Swisscom LPN Portal

Developer Guide for CMP LPN

March 2025

User Guide

swisscom

Enable, drive and protect your business.

Notice

This document contains proprietary material of Swisscom (Schweiz) Ltd. 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.

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/Ipn

Versions

Version	Date	Details
1	31/08/16	Initial version
1.1	30/10/16	Revised initial version
1.2	31/01/17	New GUI features, new layout
1.3	04/05/17	DX API description
2	16/04/19	Merged with Application Development Guide
2.1	03/08/20	New features and uplink fields, valid with new platform release
2.2	02/10/20	Minor improvements, DX-API error codes added
2.3	25/06/21	IoT Flow Connector, Connectivity plan features and minor improvements
2.4	01/04/23	IoT Flow GUI, Tunnel API v2, Network Manager
2.5	01/08/23	MFA
2.6	06/10/23	User Management, Device Suspension
3.0	11/03/25	TPW Release 8

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/Ipn

swisscom

Table of Contents

DEFINITIONS AND ACRONYMS 5

AVAILABLE DOCUMENTS AND SCOPE 6

1. SWISSCOM LPN SYSTEM AND FEATURES 7

1.1. Swisscom LPN System 7
1.2. Swisscom LPN Features 8
1.2.1 Roaming 8
1.2.2 IoT Flow 8
1.2.3 Network Geolocation 9
1.2.4 Multicast 9
1.2.5 FUOTA 9

2. LPN PORTAL 10

2.1. Dashboard 10
2.2. Device Management and Device Creation 11
2.2.1 Device list 11
2.2.2 Device details 12
2.2.3 Delete a device 12
2.3. Device provisioning 13
2.3.1 OTAA Manual Device creation 13
2.3.2 Upload Devices with CSV file 14
2.3.3 Connectivity plan (CP) 16
2.3.4 Device suspension by removing CP 16
2.3.5 Billing of different message types 16
2.4. Connections 17
2.4.1 Create a connection 17
2.4.2 Edit an existing connection 17
2.4.3 Activate Tunnel Interface Authentication (legacy UL/DL Security) on older connections (legacy application server) 18
2.4.4 Delete a Connection 18
2.5. IoT Flow 18
2.5.1 Add connection and tags to your devices 18
2.5.2 Set up your connection (IoT Flow) 19
2.5.3 Set up your Flow 19
2.5.4 Create custom drivers 21
2.5.5 Uplink Transformation 22
2.5.6 Troubleshooting and Testing 22
2.6. Gateway Management 23
2.6.1 Dashboard 23
2.6.2 Base Station List 23
2.6.3 Base Station View 24
2.6.4 Managing Gateway Alarms 24
2.7. Wireless Logger 26
2.7.1 Metadata 26
2.7.2 Expanding a message 28
2.8. Administration 29
2.8.1 User Accounts 29
2.8.2 Multi factor authentication (MFA) setup 29
2.8.3 Service Accounts 30
2.8.4 Domains 30
2.8.5 Settings 30

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

swisscom

3. TELEMETRY INTERFACES 31

3.1. Tunnel Interface 31
3.1.1 Connectivity 32
3.1.2 Parameters Format 32
3.2. Tunnel Interface (Uplink) 34
3.2.1 Uplink Frame Report 34
3.2.2 Sample of Uplink Frame HTTP Request 36
3.2.3 LRC Authentication for UL Frame and DL Frame 37
3.2.4 Token must be verified as following by the Application Server: 38
3.2.5 XML or JSON Encoding 38
3.2.6 Sample of Uplink JSON Payload 39
3.3. Tunnel API (Downlink) 40
3.3.1 Downlink Frame 40
3.3.2 Sample of Downlink Frame HTTP Request 41
3.3.3 Confirmed Downlink 41
3.3.4 LRC HTTP Response Codes (v1): 42
3.3.5 LRC HTTP Response Codes (v2): 42
3.3.6 Multicast Downlink 44
3.3.7 Application Server Authentication for Downlink Frame 45
3.3.8 Token computation by the Application Server 45
3.4. IoT Flow 45
3.4.1 Queueing behavior 45
3.4.2 IoT Flow source IP addresses 46
3.4.3 Sample of Uplink JSON with decoded payload 46
3.4.4 Sample of Downlink JSON payload 47

4. MANAGEMENT APIS 48

4.1. OSS API 48
4.1.1 Quickstart 48
4.1.2 Authentication 48
4.1.3 Devices and connections API 49
4.1.4 Gateways API 49
4.1.5 Administration API 49
4.1.6 Error Codes 50
4.2. DX API (deprecated) 50
4.2.1 DX API Authentication 50
4.2.2 DX Admin API 51
4.2.3 DX Core API 52
4.2.4 DX API error codes 52

5. LORAWAN SPECIFICATION 53

5.1. Globally unique EUI-64: DevEUI 54
5.2. Over-The-Air Activation (OTAA) 55
5.2.1 128 bit application key: AppKey 55
5.2.2 64 bit application server identifier: JoinEUI (former AppEUI) 55
5.3. Activation By Personalization (ABP) 56
5.3.1 Device address: DevAddr 56
5.3.2 128 bit network secret: NwkSkey 56
5.4. Channel plans ETSI EU 868 57

6. DEVELOPING ON LORAWAN 58

6.1. The LPN Developer Portal 58
6.2. Swisscom IoT Qualified for LoRaWAN devices 58

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

Definitions and acronyms

ABP	Activation by Personalization
ACK	Acknowledgement of an alarm
ADR	Adaptive Data Rate
AES	Advanced Encryption Standard
AS	Application Server
CP	Connectivity Plan
ESP	Estimated Signal Power
ETSI	European Telecommunications Standards Institute
FCtrl	Frame Control
GUI	Graphical User Interface
HSM	Hardware Security Module
HTTPS	Hypertext Transfer Protocol Secure
IEC	International Electrotechnical Commission
IoT	Internet of Things
ISM	Industrial Scientific Medical
JSON	JavaScript Object Notation
LAT	Latitude
LC	Logical Channel
LON	Longitude
LoRaWAN™	Long Range Wide Area Network
LRC	Long-Range Controller: Network Server
LRR	Long-Range Relay: software inside the gateway
MAC	Media Access Control
MIC	Message Integrity Code
NW	Network
OTAA	Over-The-Air-Activation
PER	Packet Error Rate
REST	Representational State Transfer
RF	Radio Frequency
RFU	Reserved for Future Use
RSSI	Received Signal Strength Indicator
Rx	Receiver
RX1	First receive window
RX2	Second receive window
SaaS	Software As A Service
SF	Spreading Factor
SNR	Signal to Noise Ratio
Tx	Transmitter
XML	Extensible Markup Language

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

swisscom

Available Documents and Scope

Please get the newest version of this document via this link. There are also LoRaWAN specifications, gateway guides, csv templates and tutorial videos available to get you started. The Scope of this Developer Guide is to provide the guidelines to a developer during the Swisscom LPN Portal connectivity integration.

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

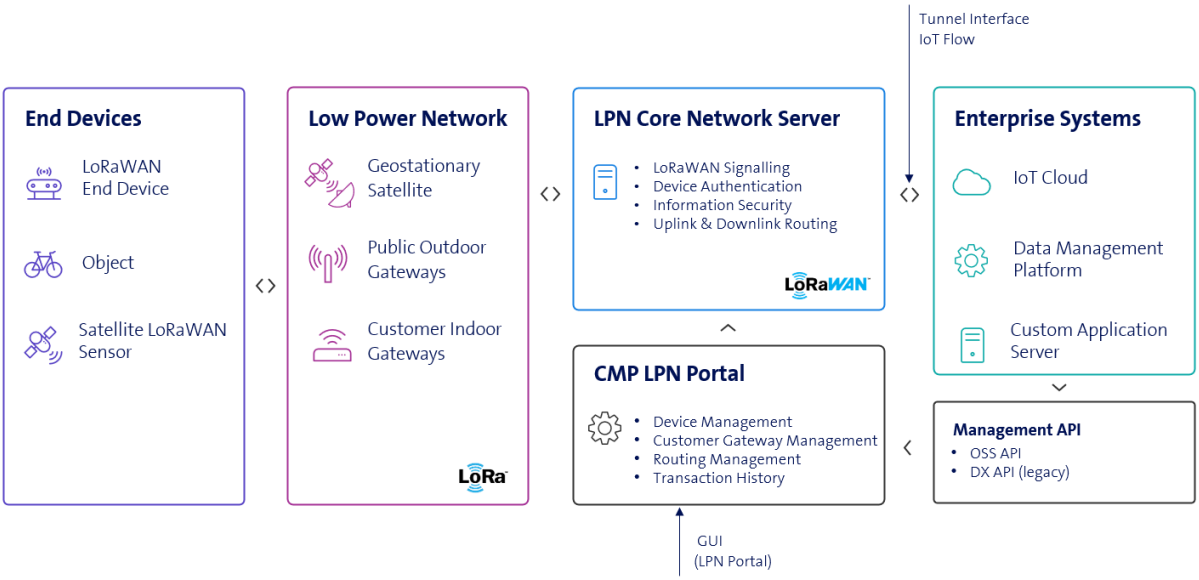
The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

1. Swisscom LPN System and Features

1.1. Swisscom LPN System

This chapter will explain what the Swisscom LPN System offers. Below is an overview of the LPN system:



Everything related to the LoRaWAN MAC layer is located in the Network Server. You will need to provision some information for each device, and you can also define where the messages will be routed. You have three options for the provisioning, that are described in the following chapters:

- Provision your devices one by one using the GUI (only recommended for tests)
  See section 2.3.1
- Generate a csv in the right format and use the csv upload function
  See section 2.3.2
- Automate your device provisioning using the OSS API
  See section 4.1

Concerning the message routing (receiving your device's messages on your end application), you currently have two options:

- Use the standard REST API connector (HTTP / HTTPS) to push each device message to your application server upon arrival at the network server. In this case the message payloads are **not** decoded and need to be decoded on the application server at your end.
  See section 0
- Use the IoT Flow connector for direct integrations into Microsoft Azure, AWS, your MQTT broker and more. **IoT Flow allows to decode the payloads** such that the application server receives an already decoded message. Decoding is accomplished with either predefined or custom JavaScript decoders.
  See section 2.5

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

1.2. Swisscom LPN Features

Features like pricing plans and pricing models are configured by choosing the according **Connectivity Plan**. Also other features like third-party routing (Microsoft Azure, MQTT, AWS, etc...), decoding, roaming and network geolocation can be activated on a device by assigning the according connectivity plan to the device. If you are missing a specific connectivity plan in your profile, please contact your sales person or IoT.SPOC@swisscom.com to get it added to your contract.

Below is a list of the default connectivity plans and the features connected to them. Depending on your contract, you might also have additional and specific connectivity plans.

	Basic	Plus	Advanced	Satellite
Switzerland-wide connectivity	✓	✓	✓	✓
Device class A, B, C	✓	✓	✓	✓ **
LoRaWAN Roaming		✓	✓	
LoRaWAN Satellite Roaming	○*	○*	○*	✓
IoT Flow	○*	✓	✓	✓
Geolocation			✓	
Multicast			✓	✓ ***
FUOTA			✓	✓ ***

○ * Only available in combination with Satellite
** Support for device class A and C
*** Available on request

1.2.1 Roaming

The use of roaming features will involve some more technical requirements on the device side:

- The use of the Swisscom JoinEUI is mandatory
- Roaming involves some more MAC-layer features that might be unused when running the device only in Switzerland. Make sure that your device is Swisscom IoT Qualified before using it in roaming, see section 6.2.

1.2.2 IoT Flow

With IoT Flow it is possible to **decode messages** and route them directly **to third party services** like Microsoft Azure, Amazon Web Services or to your MQTT broker. You will need a connectivity plan that includes the IoT Flow feature (for example *Plus* or *Advanced*). For more details on how to configure IoT Flow, refer to section 2.5.

Currently, there is a wast selection of connectors available, for example:

- HTTPS (REST API)
- Microsoft Azure IoT Hub / Event Hubs
- Amazon Web Services (AWS) IoT Core
- MQTT over SSL, WSS or TCP (you need your own MQTT broker)

1.2.3 Network Geolocation

The Swisscom LPN network allows you to localize your LoRaWAN device, based on the signal strength (RSSI) and the time difference on arrival (TDOA) of the message at each gateway. The solver is already integrated in our network server infrastructure. What will be provided to the application server, are the approximate coordinates of your device, together with some precision estimations.

Network geolocation is included in the **Advanced** connectivity plan. The approximate precision of the localization can vary between 200m (urban areas) and a few kilometers (rural areas). There is no requirement on the hardware side, any LoRaWAN device sending uplinks can be localized. However, this feature will lead to the end device consuming more energy as the messages are generally sent with higher SF. Please follow the separate network localization user guide for more details.

1.2.4 Multicast

Multicasting allows to create groups of class B oder class C devices to then send a downlink to all of them at the same time. This feature is very useful for some specific use cases, please contact Support.LPN@swisscom.com to get onboarded and use the feature. To use it on Swisscom LPN the devices nees a connectivity plan **Advanced**.

1.2.5 FUOTA

Swisscom LPN supports Firmware Update Over-The-Air according to LoRa Alliance Specification. Please contact Support.LPN@swisscom.com for more information if it is relevant for your use case.

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

2. LPN Portal

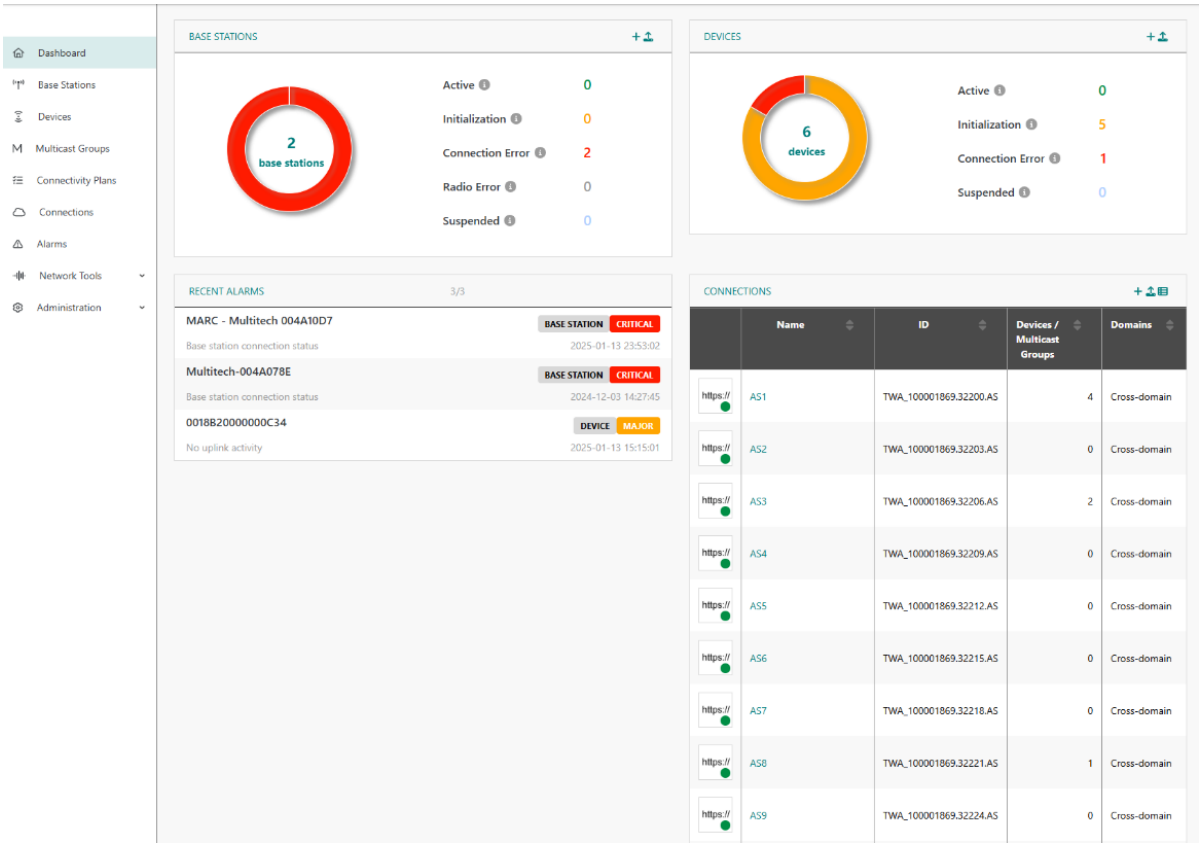
The LPN Portal is the first option to provision and manage your devices. All functions available in the LPN Portal are also available via the OSS API.

› Access the LPN Portal via the following link:
https://portal.lpn.swisscom.ch

2.1. Dashboard

When you login for the first time, you need to setup a 2 factor authentication. If you struggle with the setup, please refer to the instructions in chapter 2.8.2.

Once you login to the CMP LPN portal, you will get a dashboard displaying an overview of the devices, gateways, connections and alarms in your account. The interface is based on two frames, a navigation sidebar on the left and a main application frame showing device data. The left sidebar menu gives access to devices, gateways (Base Stations), connections, alarms, Wireless Logger and administration.



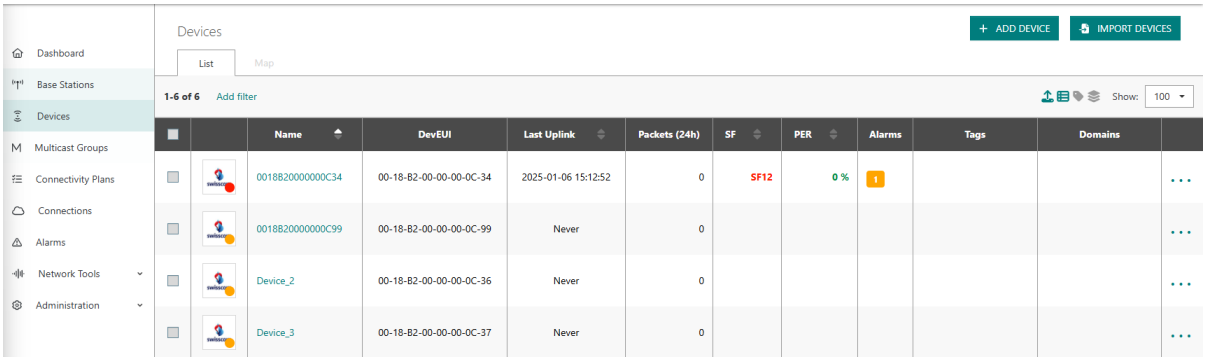
The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

2.2. Device Management and Device Creation

2.2.1 Device list

To access the devices list go to *Devices* in the sidebar on the left.



The list is customisable via this option on the top right corner of the list. This allows you to select the columns to be displayed, such as:

- Name
- Model
- DevEUI
- Device Address: DevAddr of the device
- Connectivity plan
- Connection
- Last Uplink
- Last Downlink
- Packets (24h)
- SNR: Signal to Noise Ratio
- SF: Spreading Factor
- PER: Packet Error Rate
- ESP: Estimated Signal Power
- Best Gateway
- Creation date
- Alarms
- Tags
- Domains

The filtering can be done on some of the above mentioned properties using the *Add filter* function at the top of the list. Users can therefore easily display all devices that for example use a specific connectivity plan, contain a substring in its name (if you are using a naming convention), have raised critical alarms, etc.

Use the function to export the device list in csv format and use to three dots on the right to delete a device or send a downlink.

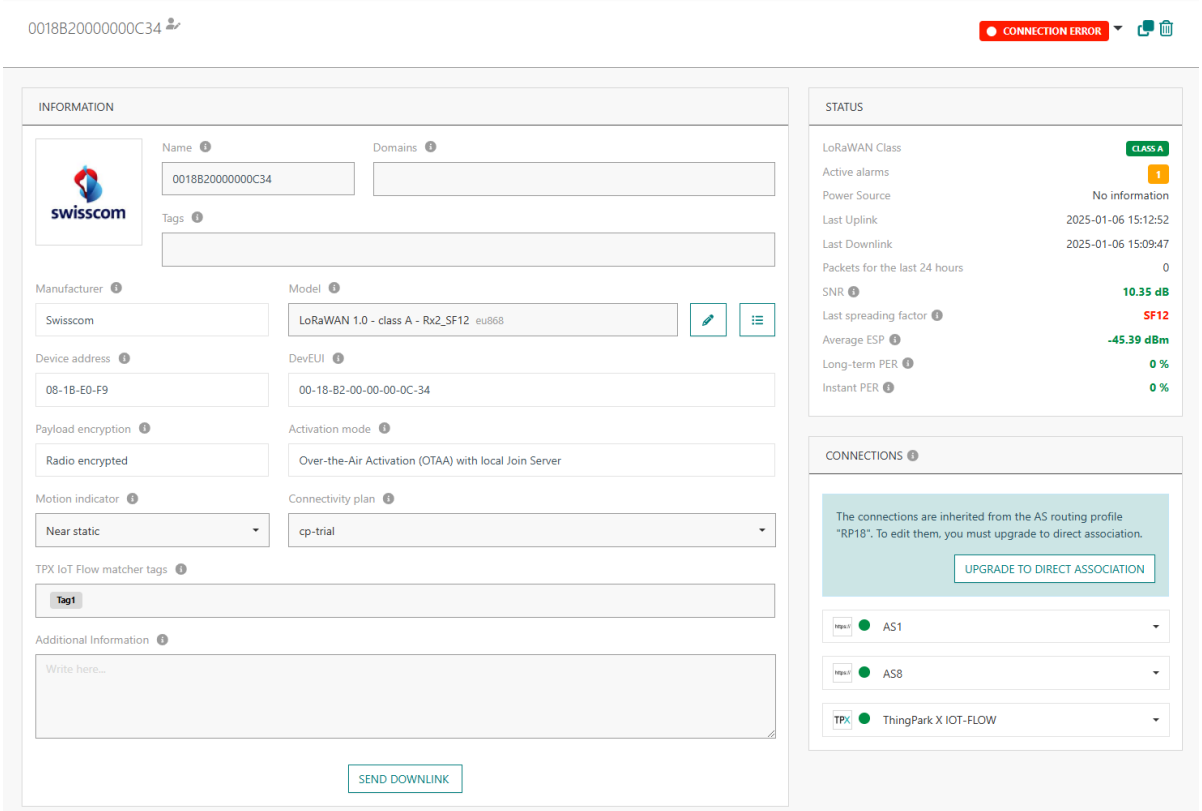
The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/Ipn

2.2.2 Device details

This frame displays the basic information of the device such as the model (device profile), name, DevEUI, DevAddr and its associated connectivity plan and connection.

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.



The location of the device on the map could be provided in 2 different ways:

- Manual location: The position of the device is manually set in the device's configuration by the user
- Network location: Uses the location solver of the platform

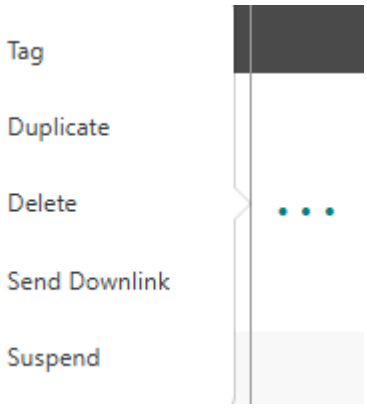
All the way on the bottom, alams can be setup in case the device is not sending uplinks anymore. Chose the condition and severity according to your needs. A lot of fields can be edited by a user with according rights.

2.2.3 Delete a device

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

Below are the steps to delete a device:

- Click on the three dots to open the menu
- Confirm in the pop-up to delete the device



Note: Device detetion is irreversible. To get a deleted device to communicate with the network server again, it has to be reprovisioned and rejoined to the network, i.e., a rejoin procedure has to be triggered

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

2.3. Device provisioning

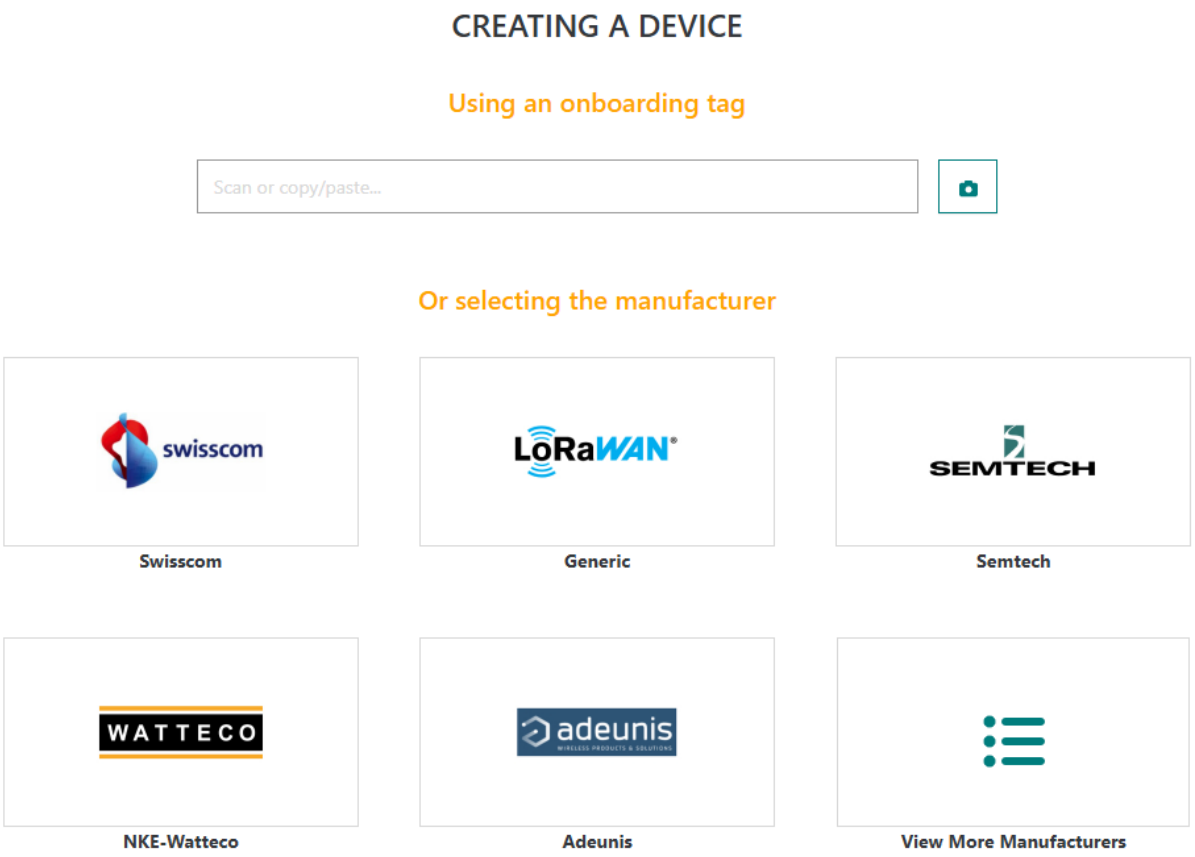
The device provisioning allows users to create devices and register them on the network through **Over the Air Activation (OTAA)**. There are three ways to create a device:

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

2.3.1 OTAA Manual Device creation

Chose the manufacturer of your device. If its not present in the available selction, select Swisscom.

Fill out the Device identification section. Please note that all fields with an asterisk (*) are mandatory

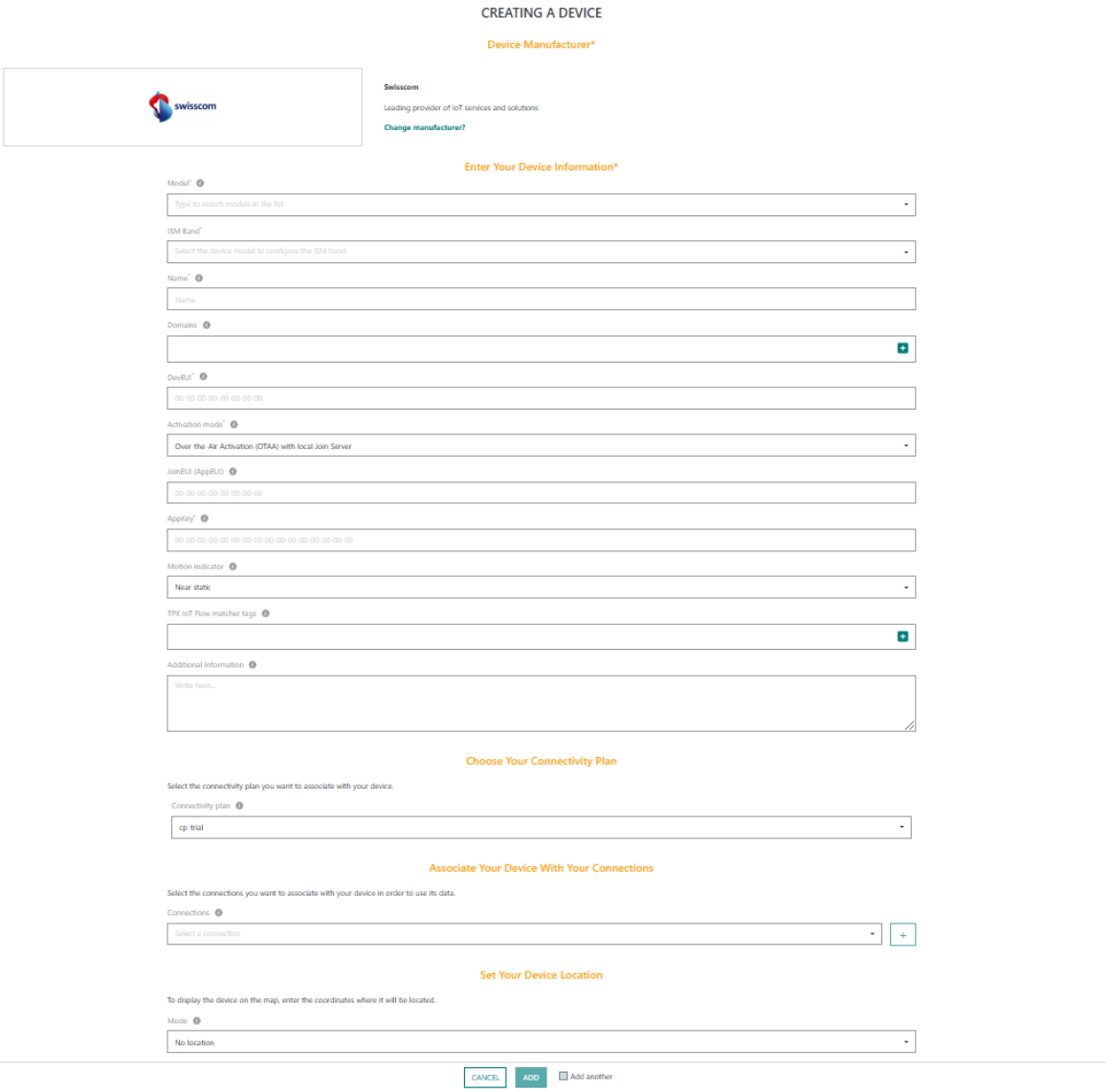


- Model: Choose the correct model for your device (Class A/B/C, LoRaWAN version). If you are unsure about this, ask the device manufacturer for the LoRaWAN version of your device. This selection impacts the driver behaviour on IoT Flow, when selecting "automatic driver" on IoT Flow. Therefore, make sure the model is correct.
- ISM Band: EU868
- DevEUI (hex): Device EUI, globally unique IEEE EUI-64 address
- JoinEUI (hex): JoinEUI/ApplicationEUI is a global application ID in IEEE EUI64 address space that uniquely identifies network provider of the end-device; the Swisscom LPN JoinEUI is: F0:3D:29:AC:71:00:00:01
- AppKey (hex): ApplicationKey is an AES-128 key specific for the end-device that is assigned by the application owner to the end-device and is responsible to encrypt JOIN communication.

The information in this document does not constitute a binding offer. It is subject to revision at any time.

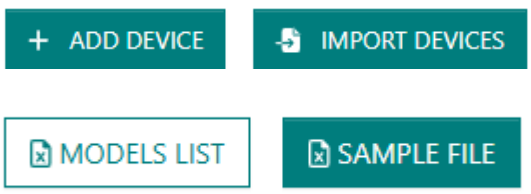
Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/Ipn

- Motion Indicator: Choose the behaviour that matches to your device the most. The motion indicator is used to enhance the network geolocation accuracy and impacts the choice of the gateway used to send a downlink to the device.
- TPX IoT Flow matcher tags: If you are using IoT Flow, make sure to tag your device accordingly here.
- Connectivity Plan: Chose the connectivity plan to be assigned to your device.
- Connections: Chose the connection to be assigned to your device
- Key format: enter your key unencrypted



2.3.2 Upload Devices with CSV file

The creation of multiple devices can either be done in an automated way using the OSS-API, or by batch upload with a csv file. Please us the available examples on the Portal to create the input file for your devices.

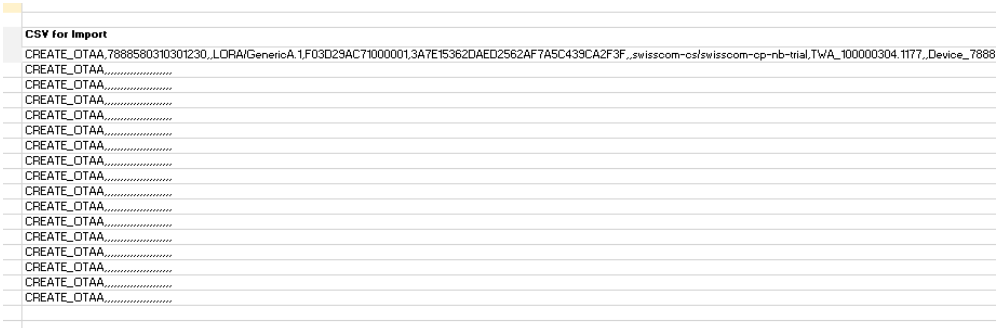


The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/Ipn

To help you generate your import file, we provide a CSV generator file. It is available here.

Task (Mandatory)	'CREATE_OTAA': OTAA device creation
DevEUI (Mandatory)	Globally unique IEEE EUI-64 address in hexadecimal format.
DevADDR	leave empty
Device profile ID (Mandatory)	Choose one of the following IDs corresponding to the device model you have, browse the MODELS LIST for the exact model.
AppEUI (Mandatory)	F03D29AC71000001 for Swisscom
AppKey (mandatory)	AES-128 application key specific for the device that is assigned by the application owner to the device
ThingPark X IoT Flow matcher tags	Tag to be used on Flows
Connectivity plan ID	ID of the connectivity plan
AS routing profile ID (optional)	To be found in your device manager, under AS routing profiles. e.g. TWA_100000304.1177 Default will be taken if none specified
Device name (optional)	Free text


CSV for Import
CREATE_OTAA,7888580310301230,,LORA/GenericA.1,F03D29AC71000001,3A7E15362DAED2562AF7A5C439CA2F3F,,swisscom-cs/swisscom-cp-nb-trial,TWA_100000304.1177,,Device_7888
CREATE_OTAA,,,

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

2.3.3 Connectivity plan (CP)

The Connectivity plan defines the network connectivity features (e.g. confirmed messages, downlink traffic), and traffic policing parameters and is associated to a given activation and recurring fee. To access Connectivity plans, click on Connectivity plans in the left sidebar menu. The *List* overview displays the available connectivity plans in your account:

- Name: name of the connectivity plan
- ID: ID of the connectivity plan (required for batch provisioning)
- Purchased credit for end devices: number of maximum devices allowed in the plan
- Remaining licences: number of remaining credits to be used on the selected plan

Additional connectivity plans can be purchased from your Swisscom sales contact.

Naming examples:

Price plan (contract)	Connectivity plan name in the platform
Basic Low	swisscom-cs/swisscom-cp-basic-low
Advanced High	swisscom-cs/swisscom-cp-advanced-high
Trial	swisscom-cs/swisscom-cp-trial

2.3.4 Device suspension by removing CP

Devices can be disabled by removing the Connectivity Plan. The device is then unprovisioned and will not be billed any more from the following month on. From release 7.3 on, suspended devices do not need to perform a new join procedure. When a device is suspended, the network automatically sends the DutyCycleReq MAC command to the device to configure its transmission duty cycle to the lowest value. This allows preserving the battery of the suspended devices.

Use this feature with caution! To work successfully, DutyCycleReq and DutyCycleAns needs to be correctly implemented in the LoRaWAN device. Otherwise, this feature will not work and might put the device in an unknown state from which the device cannot be recovered without a reboot or rejoin.

2.3.5 Billing of different message types

The Pay-as-ou-go model bills for the number of uplink and downlink messages. Here is an overview of the messages that will be billed (✓) and not be billed (✖).

Billed as uplink message	
Uplink payload messages other than MAC (FPort > 0)	✓
Join-requests	✓
MAC request: LinkCheckReq	✓
MAC answer: DevStatusAns	✓
MAC request: DeviceTimeReq	✓
ACK after confirmed downlink	✓
All other MAC traffic (ACK bit not set and FPort=0)	✖

Billed as downlink message	
Downlink payload messages other than MAC (FPort > 0)	✓
Join-accepts	✓
MAC answer: LinkCheckAns	✓
MAC request: DevStatusReq	✓
MAC answer: DeviceTimeAns	✓
ACK after confirmed uplink	✓
All other MAC traffic (ACK bit not set and FPort=0)	✖

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

2.4. Connections

Connections define where the data is routed to. **Data is never stored** on the LPN portal. Redundancy may be added to handle a failover to a second connection with the sequential strategy explained later.To access the Connections settings, click on *Connections* on the left sidebar menu.

2.4.1 Create a connection

- Click on + ADD CONNECTION in the top right corner.
- In order to create a new connection the following information needs to be provided:
  - Connection Information
  - Name
  - Domain
  - Additional Information
  - Configure tunnel interface authentication
  - Key: new connections cannot be created without a key. This is a security feature that ensures that only the listed endpoints below can send Downlinks to devices using this connection. In earlier versions this was not mandatory.
  - Maximum Timestamp Deviation
  - Configure the destinations
  - Multiple routes can be defined per connection. Each route has a strategy and URLs assigned. The sequential strategy will send the uplink to the next destination if there has not been an answer from the first endpoint after 5 seconds.
  - Configure the HTTP requests
  - Custom HTTP Header
  - Content Type
- When all information are entered click on add at the bottom.

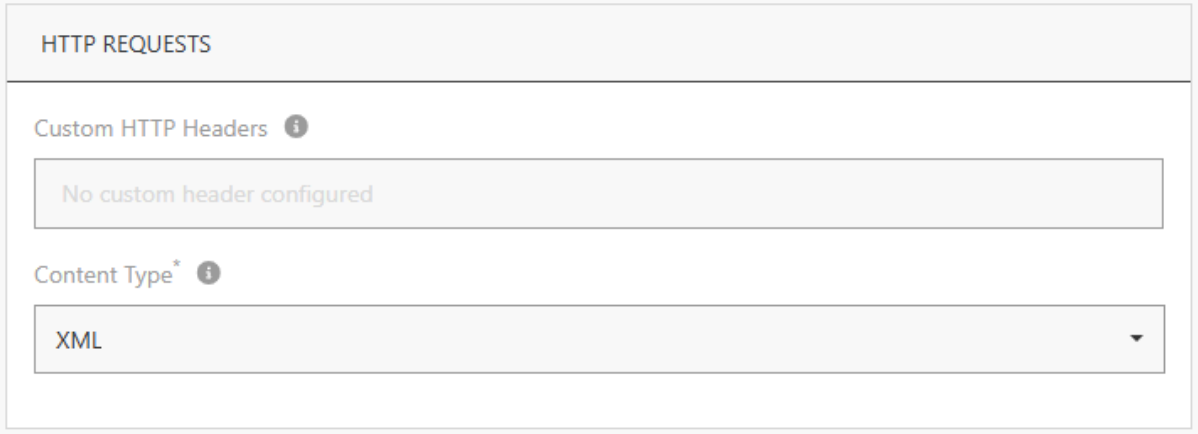
2.4.2 Edit an existing connection

In the connections list, click on the connection name that you want to edit.

Note: Basic authentication is supported. Please use the following format in the destination URL:

```
https://user:password@www.example.com/
```

Or use the field HTTP custom headers above. Note that you need to do the base64 encoding yourself if you use the HTTP custom header field.



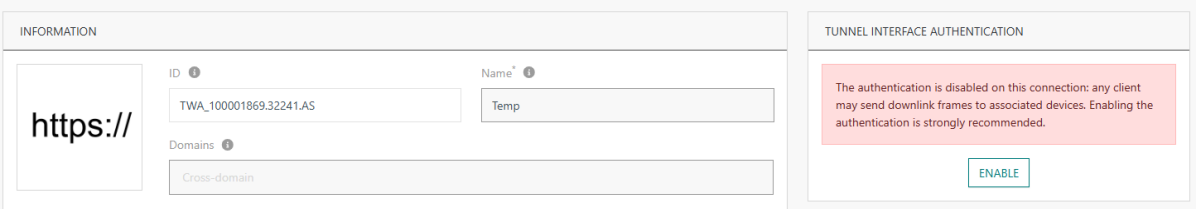
The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

2.4.3 Activate Tunnel Interface Authentication (legacy UL/DL Security) on older connections (legacy application server)

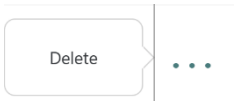
It is **highly recommended** to set a key for older connections to ensure that only your listed destinations are allowed to send downlinks to your devices.

First, click on the Connection you want to activate the authentication. At Tunnel Interface Authentication, click enable, copy the randomly generated key and set the maximum timestamp deviation which shall be accepted by the LRC.



2.4.4 Delete a Connection

In the list of connections, click on the three dots of the connection you want to delete and click on delete; then confirm that you want to delete the connection.



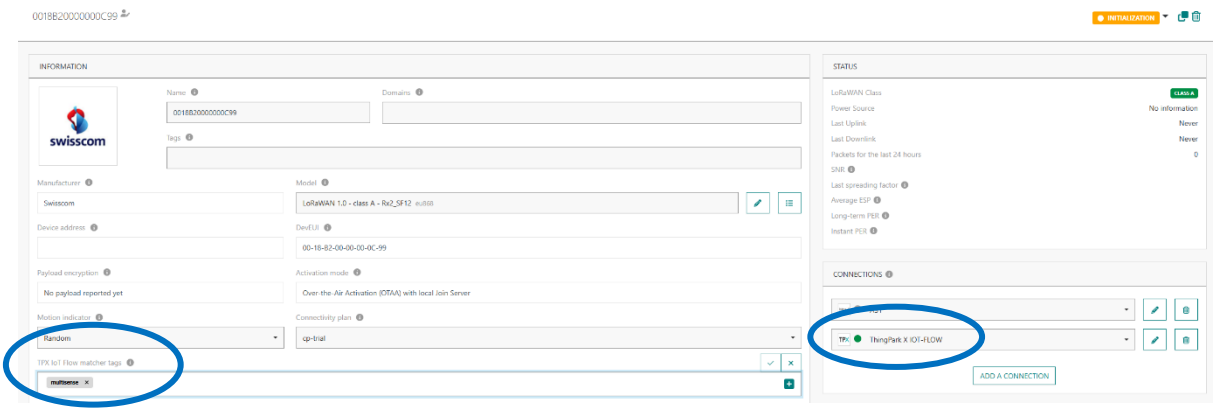
2.5. IoT Flow

It is possible to route payloads directly to third party services like Microsoft Azure, Amazon Web Services or to your MQTT broker, etc.. IoT Flow also offers a vast selection of decoders, such that the messages received at the endpoint are already decoded.

2.5.1 Add connections and tags to your devices

In the LPN Portal, add the ThingPark X IOT-FLOW connection to your device. If you want to route all of your traffic to the same place, you don't need to follow this step. However, if you have different destinations for your devices or if you want to decode payloads of different manufacturers, you can use tags on your devices.

Go to your device in edit mode to add your tag to the field TPX IoT Flow matcher tags.



The information in this document does not constitute a binding offer. It is subject to revision at any time.

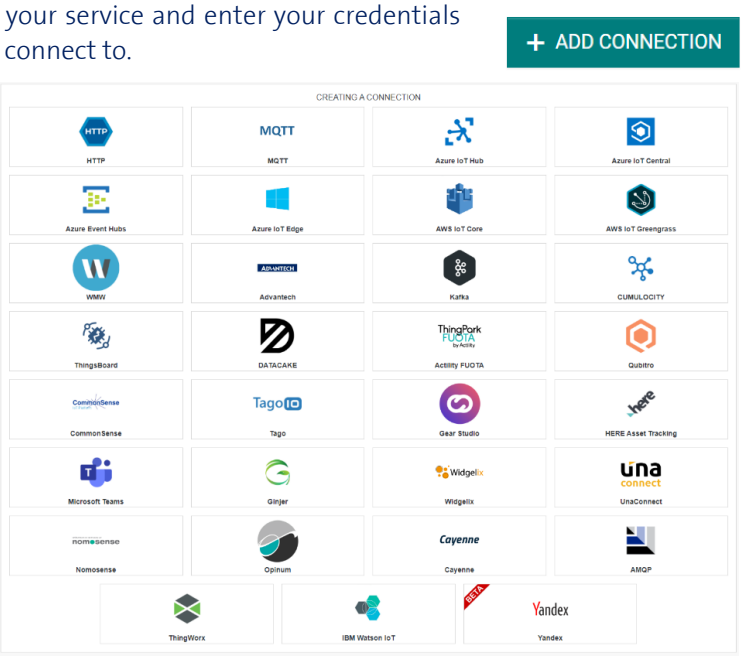
Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

2.5.2 Set up your connection (IoT Flow)

After you have routed your traffic to the IoT Flow service, please log in to the IoT Flow GUI at:

› https://portal.lpn.swisscom.ch/tpx/login

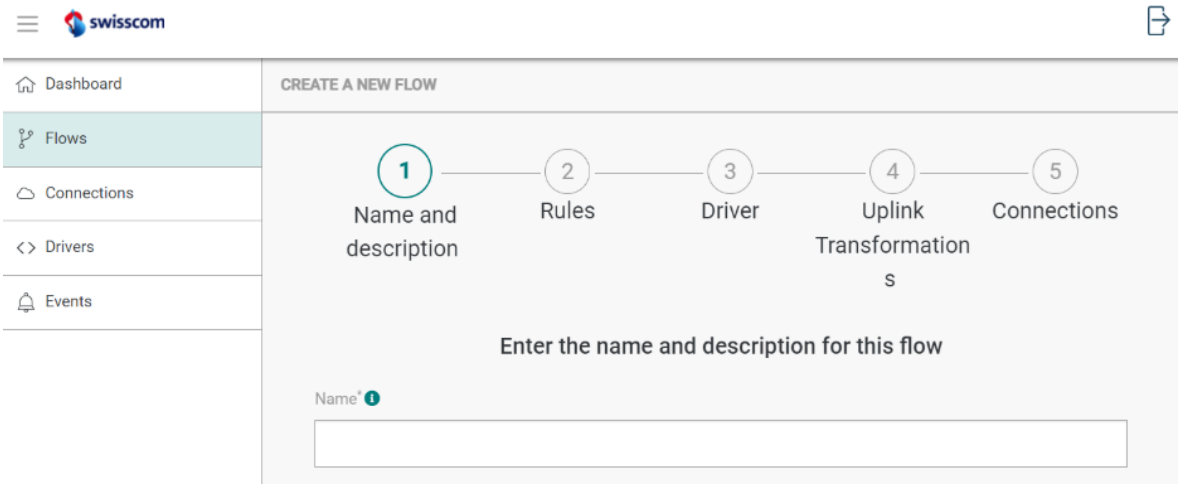
Click on "Add Connection", choose your service and enter your credentials depending on your cloud service to connect to.



Warning: Do not use connectors marked "BETA" in production! These are only a preview to be used for tests. Changes to the configuration and connector restarts are reserved without notification!

2.5.3 Set up your Flow

Click on "Flow" on the left hand menu and "Create new Flow".



You will be guided through the following steps:

› **Name and description:** Give your flow a name

- **Rules:** Choose "Keys" to match this flow on specific DevEUIs. Choose "Tags" in order to use the tags which you have set up previously in section 0. Please mind that only devices **that match all of the tags in the group** will be routed to this flow. You can create "OR" rules by adding another tags group.
  Note: Tags are optional, you can set none and all the traffic will be routed to this flow.

- **Driver:** Choose "Force driver", if your devices have a generic or swisscom deviceprofile and choose your device's driver from the list. Chose "automatic" if your devices have the device profile from the manufacturer to use the decoder on the platform. You can create your own driver following section 2.5.4.
  Note: Message decoding on IoT Flow is optional. If you prefer not to activate any driver, choose "Automatic". The "Swisscom" profiles used on our platform do not have any drivers associated to them. Only Swisscom Multisense will be decoded with configuration on "automatic".
- **Uplink Transformations:** This step is not mandatory
- **Connections:** Choose at least one connection where you would like your data to be sent.

After having set both a connection and a flow, IoT Flow is ready to route your data. You can monitor the state of your connections in the "Connections" menu. You will see a list of open and closed connections, as well as the number of routed messages.

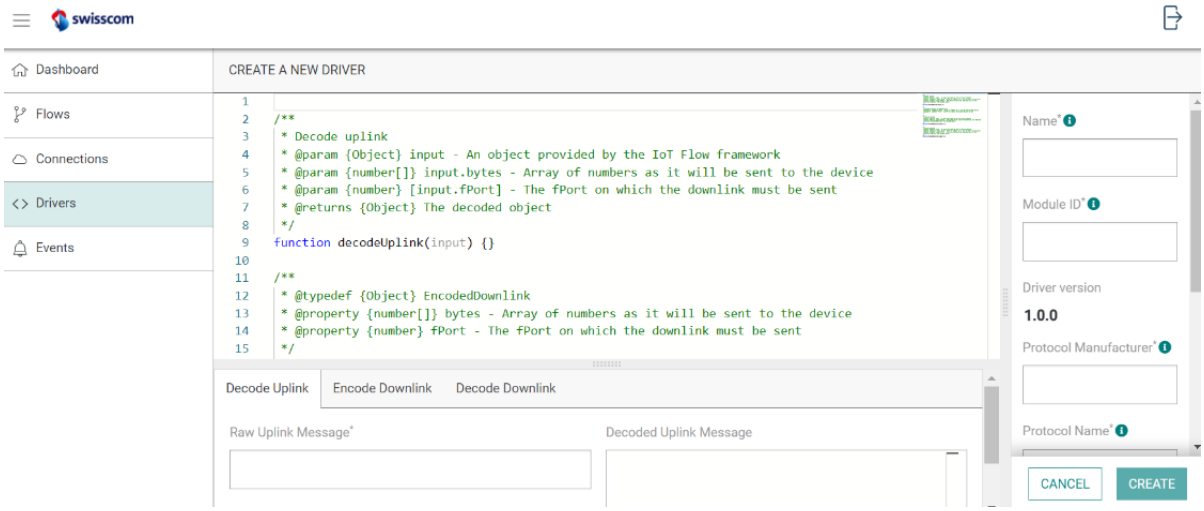
Last Restart	Active Devices (last 1h/24h)	Uplinks (last 1h/24h)	Downlinks (last 1h/24h)	State
5 days ago	1 / 1	1 / 24	0 / 0	OPENED
–	0 / 0	0 / 0	0 / 0	CLOSED

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/Ipn

2.5.4 Create custom drivers

In the "Drivers" menu point, you can create and test your own JavaScript driver. A custom driver will be created, and is only visible inside your organization. Please follow the Github links and further documentation available in the portal. Let us know if you want to make your driver public and we will assist you. This enables other users to use the same driver.



The information in this document does not constitute a binding offer. It is subject to revision at any time.

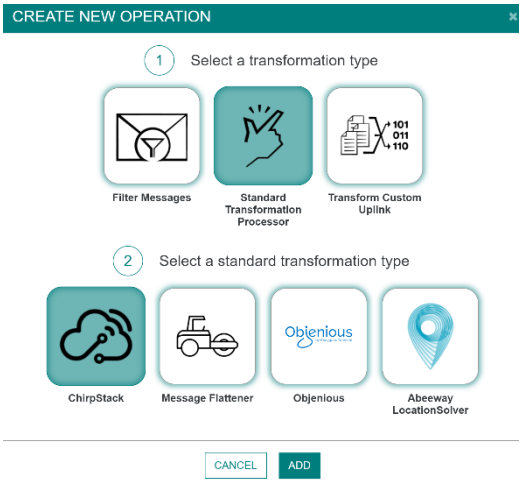
Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/Ipn

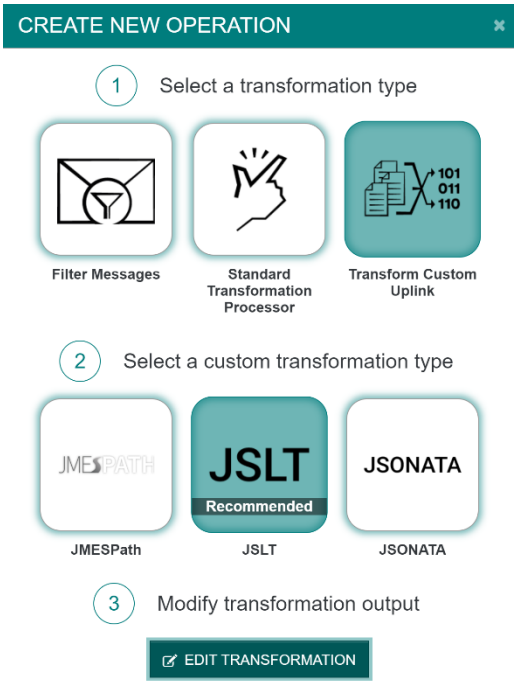
The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

2.5.5 Uplink Transformation

If you prefer to use a custom JSON format rather than the standard format, you can use of the uplink transformation. This allows to easily transform the uplink such that it can be directly integrated to for example Chirpstack. The uplink transformation is done on the connection. You can also define a custom uplink via JSLT. On "edit transformation" changes to the JSON are defined for each tag in the JSON.

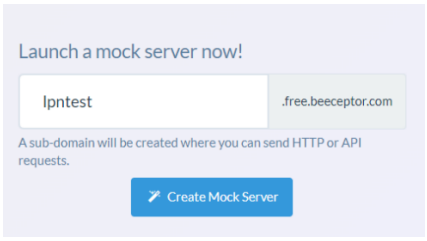


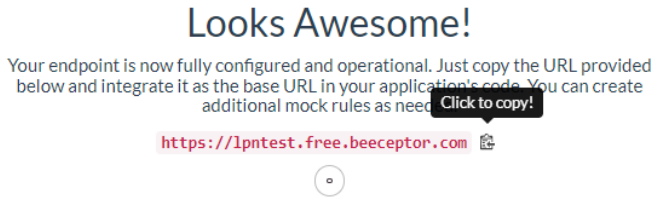


2.5.6 Troubleshooting and Testing

We encourage you to use https://beeceptor.com/ for troubleshooting and testing. At Beeceptor you can create endpoints for testing and add this endpoint as connection on the LPN Platform. The device messages are then directly visible on Beeceptor.

- Enter a name and click on Create Mock Server
- Copy the endpoint URL and save it in destination of the connection as described before in section 2.4.2.
- Beeceptor can be used for the standard connections and for the HTTP connector on IoT Flow.

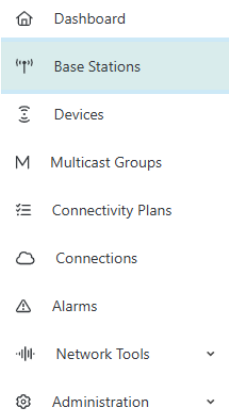




The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

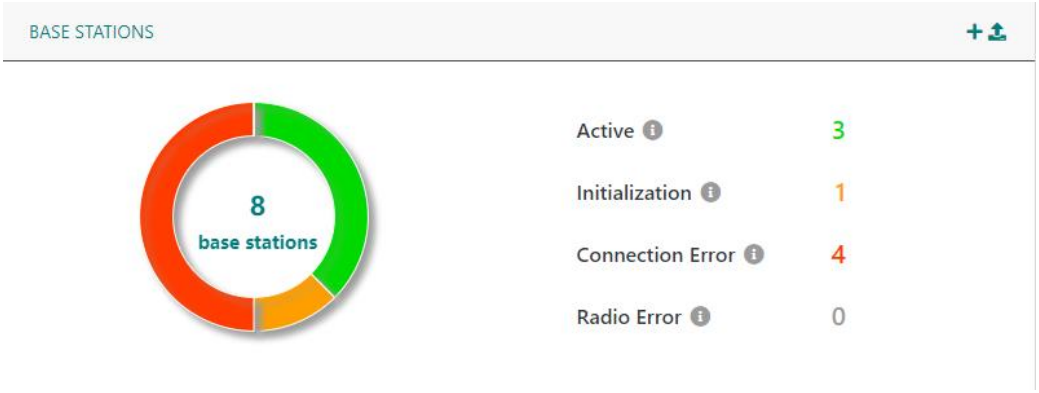
2.6. Gateway Management



To access your gateways go to the option Base Stations in the sidebar on the left. It shows a complete list of Swisscom LoRaWAN indoor and outdoor gateways assigned to your account, i.e., the gateways you bought from Swisscom. Additionally, this site gives access to metadata such as name, LRR-ID, connection status, traffic history, served devices, etc. of all your gateways.

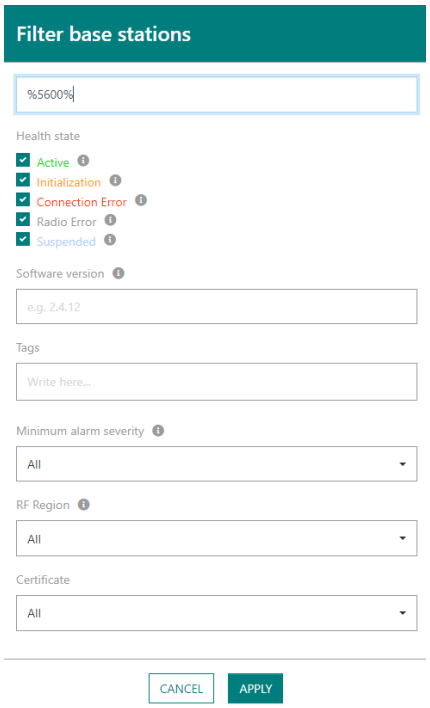
2.6.1 Dashboard

The dashboard gives a compact status overview of the gateways. It displays how many gateways are connected or initializing and how many gateways are offline, experiencing connection or radio errors and it shows the ten most recent alarms ordered by descending severity.



2.6.2 Base Station List

The Base Station List provides a list of all gateways of your account. It can be sorted by name or last uplink and filtered across various aspects such as name, LRR ID, health state, software version, tags or alarm state. Clicking on a gateway in this list opens the gateway view where a lot of information regarding the selected gateway is visible. The page shows name, manufacturer, model, LRR ID of the gateway and its status information. The "%" sign can be used in the search field to filter with partial entries.

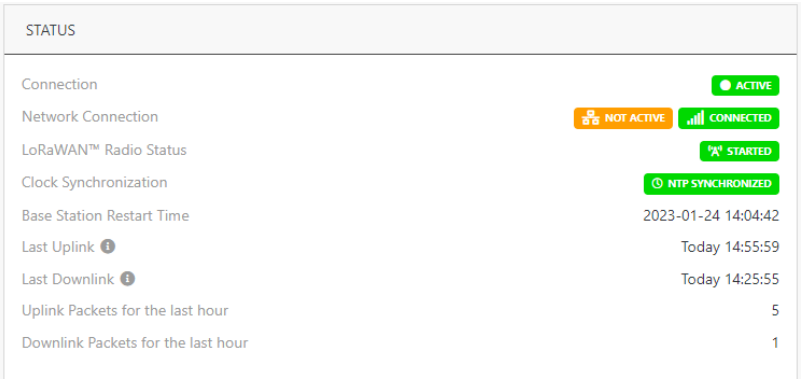


The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

2.6.3 Base Station View

Clicking on a gateway in the Base Station List leads to the gateway view, where the status of a gateway is easily visible. The status overview shows if the gateway is connected and online, if its connected via mobile or ethernet, if the LoRaWAN software is running, the last uplink/downlink timestamp and the amount of uplinks/downlinks in the last 24h. Furthermore, the served devices of a given gateway can be seen on this page.



2.6.4 Managing Gateway Alarms

To help you identify and prioritize corrective actions when required, the LPN Platform triggers alarms regarding a gateway when a gateway is not working properly or a gateway is exposed to a replay attack threat. When a gateway is not working properly, the following alarms are triggered automatically: beacon transmission failure, abnormal log activity, base station connection status, downlink frame rate exceeds the RF cell capacity.

An alarm is a visible signal used to indicate that a gateway malfunction, a process deviation or an abnormal condition requires a response. The alarm management system qualifies the alarm and assigns a specific state to it. The alams states are associated with colors and relate to the following severity levels:

Alarm state	**Color**	**Definition**
Critical	Red	The service is affected and an immediate corrective action is required
Major	Orange	The service is partly affected and an urgent action is required
Minor	Yellow	A fault that does not affect the service should be corrected to prevent a more serious problem
Warning	Blue	A potential or impending fault affecting the service should be diagnosed and corrected if necessary
Indeterminate	Purple	The severity cannot be determined
Cleared	Green	The alarm has satisfied the clearing condition and has been cleared by the system. Note that the current status of a cleared alarm is available in the active alarms panel and the last status before clearance of the alarm is available in the alarms history panel during 15 days after clearance

The **most important gateway alarms** to monitor are the **following**:

- LRR Restart 101: The gateway was restartet because no uplink frames were received to a specified time window
- Connection Status 102: The gateway is not connected to the Network Server
- Time Sync lost 109: The gateway has lost its time synchronization
- GPS lock failure detected 122: The gateway cannot use GPS position and time synchronization

- Downlink frame rate exceeds the RF cell capacity 105: This alarm is raised when some downlink packets were dropped by the gateway because the RF cell modem was busy. The downlinks were received from the gateway but not sent over-the-air. For a class A device the trasmission window is then outdated.

The **following gateway alarms** can be **misleading and shall be ignored**:

- Unusually high RAM usage level 107: The RAM usage of the base station reaches an abnormal level.
- Unusually high file system usage level 108: The file system usage of the base station reaches an abnormal level.

To setup alarms please refer to section 2.8.5.

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

The information in this document does not constitute a binding offer. It is subject to revision at any time.

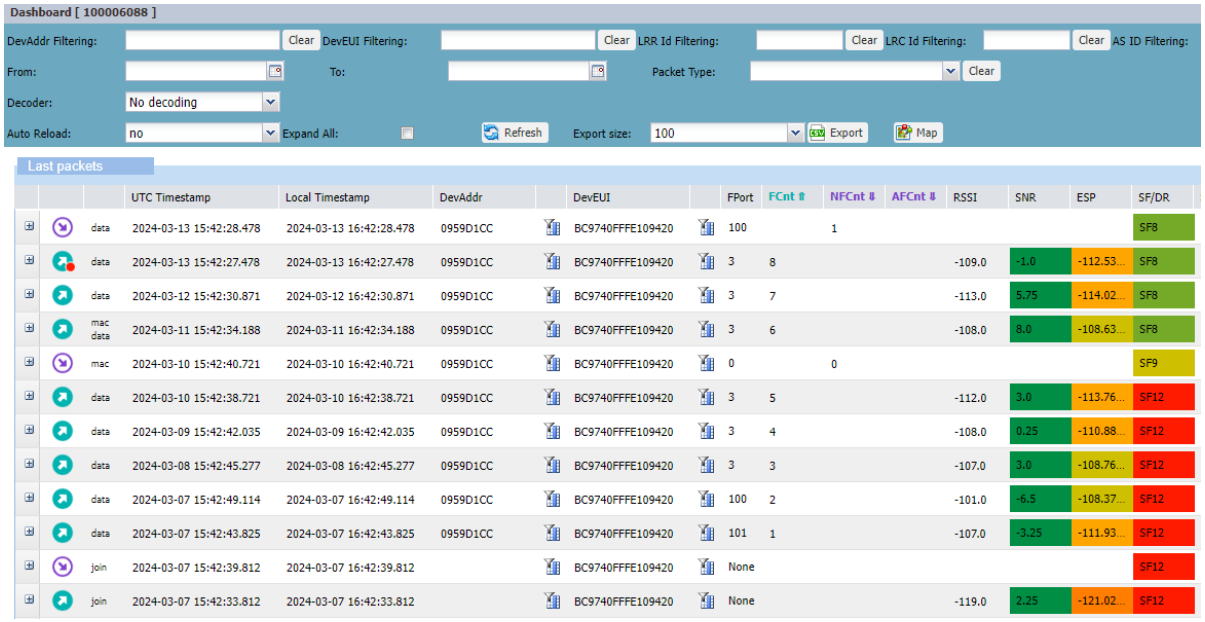
Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

2.7. Wireless Logger

The Wireless Logger is a convenient debugging tool to check the uplink and downlink messages sequence of each device. It will show the complete list of LoRaWAN transactions from your account with all its metadata (Timestamp, Signal strength ESP, Gateway IDs, MAC commands etc...) However, it will show neither decoded, nor raw payload. This is sent to your application server only, and not stored on the Swisscom LPN portal.

- Launch the Wireless Logger from the sidebar menu *Network Tools* → *Wireless Logger*

Upon accessing the Wireless Logger, the interface automatically displays the 50 last messages received from the devices provisioned into your Account. There are many options to filter for DevEUI, Gateway ID, Timestamp, etc. at the top.



		UTC Timestamp	Local Timestamp	DevAddr	DevEUI	FPort	FCnt	NFCnt	AFCnt	RSSI	SNR	ESP	SF/DR
data		2024-03-13 15:42:28.478	2024-03-13 16:42:28.478	0959D1CC	BC9740FFFE109420	100		1					SF8
data		2024-03-13 15:42:27.478	2024-03-13 16:42:27.478	0959D1CC	BC9740FFFE109420	3	8			-109.0	-1.0	-112.53...	SF8
data		2024-03-12 15:42:30.871	2024-03-12 16:42:30.871	0959D1CC	BC9740FFFE109420	3	7			-113.0	5.75	-114.02...	SF8
mac data		2024-03-11 15:42:34.188	2024-03-11 16:42:34.188	0959D1CC	BC9740FFFE109420	3	6			-108.0	8.0	-108.63...	SF8
mac		2024-03-10 15:42:40.721	2024-03-10 16:42:40.721	0959D1CC	BC9740FFFE109420	0		0					SF9
data		2024-03-10 15:42:38.721	2024-03-10 16:42:38.721	0959D1CC	BC9740FFFE109420	3	5			-112.0	3.0	-113.76...	SF12
data		2024-03-09 15:42:42.035	2024-03-09 16:42:42.035	0959D1CC	BC9740FFFE109420	3	4			-108.0	0.25	-110.88...	SF12
data		2024-03-08 15:42:45.277	2024-03-08 16:42:45.277	0959D1CC	BC9740FFFE109420	3	3			-107.0	3.0	-108.76...	SF12
data		2024-03-07 15:42:49.114	2024-03-07 16:42:49.114	0959D1CC	BC9740FFFE109420	100	2			-101.0	-6.5	-108.37...	SF12
data		2024-03-07 15:42:43.825	2024-03-07 16:42:43.825	0959D1CC	BC9740FFFE109420	101	1			-107.0	-3.25	-111.93...	SF12
join		2024-03-07 15:42:39.812	2024-03-07 16:42:39.812		BC9740FFFE109420	None							SF12
join		2024-03-07 15:42:33.812	2024-03-07 16:42:33.812		BC9740FFFE109420	None				-119.0	2.25	-121.02...	SF12

2.7.1 Metadata

The metadata available for each message are:

- Direction of the data: up or down represented by a directional green or red 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: LoRaWAN sub band used for the message
- Channel: LoRaWAN 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 (if device location available)
- Device Lon: Device longitude (if device location available)
- 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 information in this document does not constitute a binding offer. It is subject to revision at any time.

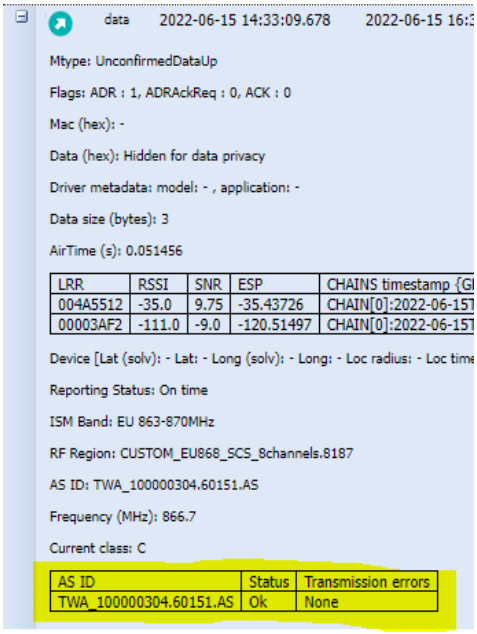
Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

2.7.2 Expanding a message

Click on the ⊞ icon on the left of a message in order to expand and display the message details. You will see the flags, packet size, the list of gatways which have received your signal and Application server delivery information. It is displayed if a message arrived at the Application server in the table at the bottom of every uplink frame. In case an error occurred, it will be visible as a red dot and more information will be displayed in this field.



data 2022-06-15 14:33:09.678 2022-06-15 16:3

Mtype: UnconfirmedDataUp

Flags: ADR : 1, ADRAckReq : 0, ACK : 0

Mac (hex): -

Data (hex): Hidden for data privacy

Driver metadata: model: - , application: -

Data size (bytes): 3

AirTime (s): 0.051456

LRR	RSSI	SNR	ESP	CHAINS timestamp {G
004A5512	-35.0	9.75	-35.43726	CHAIN[0]:2022-06-15T
00003AF2	-111.0	-9.0	-120.51497	CHAIN[0]:2022-06-15T

Device [Lat (solv): - Lat: - Long (solv): - Long: - Loc radius: - Loc time

Reporting Status: On time

ISM Band: EU 863-870MHz

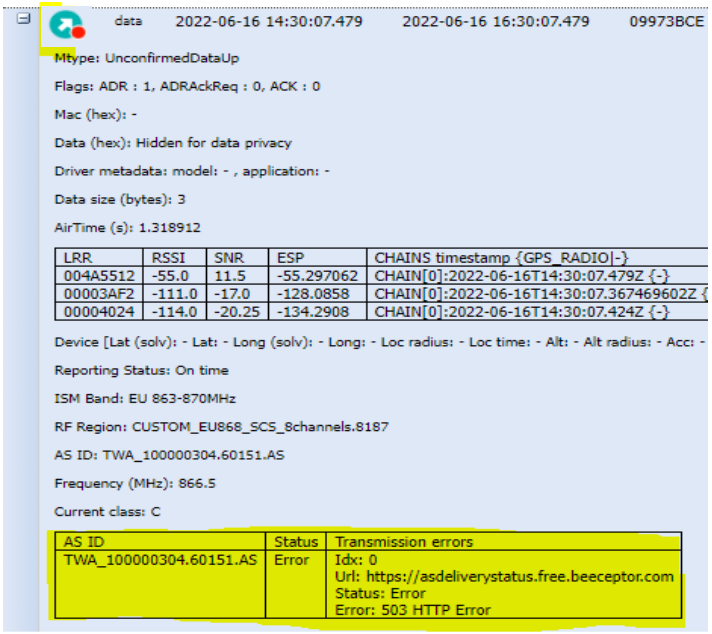
RF Region: CUSTOM_EU868_SCS_8channels.8187

AS ID: TWA_100000304.60151.AS

Frequency (MHz): 866.7

Current class: C

AS ID	Status	Transmission errors
TWA_100000304.60151.AS	Ok	None



data 2022-06-16 14:30:07.479 2022-06-16 16:30:07.479 09973BCE

Mtype: UnconfirmedDataUp

Flags: ADR : 1, ADRAckReq : 0, ACK : 0

Mac (hex): -

Data (hex): Hidden for data privacy

Driver metadata: model: - , application: -

Data size (bytes): 3

AirTime (s): 1.318912

LRR	RSSI	SNR	ESP	CHAINS timestamp {GPS_RADIO|-}
004A5512	-55.0	11.5	-55.297062	CHAIN[0]:2022-06-16T14:30:07.479Z {-}
00003AF2	-111.0	-17.0	-128.0858	CHAIN[0]:2022-06-16T14:30:07.367469602Z {
00004024	-114.0	-20.25	-134.2908	CHAIN[0]:2022-06-16T14:30:07.424Z {-}

Device [Lat (solv): - Lat: - Long (solv): - Long: - Loc radius: - Loc time: - Alt: - Alt radius: - Acc: -

Reporting Status: On time

ISM Band: EU 863-870MHz

RF Region: CUSTOM_EU868_SCS_8channels.8187

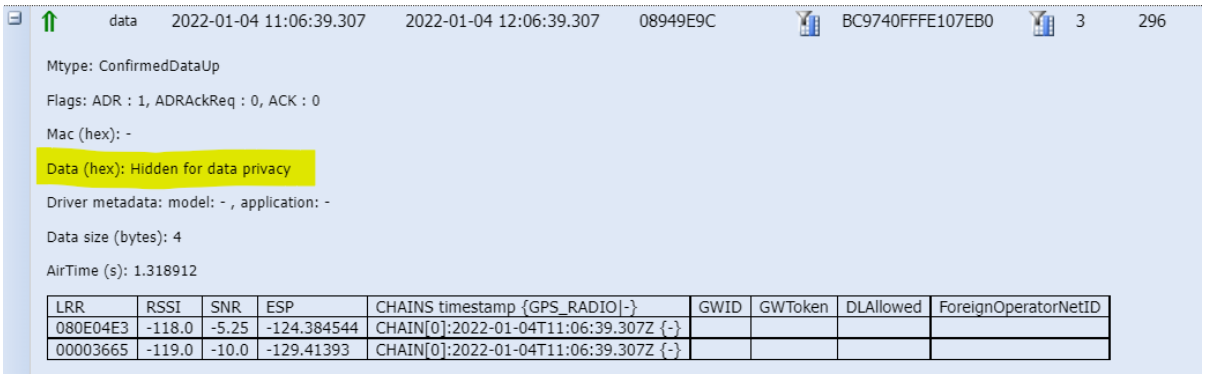
AS ID: TWA_100000304.60151.AS

Frequency (MHz): 866.5

Current class: C

AS ID	Status	Transmission errors
TWA_100000304.60151.AS	Error	Idx: 0 Url: https://asdeliverystatus.free.beeceptor.com Status: Error Error: 503 HTTP Error

Note: Your payload data is not stored on the LPN Network server. It can therefore not be displayed here.



data 2022-01-04 11:06:39.307 2022-01-04 12:06:39.307 08949E9C BC9740FFFE107EB0 3 296

Mtype: ConfirmedDataUp

Flags: ADR : 1, ADRAckReq : 0, ACK : 0

Mac (hex): -

Data (hex): Hidden for data privacy

Driver metadata: model: - , application: -

Data size (bytes): 4

AirTime (s): 1.318912

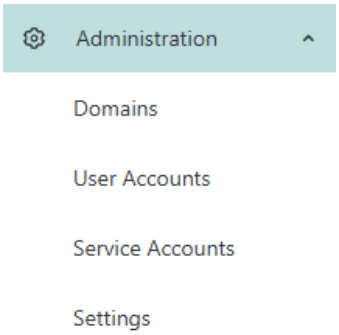
LRR	RSSI	SNR	ESP	CHAINS timestamp {GPS_RADIO|-}	GWID	GWToken	DLAllowed	ForeignOperatorNetID
080E04E3	-118.0	-5.25	-124.384544	CHAIN[0]:2022-01-04T11:06:39.307Z {-}				
00003665	-119.0	-10.0	-129.41393	CHAIN[0]:2022-01-04T11:06:39.307Z {-}				

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

2.8. Administration

Access Management is available via the Administration section in the left side bar.



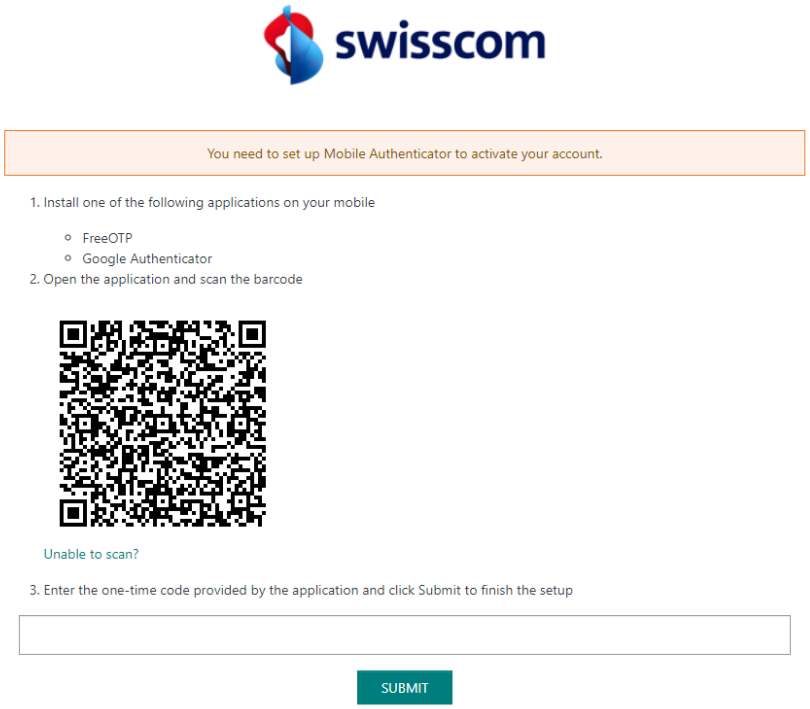
2.8.1 User Accounts

This submenu allows to manage User Accounts of the subscription. Click on an existing user to edit. To reset the password, lock or delete an end user, click on three dots on the right.

When adding a new end user, make sure to **not** give admin rights, as only one user should be admin of your subscription.

2.8.2 Multi factor authentication (MFA) setup

Upon your first login to the LPN Portal or IoT Flow, you will need to scan the QR Code with an authenticator app on your smartphone to activate multi factor authentication. You can use any authenticator app such as FreeOTP, Google Authenticator, Microsoft Authenticator, Authy, etc. In any case you need to add an account, scan the QR code, enter the code displayed in the app to the portal and click submit. With every login you will need to put the one-time code generated in your authenticator app. If you need to relink the account to a new app, click on "forgot password" in the login interface to start the process again.

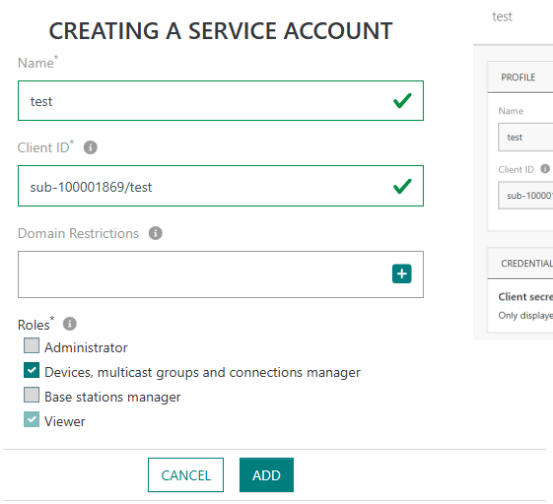


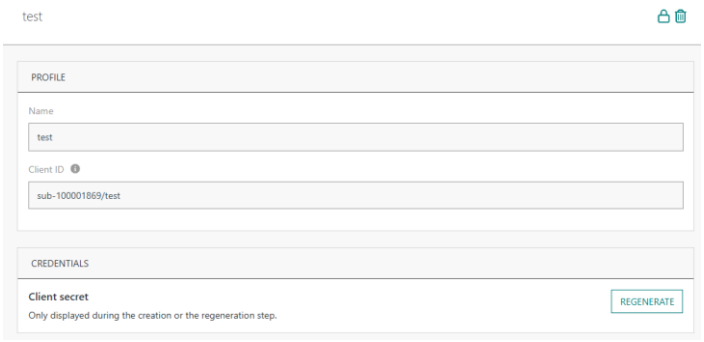
The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

2.8.3 Service Accounts

Serivce Accounts are used to access the OSS API. The first Service Account of your subscription needs to be added in the graphical user interface. This created Service Account can then create more Service Accounts via the OSS API. To create the first service account, go to Administration/Service Accounts/Create Service Account, fill out the form and click add. Then the secret of this service account is displayed. Make sure to save it accordingly as it will not be shown again. It is only possible to create a new secret with the regenerate button.





2.8.4 Domains

With Version 8 of the LPN Portal it is possible to group devices, gateways, users and service accounts into domains such that a user/service account can only manage the assets assigned to the same domain.

In order to set this up, the domain needs to be created first. In a second step the domain is assigned to the asset and the user. Multiple domains can be assigned to an asset or user.

2.8.5 Settings

In the settings menu, email alarms can be setup for devices and gateways. Add your email addresses in the according fields. For gateway alarms, there is also an option to send them via IoT Flow to your endpoint, which is recommended. Add a tag for gateway alarms in the field *ThingPark X IoT Flow*. This tag is then available to be used on IoT Flow to route gateway alarms. For a list of gateway alarm codes, follow this link.

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

3. Telemetry Interfaces

In the Swisscom LPN environment, the different APIs serve for different purposes. Here is a quick overview of the APIs and additional systems available, and where to find information.

API	Purpose	Rate limit	Documentation
Tunnel API (Uplink)	Receiving data uplinks from devices (REST API)	8 parallel HTTP sessions per Application Server Destination*	3.2 Tunnel Interface (Uplink)
Tunnel API (Downlink)	Send data downlink messages to the device	51 requests / s per source IP	3.3 Tunnel Interface (Downlink)
IoT Flow	Data connectors to third party systems (Azure, AWS...)	3000 msg/minute 50 msg/second	3.4 IoT Flow

* The maximum number of requests per second will depend on your application server's response time. This number can be increased if necessary, please contact us.

3.1. Tunnel Interface

The Swisscom LPN Portal provides an interface for applications to receive uplinks and send downlinks.

- **Tunnel mode interface**: A simple message passing interface between the Swisscom LPN servers which implement the network MAC layer (LRC servers), and application servers. This interface forwards the uplink radio packets raw payload data and associated metadata (RSSI, SNR...) to one or more connections associated to the network node MAC address. As the Swisscom LPN Portal supports bidirectional communications, application servers may also send requests to one of the LRC nodes to send downlink frames to a device with its unique DevEUI.

This document provides information on:

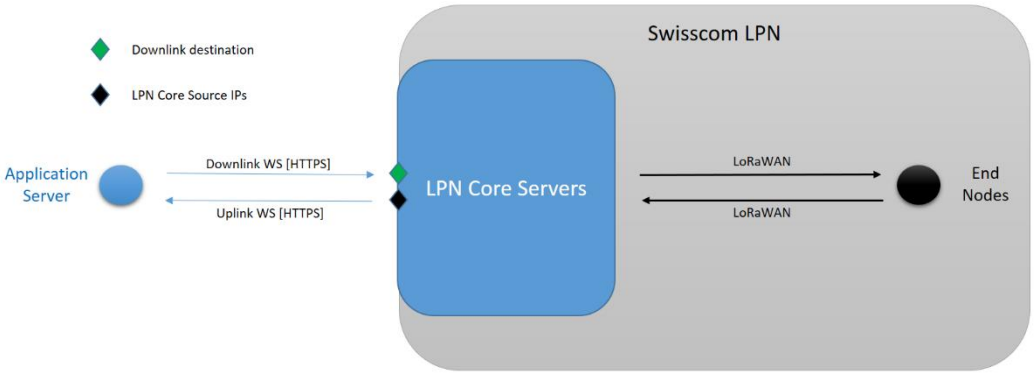
- Low level LPN core configuration parameters required to associate one or more application servers with a given network node MAC address.
- Format of LPN core to application server messages that encapsulate uplink payload data and associated metadata
- The format of application server to LPN core messages that encapsulate downlink payload.

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

3.1.1 Connectivity

The tunneling interface is based on HTTPS for uplink & downlink packets flows. The confidentiality over the Internet of the uplink/downlink are managed by the HTTPS connection. The uplink HTTPS session is mounted between the LRC cluster and the Application Server. The downlink HTTPS session is mounted



between the Application Server and the Reverse HTTPS proxy in front of the LRC cluster.

The uplink and downlink packets may be secured by an authentication layer. The LRC will share an AS key with the Application Server.

The AS key will be used by the Application Server to generate a signature added to downlink packets. This signature will be used by the LPN core to verify the identity and the authorization of the Application Server. If the identity or the authorization cannot be successfully verified, the packet will be dropped.

The AS key will also be send by the LPN core to generate a signature added to uplink packets. The signature will be used by the AS to verify the identity of the LPN core. If the identity cannot be successfully verified, the packet should be dropped by the AS.

- The downlink destination is as follows:

  https://proxy1.lpn.swisscom.ch/thingpark/lrc/rest/downlink/?DevEUI=<DEVEUI>&FPort=<FPORT>&Payload=<PAYLOAD>

- The LPN Portal source IPs for uplink messages which should be approved by your ASs firewall are the following:
  - 195.65.47.212
  - 195.65.47.242
  - 194.209.209.249

3.1.2 Parameters Format

3.1.2.1 ISO 8601 timestamps

All ISO 8601 timestamps described in this section use the following convention:

```
YYYY-MM-DDThh:mm:ss.s+|-hh:mm (e.g. 2016-08-01T09:06:06.0+02:00)
```

Where:

- YYYY = four-digit year
- MM = two-digit month
- DD = two-digit day
- hh = two digits of hour (00 through 23)
- mm = two digits of minute (00 through 59)
- ss = two digits of second (00 through 59)
- s = one to three digits of millisecond (0 through 999)

- +|-hh:mm = timezone designator (+hh:mm or –hh:mm)

All timestamp parameters described above are mandatory.

3.1.2.2 Timestamp Encoding

The timestamp parameter has to be encoded differently in the sha-256 and in the URL of the HTTP request:

Special Characters	In SHA256, use…	In URL, use…
+	+	%2B
-	-	-
:	:	:
.	.	.
Examples	2016-08-01T09:06:06.0+02:00 2016-11-28T09:06:06.0-04:00	2016-08-01T09:06:06.0%2B02:00 2016-11-28T09:06:06.0-04:00

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/Ipn

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/Ipn

3.2. Tunnel Interface (Uplink)

This section describes the two reports that are generated from the LRC to an Application Server.

3.2.1 Uplink Frame Report

Uplink destination URLs are defined for each device in the AS profile. The main criteria is the LoRaWAN port numbers expressed as ranges, lists or single values. A default LoRaWAN port destination can be declared. Sequential mode allows packets to be delivered to a list of destinations until one of them confirms reception (200 OK).

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. The XML/JSON payload sent to application servers still include the RF metadata corresponding to all receiving base stations.

The uplink frame is transmitted in a HTTP/POST request with query parameters and an XML/JSON payload. Following query parameters are defined for uplink frames (lexicographic order):

AS_ID	Application Server ID (only reported when the authentication is activated) **NOTE: AS_ID** is not reported when the Uplink/downlink authentication is not activated in the AS profile.
LrnDevEui	Device EUI.
LrnFPort	LoRaWAN port number.
LrnInfos	Service profile name used to route the packet.
Time	ISO 8601 timestamp associated to generation of the HTTP request by the LRC (only reported when the authentication is activated) **NOTE: Time** is not reported when the Uplink/downlink authentication is not activated in the AS profile.
Token	Security token generated by the LRC (only reported when the authentication is activated) **NOTE: Token** is not reported when the Uplink/downlink authentication is not activated in the AS profile.

Following elements are defined for uplink frames (lexicographic order):

ACKbit	ACKBit set by the device. **NOTE: ACKbit** is not filled in the XML document if not set in the uplink frame.
ADRbit	ADRBit set by the device. **NOTE: ADRbit** is not filled in the XML document if not set in the uplink frame.
AppSKey	Encrypted Appskey with the ASkey.
Channel	LC used by the device.
CustomerData	ASCII customer data set by provisioning.
CustomerID	Customer ID associated to the Device Manager account.
DevAddr	Device address.
DevEUI	Device EUI.
DevLrrCnt	Number of LRRs which received this packet.
DriverCfg	Reports the device profile configuration. Should be a *swisscom* profile if you have registered your device correctly.

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

DynamicClass	The current LoRaWAN class in which the device is currently set, as devices can change class dynamically.
FCntDn	The counter value to be used for the next downlink frame.
FCntUp	The uplink counter for this packet.
LostUplinksAS	Number of uplink frames lost since the reporting of the previous uplink frame for the same device.
FPort	LoRaWAN FPort used by the device for this packet.
Frequency	The frequency of this LoRaWAN uplink
InstantPER	Instant PER (Packet Error Rate). **NOTE:** The instance PER is computed on the last 10 packets.
Late	Indicate if the packet was queued by the LRR. **NOTE: Late** is always filled. 0 means that the packet was not queued by the LRR, 1 means that the packet was queued (the LRR queues packets when the connection between the LRR and the LRC is temporarily broken).
Lrcid	ID of the LRC that processed the packet.
Lrrid	The ID of the LRR that received the packet with the best ESP. This LRR is flagged as "best LRR".
LrrLAT **LrrLON**	LAT and LON of the best LRR.
Lrrs	Can contain up to 10 gateways seen by the device.
LrrRSSI	RSSI measured by the best LRR.
LrrSNR	SNR measured by the best LRR.
Lrrs/Lrr/Lrrid	LRR ID associated to this **<Lrr>** XML element.
Lrrs/Lrr/LrrESP	ESP measured by the LRR associated to this <**Lrr**> XML element.
Lrrs/Lrr/LrrRSSI	RSSI measured by the LRR associated to this <**Lrr**> XML element.
Lrrs/Lrr/LrrSNR	SNR measured by the LRR associated to this <**Lrr**> XML element.
MeanPER	Mean PER (Packet Error Rate). **NOTE:** The Mean PER is the average of the instantaneous PER of the last 20 packets.
mic_hex	MIC in hexadecimal ASCII format.
ModelCfg	ASCII ThingPark Cloud data set by provisioning.
MType	LoRaWAN MType of the packet.
NbTrans	Number of transmissions for each uplink frame.
payload_hex	LoRaWAN payload in hexadecimal ASCII format.
SpFact	SF used by the device.
SubBand	SUB-BAND used by the device.
Time	LRR Timestamp for the packet.
TxPower	End-device transmission power in dBm. Reported in ERP for LoRaWAN 1.0/1.0.1 and in EIRP for LoRaWAN 1.0.2 revB and above.

NOTE: The **Lrrs** element is reported for max. 10 LRRs which received the packet. If the packet was received by more than 10 LRRs, only the 10 LRRs with the best ESP are reported.

NOTE: When implementing your parser, please be aware that additional fields may be added in future updates and that most of the fields are optional.

3.2.2 Sample of Uplink Frame HTTP Request

The payload may be provided to the application server either encrypted or decrypted. The following rules apply:

ABP device	Payload is provided decrypted to the AS, if the AppSKey has been provisioned for the relevant FPort. Otherwise the Payload is provided encrypted to the AS.
OTAA device Embedded security server.	Payload is provided decrypted to the AS
OTAA device HSM	The payload and AppSKey are provided encrypted to the AS.

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/Ipn

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

In this sample, *<as-url>* is the destination URL configured in the AS profile:

Note: In an URL, the "+" character must be escaped.

```
>> POST <as-url>?LrnDevEui=000000000F1D8693&LrnFPort=2&LrnInfos=UPHTTP_LAB_LO
RA&AS_ID=app1.sample.com&Time=2016-01-11T14:11:11.333%2B02:00&Token=fd0b0b004
64aa798a59282d64eaa70813e33bff87682880db49638569d096aad

<?xml version="1.0" encoding="UTF-8"?>
<DevEUI_uplink xmlns="http://uri.actility.com/lora">
  <Time>2020-06-16T07:27:01.618+02:00</Time>
  <DevEUI>0018B20000000CD7</DevEUI>
  <FPort>1</FPort>
  <FCntUp>4</FCntUp>
  <LostUplinksAS>0</LostUplinksAS>
  <ADRbit>1</ADRbit>
  <MType>4</MType>
  <FCntDn>4</FCntDn>
  <payload_hex>af1c03030e681f07</payload_hex>
  <mic_hex>9fa820cd</mic_hex>
  <Lrcid>00000404</Lrcid>
  <LrrRSSI>-28.000000</LrrRSSI>
  <LrrSNR>10.750000</LrrSNR>
  <SpFact>12</SpFact>
  <SubBand>G0</SubBand>
  <Channel>LC1</Channel>
  <DevLrrCnt>1</DevLrrCnt>
  <Lrrid>004A10D7</Lrrid>
  <Late>0</Late>
  <LrrLAT>0.000000</LrrLAT>
  <LrrLON>0.000000</LrrLON>
  <Lrrs>
    <Lrr>
      <Lrrid>004A10D7</Lrrid>
      <Chain>0</Chain>
      <LrrRSSI>-28.000000</LrrRSSI>
      <LrrSNR>10.750000</LrrSNR>
      <LrrESP>-28.350851</LrrESP>
    </Lrr>
  </Lrrs>
  <CustomerID>100001792</CustomerID>
  <CustomerData>{"alr":{"pro":"ADRF/DEMO","ver":"2"}}</CustomerData>
  <ModelCfg>0</ModelCfg>
  <DriverCfg>{"mod":{"pId":"swisscom","mId":"swisscom","ver":"1"},"app":{"pId
":"swisscom","mId":"swisscom","ver":"1"}}</DriverCfg>
  <InstantPER>0.000000</InstantPER>
  <MeanPER>0.000000</MeanPER>
  <DevAddr>099802E2</DevAddr>
  <TxPower>14.000000</TxPower>
  <NbTrans>1</NbTrans>
  <Frequency>868.1</Frequency>
  <DynamicClass>A</DynamicClass>
</DevEUI_uplink>
```

3.2.3 LRC Authentication for UL Frame and DL Frame

Securing LRC to AS frame is implemented with the following principles:

- The LRC adds the AS ID and the generation time stamp in the message.
- Then, the LRC adds a security token to sign the message based on a pre-shared AS key.
- When the AS receives a message, the AS will re-compute the security token.
- If the re-computed security token matches the security token provided by the LRC, and if the time deviation (between the generation by the LRC and the reception by the AS) is acceptable (e.g. less than 10 seconds), the AS can trust the message and process it accordingly.

The AS ID / AS Key are part of the AS profile configuration associated to the device. The generation of the security token by the LRC can be deactivated by not setting an AS_ID and AS key in the AS Profile.

3.2.4 Token must be verified as following by the Application Server:

- The application server retrieves the <query-parameters> WITHOUT the Token QP (Query parameters include the AS_ID and the Time):
  For an uplink frame (based on the example provided section 3.2.1):
  e.g. <query-parameters> :=
  LrnDevEui=000000000F1D8693&LrnFPort=2&LrnInfos=UPHTTP_LAB_LORA&AS_ID=app1.sample.com&Time=2016-01-11T14:11:11.333+02:00

- The application server builds the <body-elements> as the concatenation, without separator, of the following values:
  For an uplink frame (extract from the <DevEUI_uplink> body): CustomerID, DevEUI, FPort, FCntUp, payload_hex.
  e.g. <body-elements> := 100000507000000000F1D8693270110027bd00

- The application server re-computes the <token> as:
  SHA-256(<body-elements><query-parameters><AsKey>):
  For an uplink frame:
  e.g. <token> :=
  SHA-256(100000507000000000F1D8693270110027bd00LrnDevEui=000000000F1D8693&LrnFPort=2&LrnInfos=UPHTTP_LAB_LORA&AS_ID=app1.sample.com&Time=2016-01-11T14:11:11.333+02:0046ab678cd45df4a4e4b375Eacd096acc)

  Where 46ab678cd45df4a4e4b375Eacd096acc is the 128 bits pre-shared key (lower case hex string representation) between the Application Server and the LRC as defined in the AS profile.

- The <token> is encoded as a hex string AND can be compared to the <token> provided by the LRC in the <query parameters> line.

  For an uplink frame:
  e.g. <encrypted-token> :=
  fd0b0b00464aa798a59282d64eaa70813e33bff87682880db49638569d096aad

- Finally, if the token is valid, the application server can verify the deviation between the emission (as provided in the Time query parameter) and the reception by the application server.

3.2.5 XML or JSON Encoding

The Uplink frame HTTP request body may be encoded by the LRC has an XML or JSON payload, which can be configured in Device Manager > Application Server. The default is set to XML.

Information elements in the XML document be mapped one-to-one with information elements in the JSON document.

A 1-to-1 mapping must be assumed between information elements present in the XML document and information elements present in the JSON document.

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/Ipn

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

3.2.6 Sample of Uplink JSON Payload


```
{
  "DevEUI_uplink": {
    "Time": "2020-06-08T11:08:41.018+02:00",
    "DevEUI": "0018B20000000CD7",
    "FPort": 1,
    "FCntUp": 4,
    "LostUplinksAS": 0,
    "ACKbit": 1
    "ADRbit": 1,
    "MType": 4,
    "FCntDn": 4,
    "payload_hex": "af1a03030ea82107",
    "mic_hex": "78137e75",
    "Lrcid": "00000404",
    "LrrRSSI": -28.0,
    "LrrSNR": 10.5,
    "SpFact": 12,
    "SubBand": "G1",
    "Channel": "LC2",
    "DevLrrCnt": 1,
    "Lrrid": "004A10D7",
    "Late": 0,
    "LrrLAT": 0.0,
    "LrrLON": 0.0,
    "Lrrs": {
      "Lrr": [
        {
          "Lrrid": "004A10D7",
          "Chain": 0,
          "LrrRSSI": -28.0,
          "LrrSNR": 10.5,
          "LrrESP": -28.370777
        }
      ]
    },
    "CustomerID": "100001792",
    "CustomerData": {
      "alr": {
        "pro": "ADRF/DEMO",
        "ver": "2"
      }
    },
    "ModelCfg": "0",
    "DriverCfg": {
      "mod": {
        "pId": "swisscom",
        "mId": "swisscom",
        "ver": "1"
      },
      "app": {
        "pId": "swisscom",
        "mId": "swisscom",
        "ver": "1"
      }
    },
    "InstantPER": 0.0,
    "MeanPER": 0.0,
    "DevAddr": "099802E2",
    "TxPower": 14.0,
    "NbTrans": 1,
    "Frequency": 868.3,
    "DynamicClass": "A"
  }
}
```

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

3.3. Tunnel API (Downlink)

This section describes how downlinks are sent from the application server to a device.

URL base path:

V1: https://proxy1.lpn.swisscom.ch/thingpark/lrc/rest/downlink

V2: https://proxy1.lpn.swisscom.ch/thingpark/lrc/rest/v2/downlink

Using the v2 interface will provide more detailed HTTP responses, see section 3.3.5

3.3.1 Downlink Frame

Depending on the device provisioning encryption/decryption can be performed by the LRC. The following rules apply:

Note: When the payload is encrypted, Wireless Logger cannot decrypt it as this application does not embed the associated decryption key.

ABP device	Payload can be provided not encrypted by the AS and will be encrypted by the LRC, if AppSKey has been provisioned for the relevant FPort and if the FCntDn parameter is absent. Otherwise the Payload must be provided encrypted by the AS and the FCntDn parameter must be present.
OTAA device Embedded security server	Payload must be provided not encrypted by the AS and will be encrypted by the LRC. The FCntDn parameter must be absent.
OTAA device HSM	The payload must be provided encrypted by the AS. The FCntDn parameter must be present.

The following HTTP/POST message format is used to tunnel the radio frame payload and associated metadata from the target application server to the LRC. The application server acts as an HTTP client and the reverse HTTP proxy (PROXY_HTTP server) acts as a HTTP server. Rerouting of the HTTP request to the primary LRC or the backup LRC is handled by the reverse HTTP proxy.

The LoRaWAN MAC message integrity code (MIC) is always computed by the LRC, as part of the MAC frame formatting. The MAC payload may be encrypted either by the application or by the LRC (see table above).

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://pro
xy1.lpn.swisscom.ch/thingpark/lrc/rest/downlink?DevEUI=000000000F1D8693&FPort
=1&Payload=0102030405060708090A&FCntDn=1234"
```

Following query parameters are defined for downlink frame (lexicographic order):

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 and the FCntDn parameter must be present. 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 LRC) 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 LRC.
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).
FlushDownlinkQueue (Optional)	Empties the device AS downlink queue of the device (Boolean). When this parameter is set to FlushDownlinkQueue=1, the AS requests the LRC to purge the AS downlink queue of the device prior to add the downlink payload transported by this HTTP POST.
ValidityTime (Optional)	Associates the AS downlink payload with an expiration date (ISO 8601 timestamp or Duration in seconds) in the device AS downlink queue. If the AS downlink payload has not yet been sent to the device, the AS downlink payload will be discarded by the LRC when the expiration date is reached.
AS_ID (Optional)	Application Server ID, as provisioned in the AS Profile. The Application server ID is mandatory if the Application server authentication has been activated in the AS Profile. In this case the LRC will check that the Application Server is authorized to send downlink command to the device.
Time (Optional)	ISO 8601 time of the request. The Time is mandatory when the Application server authentication has been activated in the AS Profile. In this case the LRC will verify the time deviation between the generation and the reception of the request. The deviation must be lower than “Max Time Deviation” as defined in the AS Profile. **Note:** In the URL of the HTTP request, use "%2B" ASCII code for the "+" character.
Token (Optional)	Security token to sign the downlink frame. The Token is mandatory when the Application server authentication has been activated in the AS Profile.

3.3.2 Sample of Downlink Frame HTTP Request

Sample with v1:

```
>> POST https://proxy1.lpn.swisscom.ch/thingpark/lrc/rest/downlink?DevEUI=000000000F1D8693
&FPort=1&Payload=00&AS_ID=app1.sample.com&Time=2016-01-11T14:28:00.333%2B02:00&Token=ea8f3
1d2299cbece8e180a3012766c4df15fe3cf2e142d9fdf4035b5894ec886
```

Sample with v2:

```
>> POST https://proxy1.lpn.swisscom.ch/thingpark/lrc/rest/v2/downlink?DevEUI=000000000F1D8
693&FPort=1&Payload=00&AS_ID=app1.sample.com&Time=2016-01-11T14:28:00.333%2B02:00&Token=ea
8f31d2299cbece8e180a3012766c4df15fe3cf2e142d9fdf4035b5894ec886
```

Confirmed frames are declared as a message in the LPN Portal, therefore a confirmation is subject to deduction of your up-/downlink budget.

3.3.3 Confirmed Downlink

Unconfirmed Downlink messages are not acknowledged at LoRaWAN level and therefore the network, and the tunnel mode Application server does not know whether they have been received or not.

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/Ipn

Confirmed Downlink messages are acknowledged by the target device, but according to LoRaWAN specification "Message acknowledge bit and acknowledgement procedure (ACK in FCtrl)" lets the device free of sending delayed ACKs. Therefore, it is not possible to let the network manage retransmissions.

When the LRC receives a possibly empty (no payload) uplink message with ACK set in the FCtrl field, the LRC will add an "ACKbit" flag in the XML/JSON metadata of the uplink frame sent to the Application server. The retransmit policy is up to the application server.

3.3.4 LRC HTTP Response Codes (v1):

- **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":** An Invalid DevEUI was entered.
- **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.
- **350 "Invalid LoRa port 0":** sending on port 0 (port reserved for LoRaWan MAC commands) is unauthorized from the tunneling interface.
- **350 "Security Check. AS_ID is mandatory":** speaking to this device needs AS_ID. The Application Server authorization has been activated for this device and the application must be identified.
- **350 "Security Check. missing timestamp/token":** Time and Token query parameter are mandatory when application server authentication is activated.
- **350 "Security Check. bad AS_ID":** AS_ID is not declared in the database or is not authorized for the targeted device.
- **350 "Security Check. Server Decrypt Error":** Missing or badly formatted security token.
- **350 "**Security **Check. malformed ISO8601 time":** An ISO 8601 date/time must be used (YYYY-MM-DDThh:mm:ss.s+|-hh:mm) representing a local time with a time zone offset in hours and minutes.
- **350 "Security Check. Invalid downlink frame timestamp":** the time deviation between the frame generation by the application server and the reception by the LRC exceeds the MAX deviation configured in AS profile.
- 350 **"Security Check. bad token":** Token was not accepted by the LRC
- 350 **"ValidityTime expired or invalid":** The date or duration in the ValidityTime attribute is wrongly formatted or is invalid (for instance, date in the past).
- **350 "Payload too big or invalid size":** The payload size is greater than 5000 characters or is not a multiple of 2.
- **350 "Downlink transmission disabled":** The request for transmission of the downlink packet was rejected by the LRC due to the absence of the downlinkTransmission feature flag in the Connectivity plan associated to the device.
- **404 "No Base Station Available":** The request for transmission of the downlink packet was rejected by the LRC because no base station near to the device is connected.
- **350 "Invalid payload size according to currents DR":** The payload is too big to be transmitted according to the data rate currently used by the device.
- **503 "service temp unavailable":** The service is temporarily unavailable due to throttling. Try again at a later point.

3.3.5 LRC HTTP Response Codes (v2):

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/Ipn

HTTP response when the downlink frame is accepted:

- HTTP Response status code: 202
- HTTP Response body: empty.

HTTP response when the downlink frame is rejected:

- HTTP Response status code: 409
- HTTP Response body: JSON document

```
{
 "code": <code>,
 "message": "<message>"[,
 "xxx:": <additional info 1>][,
 "yyy:": <additional info 2>]
 ...
}
```

- The code is mandatory (see table below) and is an immutable unsigned integer.
- The message is mandatory (see table below) and is an ASCII string.
- On an error basis, additional information elements (xxx, yyy...) may be added to the JSON document (see table below).
- The table below gives the description of the error codes

Error code	Error message	Additional info 1	Additional info 2
104	Invalid LoRa port	-	-
105	Security Check: AS_ID is mandatory	-	-
106	Security Check: Missing timestamp	-	-
107	Security Check: Bad AS_ID	-	-
108	Security Check: Server Decrypt Error	-	-
109	Security Check: Malformed ISO 8601 time	-	-
110	Security Check: Invalid downlink frame timestamp	-	-
111	Security Check: Bad token	-	-
112	Validity time expired or invalid	-	-
113	Payload too big or invalid size	Attribute: maxSize Type: Integer	-

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

		Description: When set this means that the max payload size was reached. When not set this means that the payload size is invalid (must be a multiple of 2).	
114	Downlink transmission disabled	-	-
115	No Base Station Available	-	-
116	Invalid payload size according to currents DR	Attribute: rx2DR or pingSlotDR Type: Integer Description: MAX Data Rate associated to RX2 (class A/C) or PingSlot (class B). The current RX1 DR is not checked.	Attribute: maxSize Type: Integer Description: Max payload size according to RX2 (class A/C) or PingSlot (class B). The current RX1 DR is not checked.
117	Invalid Correlation ID. Must be a 64 bits hexadecimal value encoded as string	-	-
118	Invalid payload. Must not be empty	-	-
119	Payload must be provided encrypted with the downlink counter value	-	-

Note: regarding queueing of several messages: The Swisscom LPN Portal Wireless network may queue up to 5 messages per device. The network uses the FPending flag defined in the LoRaWAN protocol to signal to the device that additional messages are queued. Messages will be sent, one at a time, in the receive window following the next uplink message received from the device. A downlink has no timeout in the queue unless it is forced by the "FlushDownlinkQueue" query parameter.

3.3.6 Multicast Downlink

At Phy level Multicast support in LoRaWAN class B and class C amounts to having multiple End devices listen to the same network address. This is already supported as part of class B and class C support in the Swisscom LPN Portal, an Application server just needs to send an unconfirmed downlink message to the target group address (configured as dummy device).

However, the LoRaWAN roadmap includes future work on multicast, including group membership and key management procedures, as well as large payload fragmentation transmission e.g. for firmware updates.

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

3.3.7 Application Server Authentication for Downlink Frame

Securing downlink frame is implemented with the following principles. If there is no AS routing profil assigned to a device, the downlink securization cannot be enabled and unauthentified downlinks can be queued.

The AS must not be able to send downlink POST if:

- The AS is not in possession of AS key.
- The AS has not been authorized to send downlink packet to the device
- The time between the generation of the request by the AS and the reception of the request by the LRC is too high.

The AS ID / AS Key and max time deviation are part of the AS profile configuration associated to the device. The Application Server authentication can be deactivated by not setting an AS_ID and AS key in the AS Profile.

3.3.8 Token computation by the Application Server

- The downlink message <query-parameters> (Query parameters must include the AS_ID and the Time query parameters) are constructed WITHOUT the Token:
  `e.g. <query-parameters> := DevEUI=000000000F1D8693&FPort=1&Payload=00&AS_ID=app1.sample.com&Time=2016-01-11T14:28:00.333+02:00`
- The <token> is computed as SHA-256(<query-parameters><Askey>)
  e.g. <token> := SHA-256(DevEUI=000000000F1D8693&FPort=1&Payload=00&AS_ID=app1.sample.com&Time=2016-01-11T14:28:00.333+02:0046ab678cd45df4a4e4b375Eacd096acc)
  where 46ab678cd45df4a4e4b375Eacd096acc is the 128 bits pre-shared key (lower case hex string representation) between the Application Server and the LRC as defined in the AS profile.
- The <token> is encoded as an hex string (e.g. `ea8f31d2299cbece8e180a3012766c4df15fe3cf2e142d9fdf4035b5894ec886`) AND added at the end of the query parameters line
  `e.g.`
  `https://proxy1.lpn.swisscom.ch/thingpark/lrc/rest/downlink?DevEUI=000000000F1D8693&FPort=1&Payload=00&AS_ID=app1.sample.com&Time=2016-01-11T14:28:00.333%2B02:00&Token=ea8f31d2299cbece8e180a3012766c4df15fe3cf2e142d9fdf4035b5894ec886`
  where: `2016-01-11T14:28:00.333%2B02:00` contains the "%2B" ASCII code for the "+" character where as the ":" character has to be sent unescaped.

3.4. IoT Flow

3.4.1 Queueing behavior

The default queueing time "uplinkTimeValidity" is set to 72h. This means that, in case of a connection failure or server downtime, the messages will be stored in a Kafka topic for 72h and IoT Flow will attempt to redeliver the packets. Messages older than "uplinkTimeValidity" will be discarded. The queuing behavior can be adapted if necessary.

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

3.4.2 IoT Flow source IP addresses

Messages from IoT Flow will be sent from one of the following IP addresses:

- 54.194.219.224
- 52.211.61.192
- 52.215.140.85

3.4.3 Sample of Uplink JSON with decoded payload

Below listed is an uplink JSON with a driver selected. Therefore, it additionally includes the decoded payload according to the selected driver. If the flow does not decode the payload, there is no entry for payload, only for the plain payload_hex.


```
{
   "DevEUI_uplink" : {
            "Time": "2020-01-06T12:46:54.285+01:00",
            "DevEUI": "20635F0108000E09",
            "FPort": 17,
            "FCntUp": 265,

            "LostUplinksAS": 0,
            "ADRbit": 1,
            "MType": 2,
            "FCntDn": 9,
            "payload_hex": "0520be8800400108c9000000",
            "mic_hex": "3bcf2927",
            "Lrcid": "00000127",
            "LrrRSSI": -65.0,
            "LrrSNR": 9.5,
            "SpFact": 7,
            "SubBand": "G1",
            "Channel": "LC3",
            "DevLrrCnt": 2,
            "Lrrid": "08050376",
            "Late": 0,
            "LrrLAT": 43.615501,
            "LrrLON": 7.066182,
            "Lrrs": {
                    "Lrr": [
                            {
                                    "Lrrid": "08050376",
                                    "Chain": 0,
                                    "LrrRSSI": -65.0,
                                    "LrrSNR": 9.5,
                                    "LrrESP": -65.461838
                            }
                    ]
            },
            "CustomerID": "1000xxxx",
            "CustomerData": {
                    "alr": {
                            "pro": "ABEE/APY",
                            "ver": "1"
                    }
            },
            "DriverCfg":{
                    "id": "abeeway:asset-tracker:1"
            },
            "payload": {
                    "messageType": "HEARTBEAT",
                    "mode": "MOTION_TRACKING",
                    "batteryVoltage": 3.85,
                    "ackToken": 0,
                    "firmwareVersion": "1.8.201",
                    "bleFwVersion": "0.0.0",
                    "resetCause": 40,
```

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn


```
                    "periodicPosition": false,
                    "temperature": 24.8,
                    "userAction": 0,
                    "appState": 0,
                    "moving": false,
                    "onDemand": false,
                    "payload": "0520be8800400108c9000000",
                    "InstantPER": 0.0,
                    "MeanPER": 0.0,
                    "DevAddr": "055E1C4E",
                    "AckRequested": 0,
                    "rawMacCommands": "",
                    "TxPower": 2.0,
                    "NbTrans": 1,
                    "Frequency": 867.9,
                    "DynamicClass": "A"
                }
            }
}
```


The structure of the Uplink JSON payload can be adjusted using Uplink Transformation on your connection. For more information, see section 2.5.5.

3.4.4 Sample of Downlink JSON payload

For any connector, the following scheme must be used by IoT cloud platforms for downlink messages to IoT Flow.


```
{
    "DevEUI_downlink": {
        "Time": "2019-07-10T15:38:46.882+02:00",
        "DevEUI": "0018B20000000B20",
        "FPort": 1,
        "payload_hex": "9e1c4852512000220020e3831071"
    }
}
```


This downlink message also contains optional fields.


```
{
    "DevEUI_downlink": {
        "Time": "2019-07-10T15:38:46.882+02:00",
        "DevEUI": "0018B20000000B20",
        "FPort": 1,
        "AS_ID": "TWA_199983788.1972.AS",
        "AS_KEY": "9311e22d7d44fc52215b0dc154aa1d22",
        "payload": {
            "DownMessageType": "SET_PARAMETER",
            "ParameterName": "TRANSMIT_STRAT",
            "TransmitStrat": "DOUBLE_FIXED",
            "AckToken": 1
        },
        "Confirmed": "1",
        "ValidityTime": "2019-07-10T16:38:46.882+02:00",
        "FlushDownlinkQueue": "1",
        "DriverCfg": {
            "app": {
                "pId": "abeeway",
                "mId": "asset-tracker",
                "ver": "1"
            }
        }
    }
}
```

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

4. Management APIs

At the moment there are two Management APIs available. The OSS API is the recommended and preferred API, as it offers more robust performance and DX API is no longer developed further. DX API shall only be used if an integration already exists in your application. DX API will exist at least until end of 2025.

API	Purpose	Rate limit
OSS API (recommended)	Device lifecycle, Gateway Monitoring, Administration	Depends on request category*: low-impact – 20 reqs/s per source IP medium-impact – 6 reqs/s per source IP high-impact – 2 reqs/s per source IP
DX API (deprecated)	Device lifecycle (create, delete, change price plan...)	10 requests / s per source IP

*Check OSS API yaml file for full list of request categories.

4.1. OSS API

With the OSS API, the LPN Platform exposes many REST APIs, which are the primary way to interact with the LPN System. Use the OSS API when building an application, scripts or any other integration.

4.1.1 Quickstart

Follow these steps to get started with OSS API:

- Use the **swisscom hostname**: portal.lpn.swisscom.ch
- Add a service account: follow the steps in section 2.8.3 and take note of the ID and secret
- Issue an access token:

```
ACCESS_TOKEN=$(curl \
-d "client_id=${CLIENT_ID}" \
-d "client_secret=${CLIENT_SECRET}" \
-d "grant_type=client_credentials" \
https://portal.lpn.swisscom.ch/users-auth/protocol/openid-connect/token | jq
-r '.access_token')
```

- Retrieve Devices:

```
curl \
-H "Authorization: Bearer ${ACCESS_TOKEN}" \
-H "Accept: application/json" \
https://portal.lpn.swisscom.ch/thingpark/wireless/rest/subscriptions/mine/dev
ices
```

4.1.2 Authentication

Authentication in OSS API relies on OAuth v2. All requests should include a valid Bearer token in the Authorization HTTP header. The Bearer token is an OAuth v2 access token issued using the Client Credentials grant. The URL of the OAuth token endpoint is as follows:

- OAuth URL token endpoint:
  https://portal.lpn.swisscom.ch/users-auth/protocol/openid-connect/token

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/Ipn

To get an access token, a POST request needs to be sent on this endpoint providing a client ID and a client secret of a service account in the body. The HTTP response is a JSON object containing the following properties:

- access_token
- expires_in: the number of seconds that the access token will be valid.

An access token is usually valid for 5 minutes. When the token is about to expire, a new one should be created with the same procedure. Example authentication request and answer:

```
POST /users-auth/protocol/openid-connect/token HTTP/1.1
Host: PLATFORM-HOSTNAME
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials
&client_id=SERVICE-ACCOUNT-CLIENT-ID
&client_secret=SERVICE-ACCOUNT-CLIENT-SECRET

HTTP/1.1 200 OK
Content-Type: application/json

{"access_token":"...","expires_in":300,"refresh_expires_in":0,"token_type":"B
earer","not-before-policy":0,"scope":"email profile"}
```

4.1.2.1 Token renewal

When running an application that runs longer than the access token lifetime (5 minutes), token renewal must be defined. There are code examples to help you get started available at: https://docs.thingpark.com/thingpark-wireless/latest/docs/api-subscriber/control-plane/token-renewal

4.1.3 Devices and connections API

The devices and connections API is the core of OSS API allowing complete management of devices and connections. See the Swagger page below to get startet and download the yaml file for easy integration to your application:

- API Reference: https://docs.thingpark.com/thingpark-wireless/latest/docs/api-subscriber-tpw/control-plane/device-manager/api-reference
- The downloadable yaml file is at the top of the page.

Note: If you have used an integration with DX API and want to switch from DX API to OSS API, you **must** upgrade your API integration to direct association, i.e, switch from AS *routing profiles* to *Connections*, to be able to use OSS API. This can be done in the GUI and via API per device. For instruction steps, follow this link: https://docs.thingpark.com/thingpark-wireless/latest/docs/api-subscriber-tpw/control-plane/device-manager/upgrade-to-direct-association

4.1.4 Gateways API

The Gateway API allows to easily manage and monitor your gateways. Use the Swagger page below to get startet and download the yaml file for easy integration to your application:

- API Reference: https://docs.thingpark.com/thingpark-wireless/latest/docs/api-subscriber-tpw/control-plane/network-manager/api-reference
- The downloadable yaml file is at the top of the page.

4.1.5 Administration API

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

Manage User Accounts, Service Accounts and domains with the Administration API to have full access control from your application. See the Swagger page below to get startet and download the yaml file for easy integration to your application:

- API Reference: https://docs.thingpark.com/thingpark-wireless/latest/docs/api-subscriber-tpw/control-plane/administration/api-reference
- The downloadable yaml file is at the top of the page.

4.1.6 Error Codes

For the full List of error codes please see this list:

- List of error codes: https://docs.thingpark.com/thingpark-wireless/latest/docs/api-subscriber/control-plane/error-codes

4.2. DX API (deprecated)

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.

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

Documentation: https://portal.lpn.swisscom.ch/thingpark/dx/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://portal.lpn.swisscom.ch/thingpark/dx/core/latest/tpdx-core-api-contract.yaml) of the DX API, or start using the API right away with the Swagger UI (https://portal.lpn.swisscom.ch/thingpark/dx/core/latest/swagger-ui/index.html?shortUrl=tpdx-core-api-contract.json).

4.2.1 DX API 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://portal.lpn.swisscom.ch/thingpark/dx/admin/latest/swagger-ui/index.html?shortUrl=tpdx-admin-api-contract.json), by providing the valid existing LPN Portal credentials. For more convenience, the Swagger UI can also by providing the login credentials on Core API as described in chapter 4.2.3.

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.

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

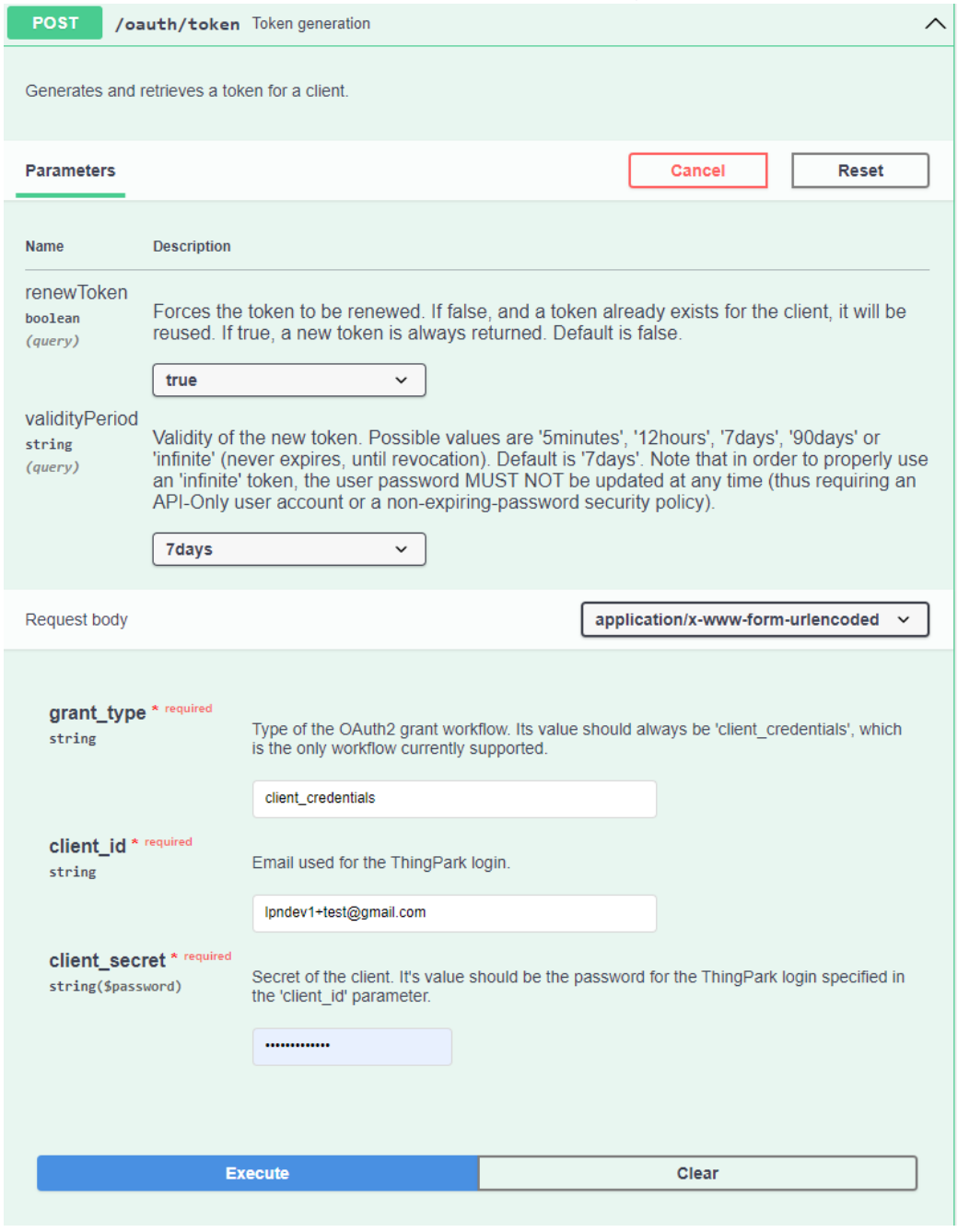
Note: To avoid token issues, please use a separate user for DX-API rather than an existing GUI-user.

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/Ipn

4.2.2 DX Admin API

The Admin API is used to get the Bearer Token to authenticate for API calls. Follow the instructions of the page and use a valid client_id and client_secret from Device Manager.



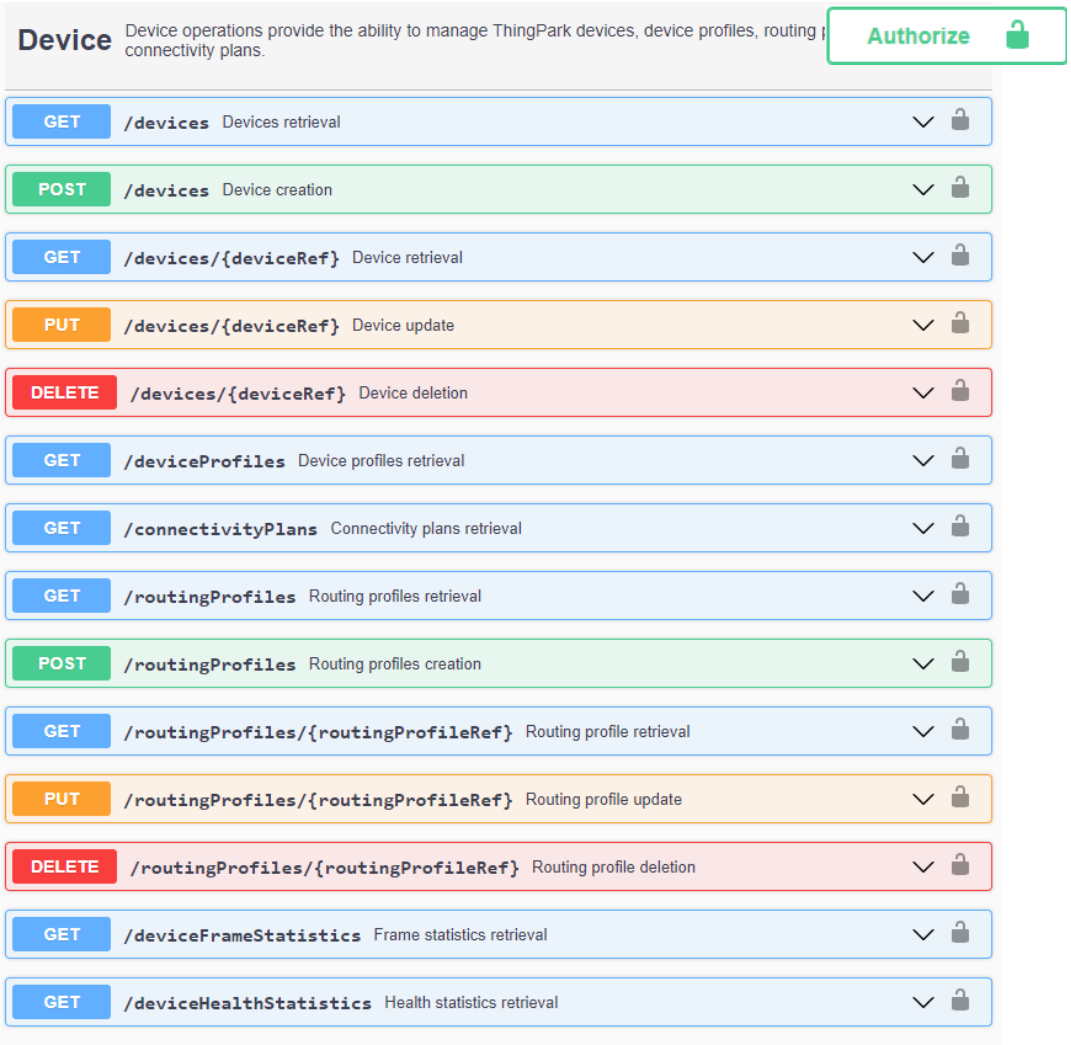
Please note that a user's password will expire after 4 months, unless you create an user with the flag **permanent password**.

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

4.2.3 DX Core API

DX Core API used for everything else e.g. creation of end users, device (add, delete, change connectivity plan, etc.), base station operations (get, rename) or to retrieve alarms. It is also possible to send downlinks for testing purposes. To authorize, the Authorize button on top can be used to log in with a Thingpark account (via Admin API) or to enter the bearer token, incase you have it already.



Please note that sending Downlinks via DX API is only for testing purposes. The preferred way for sending downlinks is using the Tunnel API. Tunnel API is the direct option and data won't leave Switzerland.

4.2.4 DX API error codes

In case of a problem, you might get an answer containing an OSS-xxx error code. The http error is always Error 400, and the specific error code can be seen in the body.

Example:

```
{"code":400,"message":"Bad request: The operation is still in progress and
has been switched to asynchronous mode. It should end in a few moments.","
errorId":"OSS-522"}
```

The complete list of error codes is available as an attachment to this document, please request it from support.LPN@swisscom.com.

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/Ipn

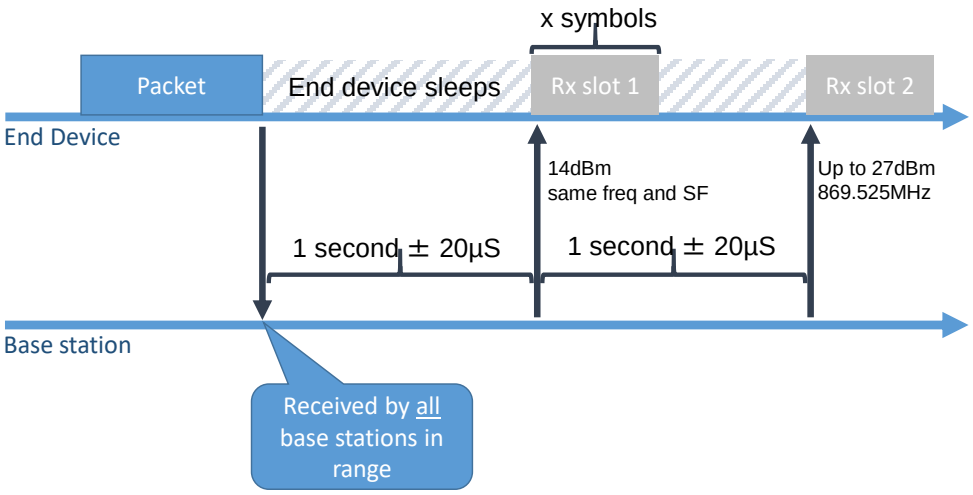
5. LoRaWAN specification

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.

5.1. 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

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

5.2. 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
JoinEUI (former AppEUI)	Operator	The AppEUI identifies the join server during the JOIN request.
AppKey	Device manufacturer	The AppKey encrypts the data during the JOIN request

Starting with LoRaWAN 1.0.4, the DevNonce sent in the Join-Request is a counter *and no longer a random value. The DevNonce has to be stored by the device,* ***resetting the DevNonce without changing the JoinEUI will cause the Join Server to discard the Join-Requests*** *of the end-device.*

5.2.1 128 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.

LoRaWAN uses symmetric encryption, which means that the same key needs to be provisioned both on the device and on the network server. The 128 bit AppKey is never transmitted on the air. Typically, the device manufacturer provisions a unique AppKey into each device and transmits it to you in a secure way, so that you can provision the correct key for each device in the network server, using either the GUI, the API or CSV upload.

5.2.2 64 bit application server identifier: JoinEUI (former AppEUI)

The JoinEUI, previously also called AppEUI, is a globally unique identifier of the target application server that will process all exchanges with the device. It will have an increasing significance when it comes to roaming between different operators.

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 JoinEUI is **F0:3D:29:AC:71:00:00:01**

Some device manufactureres still use other or random JoinEUIs. But with the increasing number of roaming partners it is important to choose the Swisscom AppEUI, or roaming will never be possible with this device.

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

5.3. 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

5.3.1 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.

Note: If you really need to use ABP for some reason or testing, Swisscom can provide you with a DevAddr range. Otherwise, please avoid ABP as much as possible and use OTAA instead. Please mind that ABP shall not be used in production. **Please do not choose a random DevAddr youself, this can lead to problems with your connection.**

5.3.2 128 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).

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

5.4. Channel plans 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 currently 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: 866.5 MHz
- LC7: 866.7 MHz
- LC8: 867.9 MHz
- RX2: 869.525 MHz / SF12

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

The information in this document does not constitute a binding offer. It is subject to revision at any time.

Swisscom (Switzerland) Ltd Enterprise Customers, Internet of Things
Pfingstweidstrasse 51, CH-8005 Zürich, www.swisscom.ch/lpn

6. Developing on LoRaWAN

6.1. The LPN Developer Portal

The Swisscom LPN Developer Portal consolidates all the necessary information for developers in one place: Documentation, FAQ, news, device qualification and new features to test. Account creation can be requested by email at support.lpn@swisscom.com.

6.2. Swisscom IoT Qualified for LoRaWAN devices

Before starting your IoT project, you want to make sure that your device properly interoperates with the Swisscom LPN network. Therefore, you should only use devices that have passed the "Swisscom IoT Qualified" tests. Customers are also contractually obligated to use only qualified devices when deploying more than 50 devices of the same type.

If you are a device manufacturer, you can use the self-testing function on the developer portal to qualify your device. This function can also be used for purchasers to evaluate devices from different manufacturers.

All the doucmentation can be found on the dedicated MyCloud folder: https://www.mycloud.ch/s/S00C76CB14E193935BDF4575858ADC9441C3B1AF099

There you will also find the CLDQP (Collective LoRaWAN Device Qualification Program) form, which is used by the main public LoRaWAN operators and defines all questions and test procedures accepted by the participating operators.

For further information regarding the qualification, please refer to the quick guide documentation