order to modify a device:

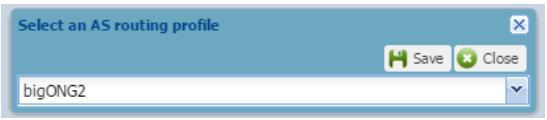
> Select a device in the list
> Click on to enter the Edit view
> Go to the Network section



> In the Network routing section
> Click on Change



> Select your new AS routing profile in the drop-down menu

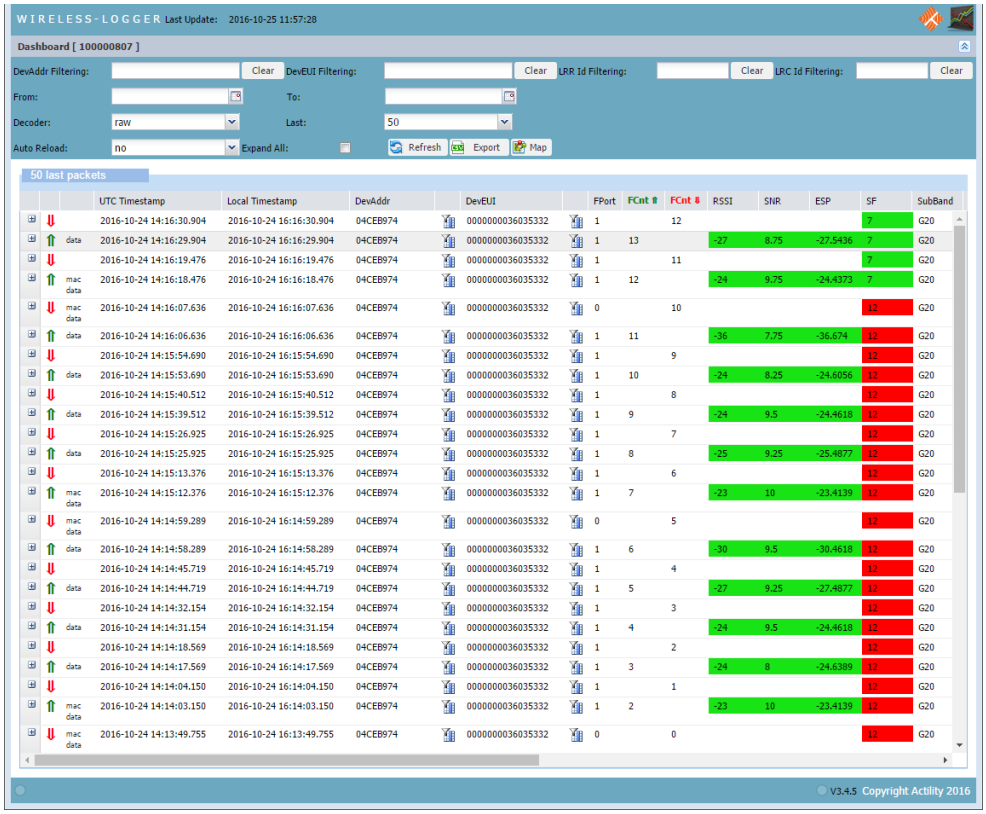


> To remove the current associated AS routing profile, click on Remove

4. Wireless Logger

> Launch the Wireless Logger from your Dashboard menu.

When you login to the Wireless Logger interface, the interface automatically displays the 50 last messages received from the devices provisioned into your Device Manager.



The interface contains a search bar and a result window displaying the messages.

4.1. Metadata

The metadata available for each message are:

> Direction of the data: up or down represented by an arrow
> Type of transmission: data, mac or simply an acknowledge
> UTC Timestamp
> Local Timestamp: UTC Timestamp translated to browser timezone
> Device Address
> Device EUI
> Port: Application port of the message
> Counter UP and Counter DOWN
> LRR RSSI: RSSI of the received message on LRR side
> LRR SNR: SNR of the received message on LRR side
> LRR ESP: ESP of the received message on LRR side
> SF: Spreading Factor

- Sub Band: LoRa sub band used for the message
- Channel: LoRa logical channel used for the message
- LRC Id: Id of the LRC server
- LRR Id: LRR with better SNR
- LRR Lat: LRR latitude
- LRR Long: LRR longitude
- LRR Count: number of LRR receiving this message. The system performs a 250ms buffering upon receiving a message to check if the same message arrives though other LRR, in which case LRR Count is incremented. If latency is uneven in the network, a message (with the same Counter Up and payload) may appear more than once in the Wlogger.
- Device Lat: Device latitude
- Device Lon: Device longitude
- LoS Distance (m): distance between the device and the LRR
- Map: displays the device and LRR on a map
- Trip: displays the location path of the device (if device location available)
- MIC: Checksum

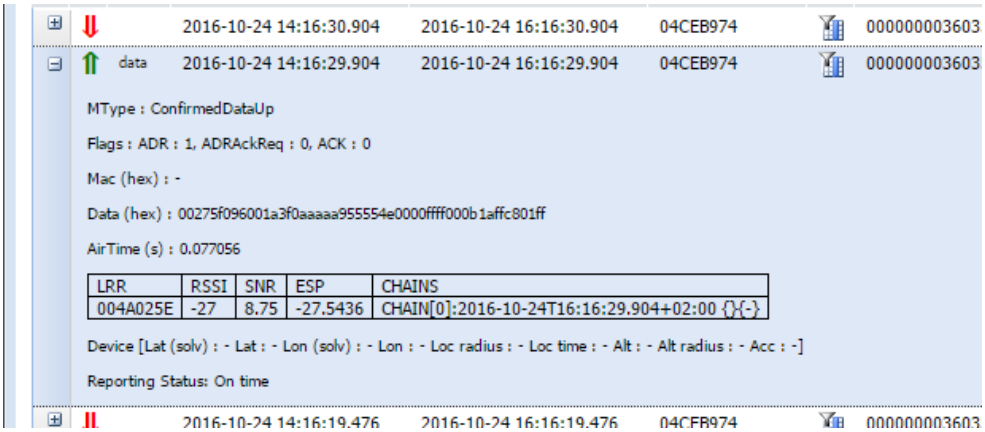
The GPS data (Device lat/lon, LoS, map and trip) are filled only if the device gives its location:

- Manual location
- Decoder selected if the location is in the payload

4.2. Payload

Click on the ⊞ icon on the left of a message in order to expand and display the message payload.

By default the message is displayed in raw data:



If the AppSKey has been entered during the Device provisioning, then the payload has been decrypted by the LRC. In any case the payload will not be displayed.

5. Application Server

5.1. Uplink interface:

Uplink destination URIs are defined for each device in service profiles (basically one per device). The main criteria is the LoRa port numbers expressed as intervals, lists or single values. A default LoRa port destination can be declared.

There are two modes of distribution which are not exclusive:

- Blast/blind mode: packets are delivered to a set of destinations without acknowledgment control.
- Sequential/Order mode: packets are delivered to a list of destinations until one of them confirms reception (200OK).

Duplicate packets (same counter up and same payload) are not sent to application servers, as long as the multiple copies are received by the network infrastructure within a maximum delay of 250ms (configurable by the LPWA operator). The XML payload sent to the application servers still include the RF metadata corresponding to all receiving base stations.

The uplink message is transmitted in a HTTP/POST request with query parameters and an XML or JSON payload.

5.1.1 Query parameters:

```
LrnDevEui: 00000000007E074F       // the device DevEUI
LrnFPort: 2                       // the LoRaWAN port number
LrnInfos: UPHTTP_LAB_LORA         // the service profile name used to route the
packet
```

5.1.2 XML payload

```
<?xml version="1.0" encoding="UTF-8"?>
<DevEUI_uplink xmlns="http://uri.actility.com/lora">
    <Time>2015-07-09T16:06:38.49+02:00</Time>        // timestamp for the packet
    <DevEUI>00000000007E074F</DevEUI>
    <FPort>2</FPort>                                 //LoRaWAN port number
    <FCntUp>11</FCntUp>                              // the uplink counter for
this packet
    <FCntDn>10</FCntDn>                              // the downlink counter for
the previous packet
    <ADRbit>1</ADRbit>
    <FCntDn>0</FCntDn>                               // the last downlink
counter to the device
    <payload_hex>0027…bd00</payload_hex>             //LoRaWAN payload in hexa
ascii format
    <mic_hex>38e7a3b9</mic_hex>                      // MIC in hexa ascii format
    <Lrcid>00000065</Lrcid>
    <LrrRSSI>-60.000000</LrrRSSI>
    <LrrSNR>9.750000</LrrSNR>
    <SpFact>7</SpFact>
    <SubBand>G1</SubBand>
    <Channel>LC2</Channel>
    <DevLrrCnt>3</DevLrrCnt>                         // number of LRRs which
received this packet
    <Lrrid>08040059</Lrrid>
    <LrrLAT>48.874931</LrrLAT>
    <LrrLON>2.333673</LrrLON>
    <Lrrs>
        <Lrr>
```

```
            <Lrrid>08040059</Lrrid>
            <Chain>0</Chain>
            <LrrRSSI>-60.000000</LrrRSSI>
            <LrrSNR>9.750000</LrrSNR>
            <LrrESP>-51.568496</LrrESP>
        </Lrr>
        <Lrr>
            <Lrrid>33d13a41</Lrrid>
            <Chain>0</Chain>
            <LrrRSSI>-73.000000</LrrRSSI>
            <LrrSNR>9.750000</LrrSNR>
            <LrrESP>-41.754934</LrrESP>
        </Lrr>
        <Lrr>
            <Lrrid>a74e48b4</Lrrid>
            <Chain>0</Chain>
            <LrrRSSI>-38.000000</LrrRSSI>
            <LrrSNR>9.250000</LrrSNR>
            <LrrESP>-46.497314</LrrESP>
        </Lrr>
    </Lrrs>
    <CustomerID>100000507</CustomerID>
    <CustomerData>…</CustomerData>                    // ascii customer data set
by provisioning
    <ModelCfg>0</ModelCfg>
    <DevAddr>007E074F</DevAddr>
</DevEUI_uplink>
```


5.1.3 JSON payload

```
{
     "DevEUI_uplink": {
               "Time": "2016-10-26T11:08:00.609+02:00",
               "DevEUI": "A81758FFFE0301EA",
               "FPort": "5",
               "FCntUp": "43749",
               "ADRbit": "1",
               "MType": "2",
               "FCntDn": "784",
               "payload_hex": "0100fd02290404b505020603c2070e0c",
               "mic_hex": "975457b1",
               "Lrcid": "00000127",
               "LrrRSSI": "-45.000000",
               "LrrSNR": "7.750000",
               "SpFact": "7",
               "SubBand": "G0",
               "Channel": "LC6",
               "DevLrrCnt": "3",
               "Lrrid": "29000119",
               "Late": "0",
               "LrrLAT": "48.875259",
               "LrrLON": "2.332962",
               "Lrrs": {
                         "Lrr": [{
                                   "Lrrid": "29000119",
                                   "Chain": "0",
                                   "LrrRSSI": "-45.000000",
                                   "LrrSNR": "7.750000",
                                   "LrrESP": "-45.673985"
                         },
                         {
                                   "Lrrid": "080E0E94",
                                   "Chain": "0",
                                   "LrrRSSI": "-51.000000",
                                   "LrrSNR": "6.750000",
                                   "LrrESP": "-51.832691"
                         },
                         {
                                   "Lrrid": "08060311",
                                   "Chain": "0",
```

```
                                        "LrrRSSI": "-59.000000",
                                        "LrrSNR": "10.250000",
                                        "LrrESP": "-59.391785"
                              }]
                    },
                    "CustomerID": "100000128",
                    "CustomerData": {
                              "alr": {
                                        "pro": "LORA/Generic",
                                        "ver": "1"
                              }
                    },
                    "ModelCfg": "0",
                    "DevAddr": "040874A4"
          }
```


5.2. Downlink interface

Depending on the device provisioning (application secret keys) encryption/decryption can be performed by the Network Server.

The following HTTP/POST message format is used to tunnel the radio frame payload and associated metadata from the target application server to the Network Server. The application server acts as a HTTP client and the Network Server acts as a HTTP server.

The downlink destination URI is the primary LRC cluster:
https://proxy1.lpn.swisscom.ch/thingpark/lrc/rest/downlink/

Such POST command may be generated easily by tools such as curl or POSTman.

```
curl -H "Content-type:application/x-www-form-urlencoded" -X POST
https://proxy1.lpn.swisscom.ch/thingpark/lrc/rest/downlink/?DevEUI=000000000F1D8
693&FPort=1&Payload=0102030405060708090A0B0C0D0E0F&FCntDn=1234
```

Query parameters:

- DevEUI (Mandatory): target device IEEE EUI64 in hexadecimal form (representing 8 octets)
- FPort (Mandatory): target port (in decimal format)
- Payload (Mandatory): hexadecimal payload. The hexadecimal payload will be encrypted by the LRC cluster if FCntDn parameter is absent, and if the LRC has been configured with an AppSKey for the specified LoRaWAN port, otherwise the Payload must be encrypted by the Application Server according to the LoRaWAN specification. The Application Server encryption uses the downlink counter, which is why the FCntDn query parameter is required in this case.
- FCntDn (Optional); LoRaWAN Downlink Counter value used to encrypt the payload. This query parameter is needed only if the Application server (not the Network Server) encrypts the payload. If present, FCntDn will be copied in the LoRaWAN header field FCnt, and the encrypted payload will be copied as-is to the LoRaWAN downlink frame by the Network Server.
- Confirmed (Optional). A value of Confirmed=0 requests transmission of an UNCONFIRMED downlink frame. A value of Confirmed=1 requests transmission of a CONFIRMED downlink frame. Default value Confirmed=0 (UNCONFIRMED).
  Support of Confirmed frame transmission is subject to Connectivity plan feature flag "ackedDownlinkFrame": if the Confirmed flag is set on the HTTP POST and the device is associated with a Connectivity plan where the "ackedDownlinkFrame" feature flag is set,

the downlink packet is processed, otherwise the processing is aborted and a specific error code is returned to the AS in the HTTP response.

LRC HTTP response codes:

- 200 “Request queued by LRC”: request accepted and queued until the class A device opens Rx slots by sending an uplink. In the case of a class C device, the downlink command will be sent as soon as the LRR base station radio is available and the maximum regulatory Tx duty cycle allows transmission.
- 350 “Invalid DevEUI”
- 350 “Downlink counter value already used. Expected=1238”: the downlink counter value was already used, for instance due to a race condition with another Application server.
- 350 “Downlink counter value increment too large. Expected=1001”: the AS supplied downlink counter value is much larger than the expected downlink counter value and was rejected by the LRC.
- 350 "Confirmed downlink is not authorized for this device": the request for transmission of a confirmed downlink packet was rejected by the LRC due to absence of “ackedDownlinkFrame” feature flag in the Connectivity plan associated to the device.

6. DX API

The purpose of the DX API is to provide the best developer experience for all developers who intend to interact with the LPN Portal for device lifecycle management.

Every call to the DX API requires three standard HTTP headers: Content-Type, Accept and Authorization. The API supports content types application/json and application/xml for requests and responses. Usage of the Authorization header is detailed in chapter 6.1.

You will find detailed documentation including examples from our platform supplier Actility on the following link.

Documentation: https://dx-api.thingpark.com/core/latest/doc/index.html

Please note that for the Swisscom DX API you will only have the rights to use the Device Operations, all other operations can be ignored.

You can also download the Swagger contract (https://dx-api.thingpark.com/core/latest/tpdx-core-api-contract.json) of the DX API, or start using the API right away with the Swagger UI (https://dx-api.thingpark.com/core/latest/swagger-ui/index.html?shortUrl=tpdx-core-api-contract.json). For other tools such as client SDK generators you can also check the Swagger homepage (http://swagger.io/).

6.1. Authentication

The DX API relies on an OAuth2 authorization workflow: in order to use the API, one must first get an API Key (also called access token), providing access to specific parts of the API, until it expires.

A new API Key can be obtained using the DX Admin API (https://dx-api.thingpark.com/admin/latest/swagger-ui/index.html?shortUrl=tpdx-admin-api-contract.json), by providing the Swisscom Thingpark-profile identifier (swisscom-api) and valid existing LPN Portal credentials. For more convenience, the Get Started page described in chapter 6.2. can also be used.

While it is valid, an API Key is associated with a scope. A scope is a group of permissions (creation, update, etc.) over a set of resources, granted by the LPN SUBSCRIBER role. As an LPN customer you will only be able to use the Device operations described under "Device operations" in the online DX API documentation.

Below is an example for such a POST command to obtain an access token:

```
curl- X POST -header 'Content-Type: application/x-www-form-urlencoded' --header
'Accept: application/json' -d 'grant_type=client_credentials&client_id=swisscom-
api%2Fvonau.n%2Bcayenne%40gmail.com&client_secret=xxxxxxxxx' 'https://dx-
api.thingpark.com/admin/latest/api/oauth/token?validityPeriod=infinite'
```

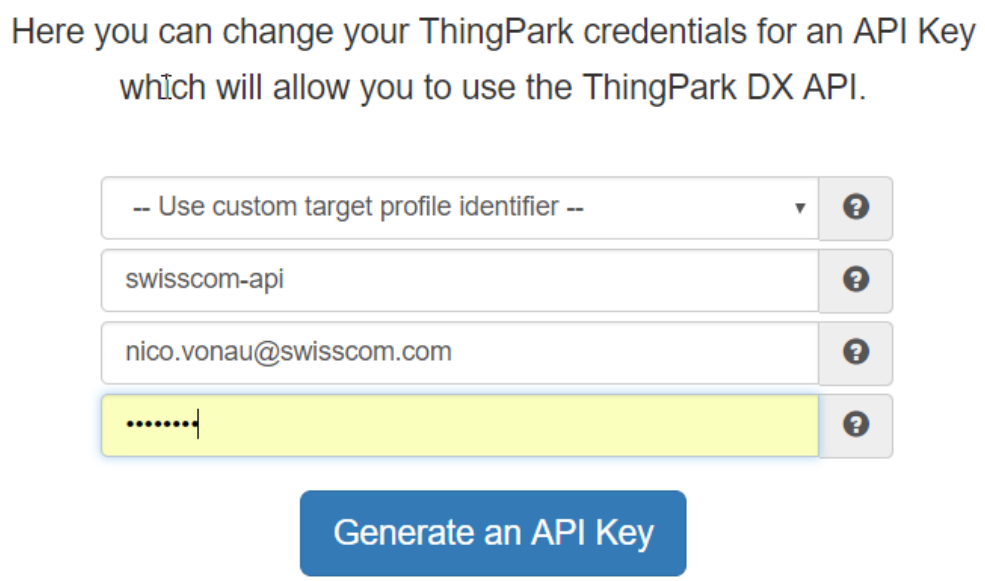
The generated API Key should be set in the Authorization HTTP header of every request as a Bearertoken, e.g.: Authorization: Bearer <access_token>.

6.2. Get Started page

With the Get Started page you will be able to generate an access token which will be valid for one week. Use the following link to acces the Get Started Page.

Get Started Page: https://dx-api.thingpark.com/getstarted/#/

To generate an access token you will have to enter the custom profile identifier (Thingpark-profile: swisscom-api) and your LPN Portal credentials as exampled below.



Swisscom (Schweiz) AG
Mobile Business Solutions
Müllerstrasse 16
CH – 8004 Zürich
www.m2m-swisscom.com