FBL800 LoRaWAN Gateway

User Manual

V1.0.2

This manual applies to the products listed below: FBL800433-XXX, FBL8004335G-XXX, FBL800470-XXX, FBL8004705G-XXX, FBL800868-XXX, FBL8008685G-XXX, FBL800915-XXX, FBL8009155G-XXX, FBL800433-MZZ, FBL800470-MZZ, FBL800868-MZZ, FBL800915-MZZ.

Xiamen Four-Faith Communication Technology Co., Ltd.
https://www.fourfaith.com

Four-Faith
LoRa 4G/5G

Document Revision History

Date	Version	Explanation	Author
2023-08-04	V1.0.0	Initial Version	YSL
2023-11-28	V1.0.1	Add Installation Manual	YSL
2024-07-12	V1.0.2	English Version	Larry

Copyright Statement

Trademark Statement

Content

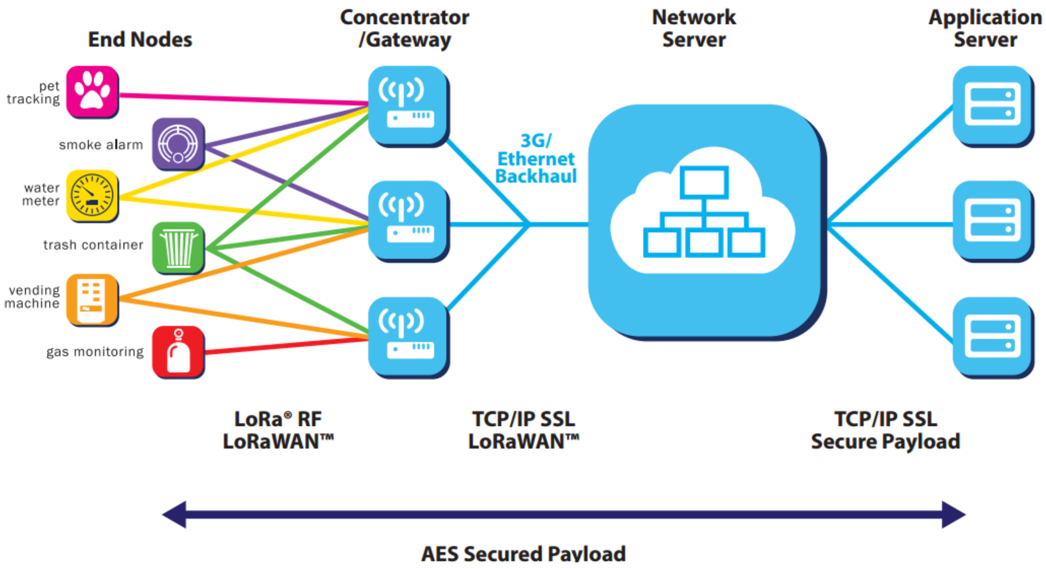
Chapter 1 Product Introduction

1.1 Product Overview

The FBL800 series device is a wireless communication base station based on the LoRaWAN protocol. It connects to various application nodes of LoRaWAN terminals, transmitting terminal information to the cloud through 4G or wired Ethernet. It supports WiFi wireless configuration management and online upgrades, GPS positioning, and can be powered by 220V AC or optional POE+.

The FBL800 gateway complies with the standard LoRaWAN protocol and supports multiple modes, including Embedded Network Server mode (Network Server deployed inside the gateway), Basicstation mode (connecting to an external Basicstation protocol server), and Semtech UDP GWMP Protocol mode (connecting to an external NS server via GWMP UDP protocol).

This product has been widely used in the M2M industry of the IoT ecosystem, including applications in smart meters, smart disasters, smart sensing, smart photovoltaics, smart grids, smart transportation, industrial automation, smart buildings, firefighting, public safety, environmental protection, meteorology, digital healthcare, remote sensing, military, space exploration, agriculture, forestry, water management, coal mining, petrochemicals, and other fields.



1.2 Product Features

Industrial-Grade Application Design:

- Utilizes high-performance industrial-grade wireless communication modules.
- Incorporates high-performance industrial-grade multi-channel LoRaWAN base station radio frequency chips.
- Features an aluminum alloy casing with an IP67 protection rating, ensuring robust protection against environmental factors.
- Supports AC220V power, POE+ Power optional

Stable and Reliable:

- WDT watchdog design ensures system stability.
- Adopts a comprehensive anti-dropout mechanism to ensure data terminals are always online.
- Ethernet interface with built-in 1.5KV electromagnetic isolation protection.
- SIM/UIM card interface with built-in 15KV ESD protection.
- Power interface with reverse polarity protection, overvoltage protection.
- Lightning protection for antenna interface.

1.3 Product Performance Parameters

- Business Channel: Uses a simple star network
- LoRaWAN Protocol Support: Class A, Class B*, Class C
- Operating Frequencies: EU433, CN470-510, CN779-787, EU863-870, US902-928, AU915-928, AS923, KR920-923
- Urban Communication Distance: 9km
- Maximum Transmit Power: 26±1dBm
- Maximum Receive Sensitivity: -142dBm @LoRa
- Communication Bandwidth: 125kHz, 250kHz, 500kHz
- 8 uplink channels, 1 downlink channel
- Implements secure, reliable, low-latency wireless transmission technology
- Communication Rate: Adaptive link rate
- Operating Modes: Supports asynchronous and synchronous frequency transmission
- Positioning Function: GPS, Beidou
- Server Reporting Methods: 4G, 5G, Wired Ethernet
- Wireless Management: WiFi wireless management and upgrades
- Local Storage: Supports TF card local storage
- Operating Temperature: -35~+75℃
- Overall Dimensions: 150*100*240mm
- Waterproof and Dustproof: IP67
- Power Supply: AC220V, POE+
- Total Power Consumption: <8W(under 5G)
- Electrical Performance

Number	Parameters	Technical Specifications
1	Rated Input Voltage	100~240VAC
2	Rated Output Voltage	12V
3	Rated Output Current	1.67A
4	Input Undervoltage Protection	None
5	Output Overvoltage Protection	Yes
6	Output Overcurrent Protection	Yes
7	Short Circuit Protection	Yes
8	Surge Voltage Resistance	Line-line, line-ground both are +-1kV
9	Lightning Protection Level	2KA
10	Input Side Wire Diameter	Recommend 3-5mm
11	POE Power Supply	POE input, Support 10/100 Base-T Adaptive.
12	Supports POE Standards	IEEE802.3af/IEEE802.3at

◆ Power consumption

Average operating voltage V（V）	Average operating current I（mA）	Power Consumption (W)	Note
12.00	TX≦460 RX≦150	6	4G module connected to the Internet with GPS, LoRa communication
12.00	TX≦560 RX≦120	6.7	5G module connected to the Internet with GPS, with LoRa

Note： * Indicates that it is under development.

Chapter 2 Installation

2.1 Overview

FBL800 must be correctly installed to achieve its designed functionality. Typically, the installation of the device should be carried out under the guidance of engineers approved by our company.

➢ *Precautions：*

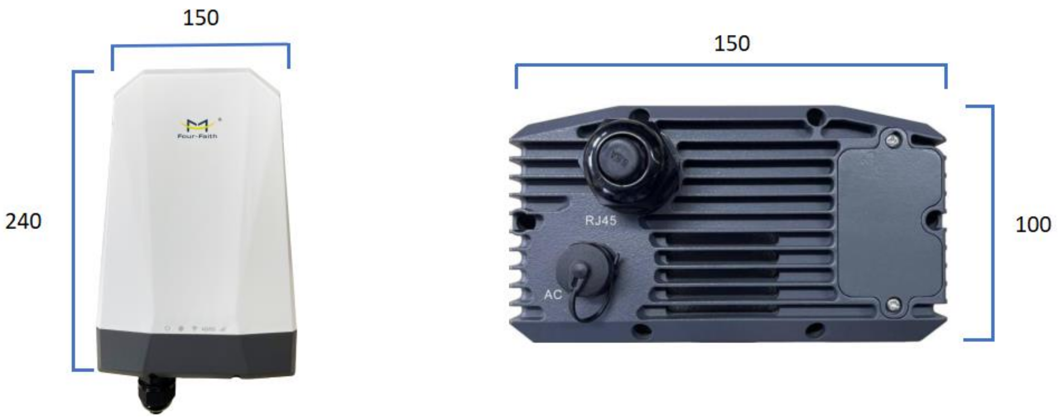
1、Please do not install FBL800 while it is powered.

2、Do not tamper with the plugs, power ports, antenna ports, and other interfaces of FBL800.

2.2 Packing List

Name	Quantity	Note
FBL800 Main Unit	1	
Bracket	2	Wall-mounted and Pole-mounted
Waterproof Rubber Stopper Rods	1	Block the Ethernet Port
Power cord	1	
Countersunk Expansion Screws	1	1 bag of 2 pieces, Wall Mounted
Ethernet Wire	1	
Product certificate	1	Optional
Product warranty card	1	Optional

2.3 Product Dimension

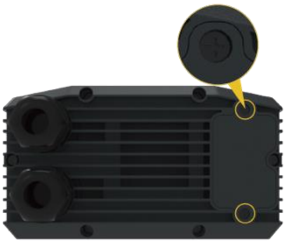


FBL800 Size

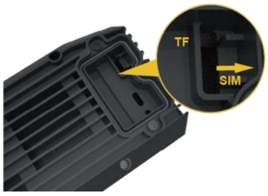
2.3.1 SIM/UIM Card Installation

1、Turn off the device power.

2、Unscrew the two screws of the device cover and remove the cover.



3、Insert the SIM card and TF card according to the directions prompted by the SIM and TF card slots

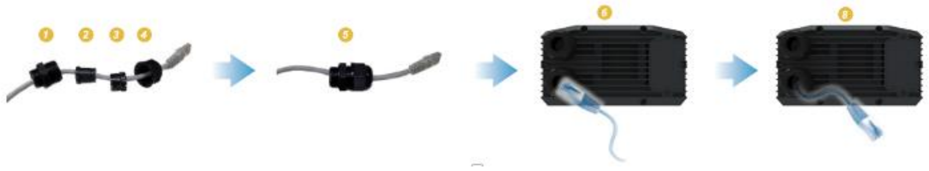


4、Lock the device cover with screws.

2.3.2 Ethernet Cable Installation

Follow the steps in the picture from left to right to insert the network cable into the waterproof connector, tighten the waterproof connector, insert the LAN port of the device, and tighten the waterproof connector with the device

Tip: It is recommended to use a wrench to tighten it, as failure to tighten it may affect the waterproof performance



2.3.3 Installation Method

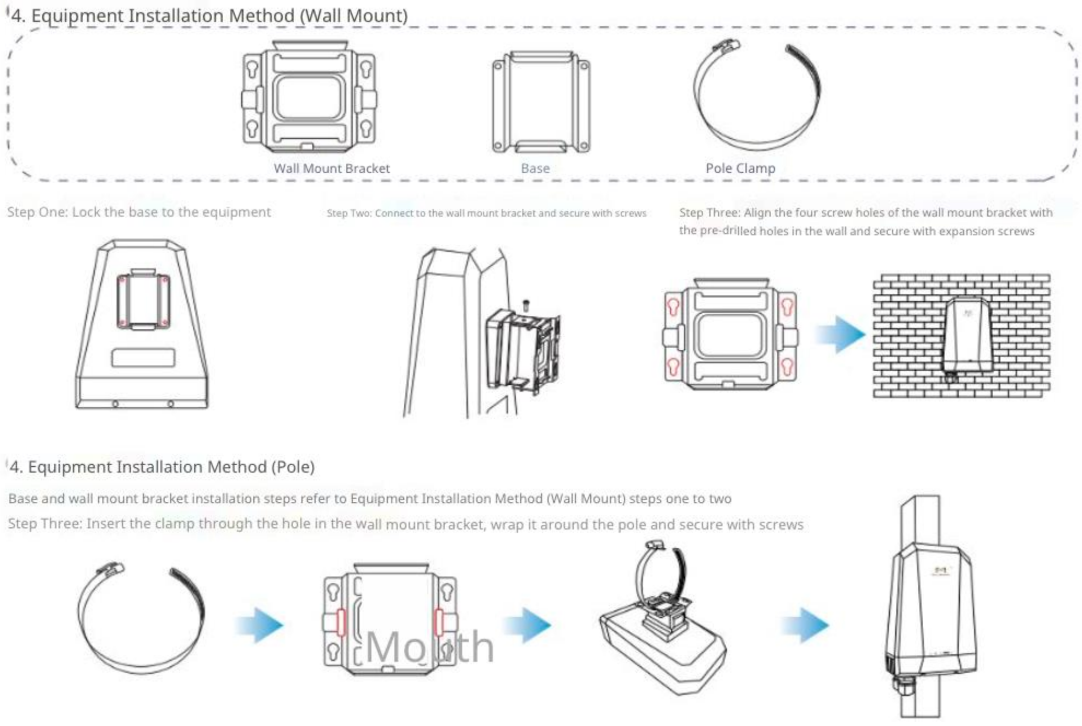
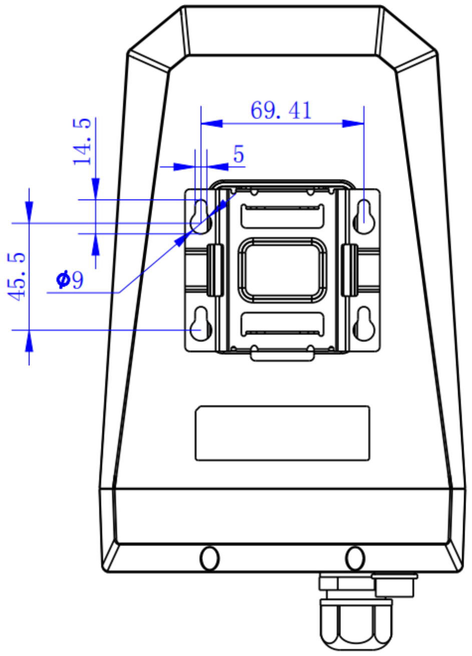
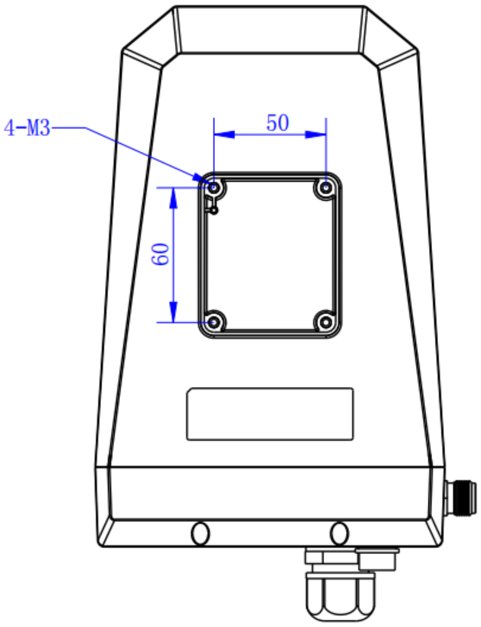

4. Equipment Installation Method (Wall Mount)
Wall Mount Bracket
Base
Pole Clamp
Step One: Lock the base to the equipment
Step Two: Connect to the wall mount bracket and secure with screws
Step Three: Align the four screw holes of the wall mount bracket with the pre-drilled holes in the wall and secure with expansion screws
4. Equipment Installation Method (Pole)
Base and wall mount bracket installation steps refer to Equipment Installation Method (Wall Mount) steps one to two
Step Three: Insert the clamp through the hole in the wall mount bracket, wrap it around the pole and secure with screws

Figure 2.3.6

2.3.4 Installation Drawing



Wall-Mounting Installation Drawing.



No Mounting Installation Drawing

2.4 Indicator Light Instructions

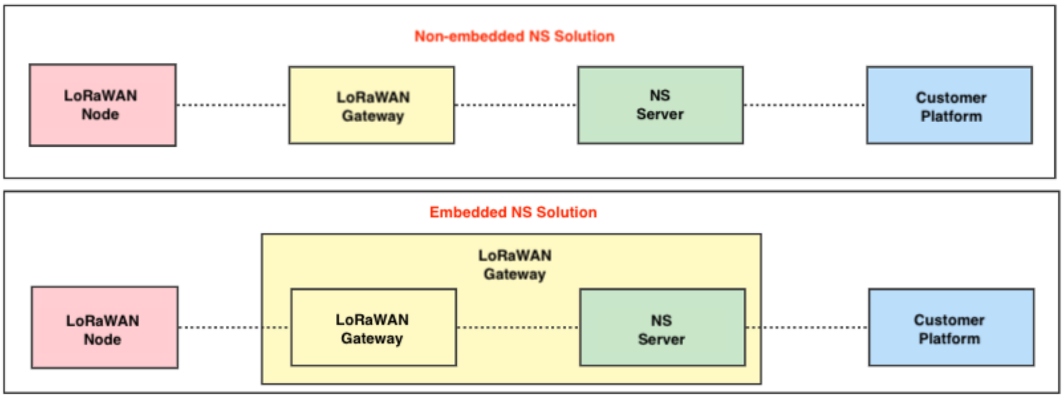
The FBL800 provides the following indicator lights: “Power”, “Online”, “LoRa”, “4G/5G”, “WiFi” The status explanations for each indicator light are as follows:

Lights	Name	Explanation
⏻	Power and System	1. Flashing indicates that the power supply and system are normal; 2. Off means that the power supply is abnormal;
🌐	Network Online	1. Solid light means that the device is successfully connected to the network; 2. Off means that the device is not successfully connected to the network;
LoRa	LoRa	1. Solid light means that the LoRa module is normal 2. Off means that the LoRa module is abnormal
4G/5G	4G/5G	1. Steady light means that 4G/5G networking is successful 2. Off means that the 4G/5G networking is not successful
🛜	WIFI	1. Steady light means WIFI on 2. Off means WIFI off

Chapter 3 Quick Start Guide

3.1 Brief Introduction to the Solution Architecture

3.1.1 Difference Between Embedded and Non-Embedded

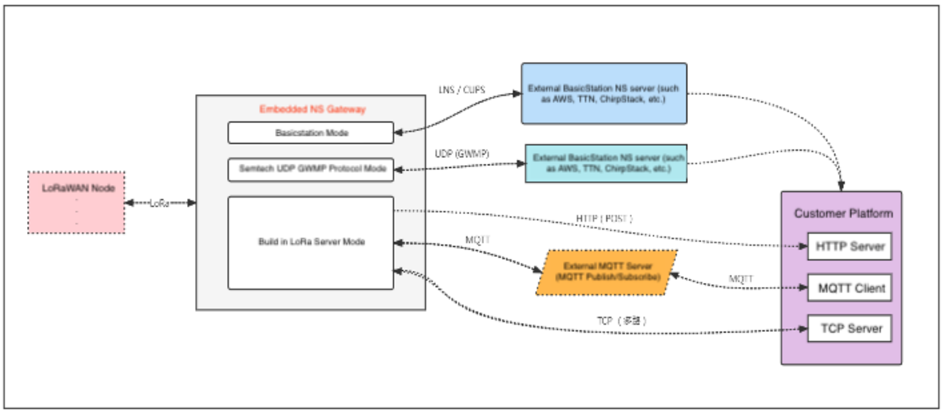


As shown in the above diagram, the main difference between the embedded and non-embedded solutions lies in the position of the NS. In the non-embedded solution, the NS is generally deployed on a server, while in the embedded solution, the NS is deployed within the gateway.

- Embedded Solution (Embedded Mode): The advantage is that there is no need to deploy NS on an external server, reducing operational costs and quickly setting up the entire LoRaWAN system. The drawback is that the performance and storage size of the gateway system are relatively poor compared to a server, limiting the number of nodes and the ability to cache a large amount of information.

- Non-Embedded Solution (External Mode): The advantage is that the server has strong performance, large storage, and can manage a large number of gateways and nodes. It can be deployed in a clustered manner, significantly improving system performance and availability. The drawback is that an additional server is required to deploy NS, which involves maintenance and increases project costs. The setup of the system and the time spent on problem-solving will be more extensive.

3.1.2 System Framework


LoRaWAN Node
LoRa
Embedded NS Gateway
Basicstation Mode
Semtech UDP GWMP Protocol Mode
Build in LoRa Server Mode
LNS / CUPS
External BasicStation NS server (such as AWS, TTN, ChirpStack, etc.)
UDP (GWMP)
External BasicStation NS server (such as AWS, TTN, ChirpStack, etc.)
HTTP (POST)
MQTT
External MQTT Server (MQTT Publish/Subscribe)
MQTT
TCP（多链）
Customer Platform
HTTP Server
MQTT Client
TCP Server

The gateway communicates with devices or terminals, and the data flow direction is determined based on web configuration:

- Basics Station (basicstation mode): In this mode, data will communicate bidirectionally with the corresponding connected server. The gateway serves only as a data forwarding function. Device management, data encryption/decryption, and integration with the customer platform need to be handled on the server side.

- Semtech UDP GWMP Protocol (external NS mode): Data will communicate with an external NS using the standard UDP protocol. In this mode, device management, data encryption/decryption, and client integration will be handled on the external NS server, such as commonly used integration with the Four-Faith cloud NS server.

- Build-in LoRa Server (embedded NS mode): Data will flow to the built-in NS server in the gateway. In this mode, device management, data encryption/decryption, and client integration will be handled on the built-in NS server (LoRa Network Server). Clients can achieve data push functionality through HTTP server configuration (HTTP POST supports only uplink push, not downlink data), or through MQTT and TCP for both uplink and downlink data.

As an embedded NS serving as the core of the LoRaWAN, this product theoretically supports a large number of gateway and node connections. It manages LoRaWAN devices, including network joining, data encryption/decryption, data uplink/downlink, and data push. For uplink data from devices, after LoRaWAN decryption, it establishes a relationship with the client through an interface and sends the uplink data to the customer platform. Clients can send downlink data via MQTT or TCP after encrypting it through LoRaWAN to the specified device.

3.2 Login Configuration Interface

3.2.1 Access the Web Management Platform

1、 Method One: After the gateway is powered on, the default WiFi name is Four-Faith, and there is no default password. After successfully connecting to the WiFi, the gateway's LAN address is default to 192.168.1.1. You can then log in at http://192.168.1.1 (or simply enter 192.168.1.1).

2、 Method Two: If you know the gateway's WAN address (e.g., set to a static IP like 192.168.1.88), you can directly access http://192.168.1.88.

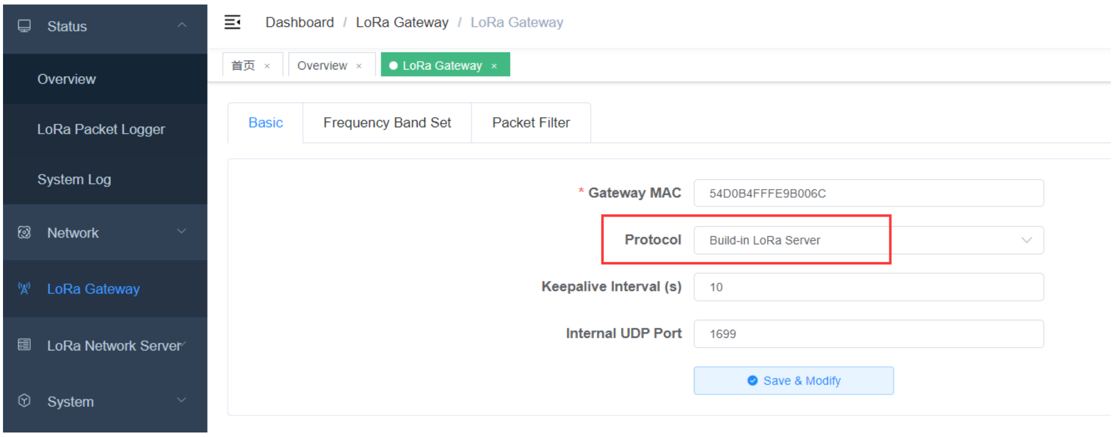
3、 Log in with the default account: admin, and the default password: admin. Click "Login" to access the Web Management Platform.

Note: Please use Google Chrome browser; other browsers may have compatibility issues.

3.2.2 Adding Devices in Embedded Mode

1、 **Identify the device's frequency band and corresponding frequency points** (e.g., for a standard EU868 terminal, frequency points: 868.1MHz, 868.3MHz, 868.5MHz).

2、 **Confirm whether it is in embedded mode** (default is embedded mode). If not, change it to embedded mode.
<Path: LoRa Gateway → Basic>



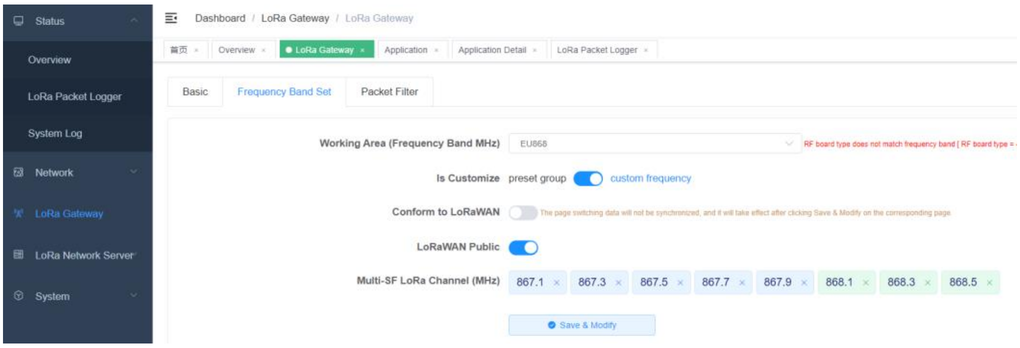
* Gateway MAC: 54D0B4FFFE9B006C

Protocol: Build-in LoRa Server

Keepalive Interval (s): 10

Internal UDP Port: 1699

3、 **Check if the gateway's frequency band and frequency points are consistent** (default frequency points follow the regional parameter specified points). If not, modify them to match.
<Path: LoRa Gateway → Frequency Band Set>



Working Area (Frequency Band MHz): EU868 — RF board type does not match frequency band [RF board type =

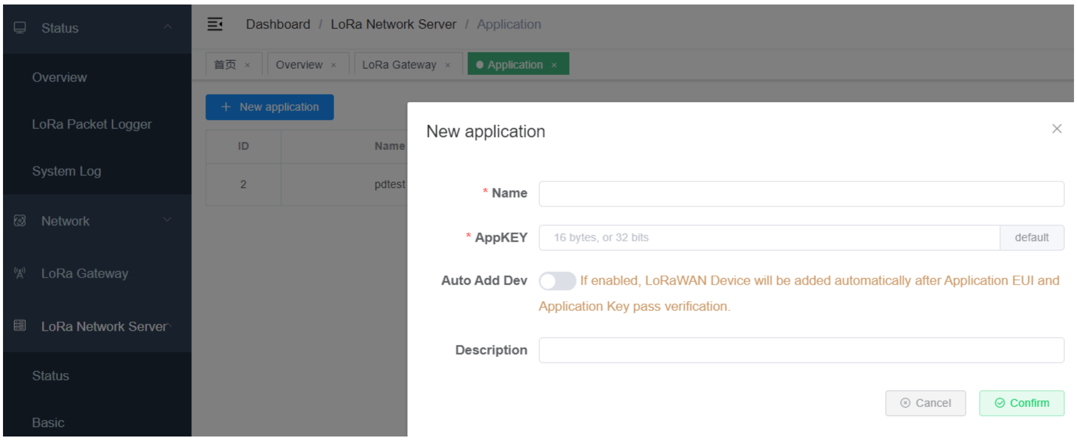
Is Customize: preset group / custom frequency

Conform to LoRaWAN — The page switching data will not be synchronized, and it will take effect after clicking Save & Modify on the corresponding page

LoRaWAN Public

Multi-SF LoRa Channel (MHz): 867.1 × 867.3 × 867.5 × 867.7 × 867.9 × 868.1 × 868.3 × 868.5 ×

4、 **Add an application** (configure it for automatic device addition in network joining mode).
<Path: LoRa Network Server => Application => New application>



New application

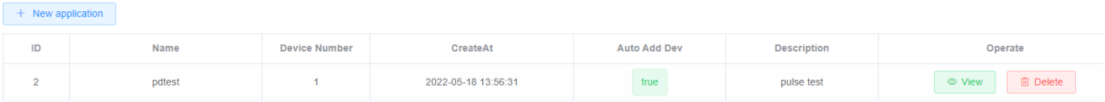
* Name

* AppKEY: 16 bytes, or 32 bits

Auto Add Dev — If enabled, LoRaWAN Device will be added automatically after Application EUI and Application Key pass verification.

Description

In the above figure, both AppKEY and AppEUI are automatically generated by clicking on the "default" on the right side (this value is the default value for Four-Faith; for devices from other manufacturers, modify it to the corresponding values). Choose ClassA or ClassC based on the device type, then click "Confirm" to add. After adding, the following page will appear:



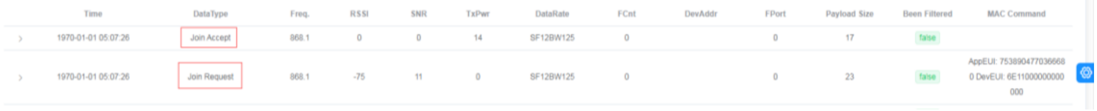
ID	Name	Device Number	CreateAt	Auto Add Dev	Description	Operate
2	pdtest	1	2022-05-18 13:56:31	true	pulse test	View Delete

5、 Device Network Joining

The device initiates a network joining request, and check whether the network joining is successful. If it cannot join the network successfully, troubleshoot as follows:

Confirm whether the gateway can receive the network joining request from the device (you can use a packet capture tool, path: Status → LoRa Message Logger).

If the gateway can receive the network joining request but does not see the network joining response (Join Accept), it is generally caused by the inconsistency between the AppKey or AppEUI configured in the application and that of the device.

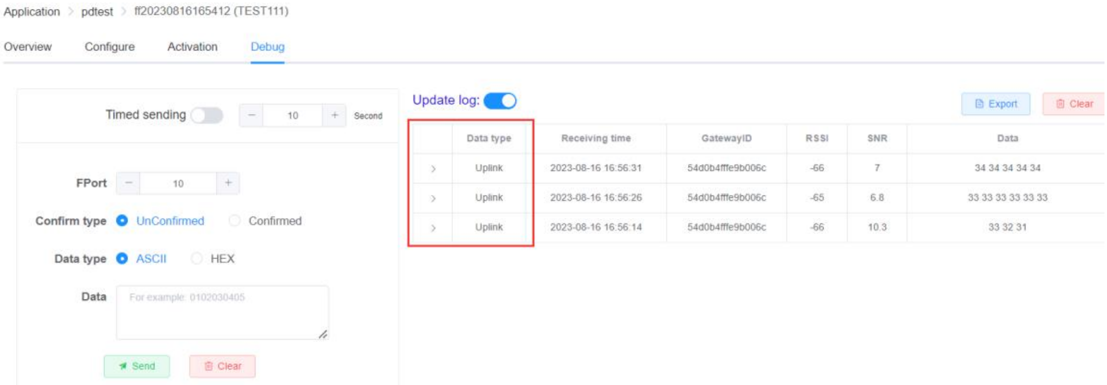


	Time	DataType	Freq.	RSSI	SNR	TxPwr	DataRate	FCnt	DevAddr	FPort	Payload Size	Been Filtered	MAC Command
›	1970-01-01 05:07:26	Join Accept	868.1	0	0	14	SF12BW125	0		0	17	false	
›	1970-01-01 05:07:26	Join Request	868.1	-75	11	0	SF12BW125	0		0	23	false	AppEUI: 7538904770366680 DevEUI: 6E11000000000000

6、 Device Uplink Data

After the device successfully joins the network, instruct the device to report any data. At this time, in the application's device list, find the corresponding device for viewing.

<Path: LoRa Network Server → Applications → Corresponding Application (click to view) → Find the corresponding device (click to view) → Online Debugging>



Timed sending: off, 10 Second

FPort: 10

Confirm type: ● UnConfirmed ○ Confirmed

Data type: ● ASCII ○ HEX

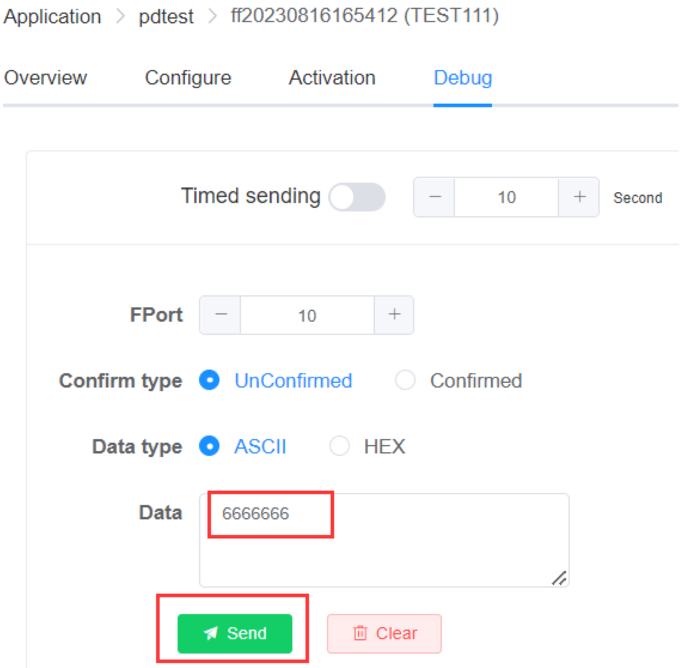
Data: For example: 0102030405

Update log: on

	Data type	Receiving time	GatewayID	RSSI	SNR	Data
›	Uplink	2023-08-16 16:56:31	54d0b4fffe9b006c	-66	7	34 34 34 34 34
›	Uplink	2023-08-16 16:56:26	54d0b4fffe9b006c	-65	6.8	33 33 33 33 33 33
›	Uplink	2023-08-16 16:56:14	54d0b4fffe9b006c	-66	10.3	33 32 31

7、 Send Data to the Device

Send data to the device on the device's online debugging page, as shown in the following figure:



Timed sending: off, 10 Second

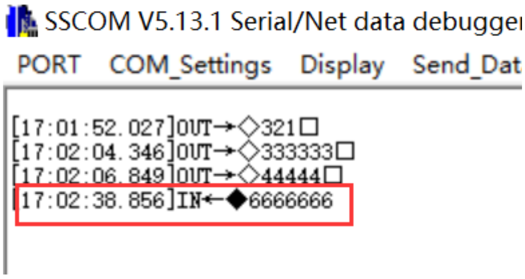
FPort: 10

Confirm type: ● UnConfirmed ○ Confirmed

Data type: ● ASCII ○ HEX

Data: 6666666

The Four-Faith module receives data as follows:



```
[17:01:52.027]OUT→◇321□
[17:02:04.346]OUT→◇333333□
[17:02:06.849]OUT→◇44444□
[17:02:38.856]IN←◆6666666
```

Attention:

The device types are divided into ClassA and ClassC, and the data reception methods are as follows:

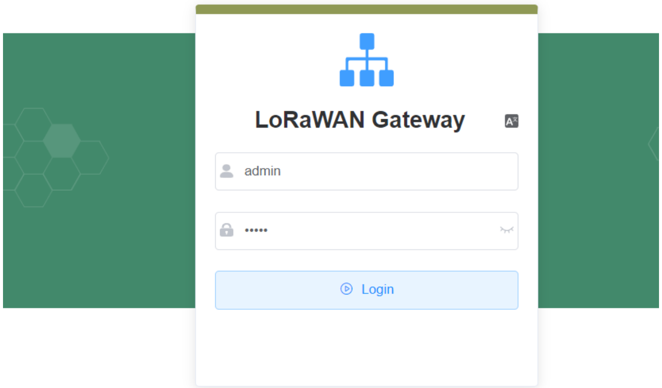
- In ClassA mode, after sending data, it will not be directly sent to the device. You need to wait for the device to send uplink data again before the sent data is delivered to the device.
- In ClassC mode, data sent will be directly delivered to the device. If the device does not receive the data, please check whether the type configured in NS matches the type configured for the device. If not, after modification, you need to rejoin the network before conducting data communication tests.

Chapter 4 Detailed Introduction to the Function Pages

4.1 Interface Management Configuration

4.1.1 Web Management Platform

- Method One: After the gateway is powered on, the default WiFi name is Four-Faith, and there is no default password. After successfully connecting to the WiFi, the gateway's LAN address is default to 192.168.1.1. You can then log in at http://192.168.1.1 (or simply enter 192.168.1.1).
- Method Two: If you know the gateway's WAN address (e.g., set to a static IP like 192.168.1.88), you can directly access http://192.168.1.88.
- Log in with the default account: admin, and the default password: admin. Click "Login" to enter the Web Management Platform.



Note: Please use Google Chrome browser; other browsers may have compatibility issues.

4.1.2 Directory Details

Here is an introduction to the functionality of each page based on the directory order:

- **Status**

➢ **Overview:** Displays statistics for data monitored by the gateway and system parameter information.

➢ **LoRa Message Logger:** Shows received data and downstream data for the gateway.

- **System Logs:** Records logs of the gateway's operational processes.

- **Network**
- **WAN Interface:** Configures the gateway's WAN settings. Network information, such as DHCP or static IP, can be configured here.
- **Wi-Fi:** Configures WiFi parameters and security settings.
- **Network Diagnostics:** Includes commands like Ping, Traceroute, Nslookup.
- **Firewall:** Configures basic parameters for the firewall.

- **LoRa Gateway**
- Configuration of gateway mode, frequency point parameters, packet filtering, etc.

- **LoRa Network Server**
- **Status:** Displays statistics for the embedded NS.
- **Basic Settings:** Configures NS-related parameters such as ADR switch, RX2 parameter settings, etc.
- **Gateway:** Displays gateway information.
- **Applications:** Displays application information, including device lists.
- **Multicast:** Manages multicast.
- **Interfaces:** Configures protocol types for customer platform integration, data conversion, heartbeat settings, etc.

- **System**
- **System:** Displays embedded NS version information, system time settings, etc.
- **Change Password:** Modifies the Web Management Platform password.
- **Restart:** Restarts the gateway.
- **Factory Reset:** Performs a factory reset.

4.1.3 Management Configuration

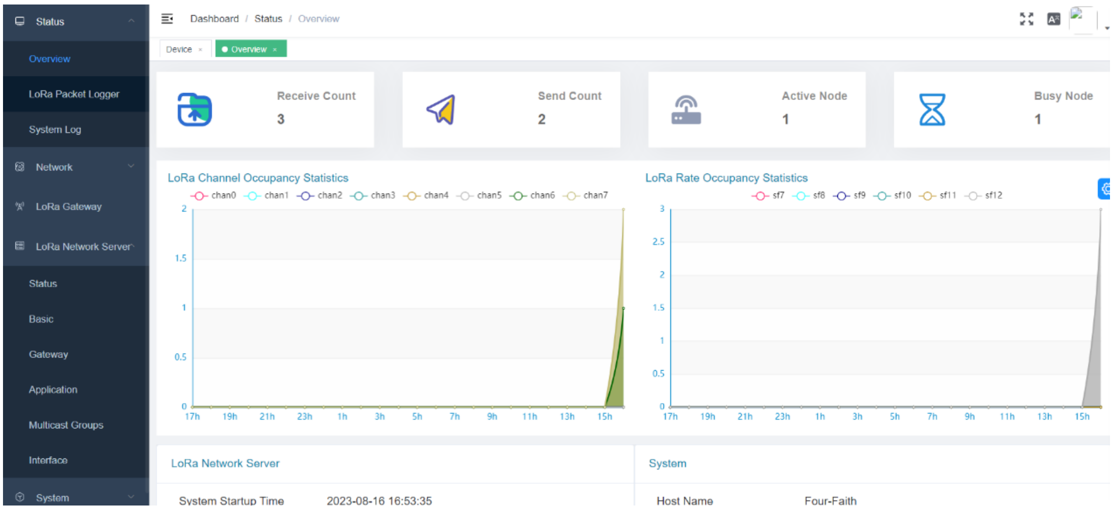
4.1.3.1 Status

1. Overview

- **Path:** Status → Overview
- **Function:** Displays communication statistics for the gateway, making it convenient to analyze the RF environment around the gateway. This helps in assessing device communication status or identifying the presence of interfering devices.
- **Details:**
  - **Received Messages:** Number of messages received since system startup.
  - **Sent Messages:** Number of messages sent since system startup.
  - **Active Nodes:** Number of upstream nodes received by the gateway.

- ➢ **Busy Nodes:** Nodes that have transmitted twice within 10 seconds are considered busy nodes. This section provides a count within the last hour.
- ➢ **LoRa Channel Occupancy Statistics:** Channel occupancy details for various time periods in the last 24 hours.
- ➢ **LoRa Rate Occupancy Statistics:** Rate occupancy details for various time periods in the last 24 hours.
- ➢ **LoRa Network Server:** Includes system startup time, LoRa protocol details, device count, NS device uplink count, NS device downlink count, and NS MQTT connection status.
- ➢ **System:** Includes host name, LAN MAC, WAN MAC, wireless MAC, WAN IP, LAN IP, and WAN protocol.
- ➢ **Wireless:** Includes wireless switch status, mode, network mode, name, channel, and transmission power.

- **Preview:**


Status
Overview
LoRa Packet Logger
System Log
Network
LoRa Gateway
LoRa Network Server
Status
Basic
Gateway
Application
Multicast Groups
Interface
System
Dashboard / Status / Overview
Device ×
Overview ×
LoRa Channel Occupancy Statistics
chan0 chan1 chan2 chan3 chan4 chan5 chan6 chan7
2
1.5
1
0.5
0
17h 19h 21h 23h 1h 3h 5h 7h 9h 11h 13h 15h
LoRa Rate Occupancy Statistics
sf7 sf8 sf9 sf10 sf11 sf12
3
2.5
2
1.5
1
0.5
0
17h 19h 21h 23h 1h 3h 5h 7h 9h 11h 13h 15h
LoRa Network Server
System

Receive Count 3

Send Count 2

Active Node 1

Busy Node 1

System Startup Time 2023-08-16 16:53:35

Host Name Four-Faith

2. LoRa Message Logger

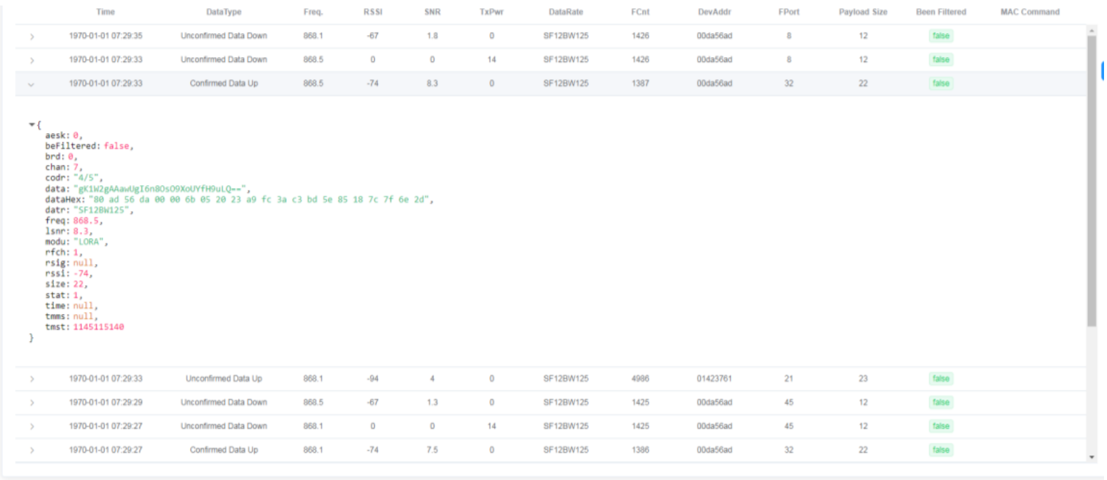
- **Path:** Status → LoRa Message Logger

- **Function:**
  - ➢ Displays LoRaWAN data received and sent by the gateway.
  - ➢ Useful for analyzing communication between the gateway and devices, allowing for the identification of issues such as unanswered join requests, missing downlink data, and communication quality.

- **Details:**
  - ➢ **Update Log Switch:** Defaulted to ON; when turned off, data is expanded for viewing, and during this period, data is received but not listed. When turned on again, the data is automatically updated to the list.
  - ➢ **LoRaWAN Data Type Selection:** Facilitates analysis of communication issues based on data type.

- **Packet Filtering Status:** Indicates whether the packet is filtered. Filtered data won't be reported to the NS server. Filter configurations can be found in LoRa Gateway → Packet Filter. Options include:
  - 0: All - Show both filtered and unfiltered data
  - UnFiltered - Show only unfiltered data
  - BeFiltered - Show only filtered data
- **devAddr:** Search by short address.
- **Time:** Timestamp of data reception.
- **DataType:** Data type, including:
  - ALL
  - Join Request
  - Join Accept
  - Unconfirmed Data Up
  - Unconfirmed Data Down
  - Confirmed Data Up
  - Confirmed Data Down
- **Freq:** Communication frequency.
- **RSSI:** Signal strength.
- **SNR:** Signal-to-noise ratio.
- **TxPwr:** Transmission power. This is 0 for uplink data.
- **FCnt:** Frame counter, useful for determining packet loss or retransmission.

- **Preview:**



	Time	DataType	Freq.	RSSI	SNR	TxPwr	DataRate	FCnt	DevAddr	FPort	Payload Size	Been Filtered	MAC Command
›	1970-01-01 07:29:35	Unconfirmed Data Down	868.1	-67	1.8	0	SF12BW125	1426	00da56ad	8	12	false	
›	1970-01-01 07:29:33	Unconfirmed Data Down	868.5	0	0	14	SF12BW125	1426	00da56ad	8	12	false	
⌄	1970-01-01 07:29:33	Confirmed Data Up	868.5	-74	8.3	0	SF12BW125	1387	00da56ad	32	22	false	

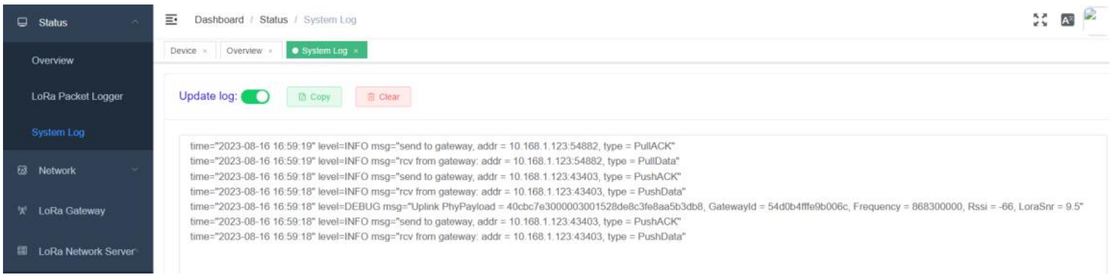
```
▾{
   aesk: 0,
   beFiltered: false,
   brd: 0,
   chan: 7,
   codr: "4/5",
   data: "gK1W2gAAawUgI6n8OsO9XoUYfH9uLQ==",
   dataHex: "80 ad 56 da 00 00 6b 05 20 23 a9 fc 3a c3 bd 5e 85 18 7c 7f 6e 2d",
   datr: "SF12BW125",
   freq: 868.5,
   lsnr: 8.3,
   modu: "LORA",
   rfch: 1,
   rsig: null,
   rssi: -74,
   size: 22,
   stat: 1,
   time: null,
   tmms: null,
   tmst: 1145115140
}
```

	Time	DataType	Freq.	RSSI	SNR	TxPwr	DataRate	FCnt	DevAddr	FPort	Payload Size	Been Filtered	MAC Command
›	1970-01-01 07:29:33	Unconfirmed Data Up	868.1	-94	4	0	SF12BW125	4986	01423761	21	23	false	
›	1970-01-01 07:29:29	Unconfirmed Data Down	868.5	-67	1.3	0	SF12BW125	1425	00da56ad	45	12	false	
›	1970-01-01 07:29:27	Unconfirmed Data Down	868.1	0	0	14	SF12BW125	1425	00da56ad	45	12	false	
›	1970-01-01 07:29:27	Confirmed Data Up	868.1	-74	7.5	0	SF12BW125	1386	00da56ad	32	22	false	

3. LoRa Message Logger

- **Path:** Status → LoRa Message Logger
- **Function:**The log is useful for analyzing the overall operation of the gateway, abnormal device communication, and other anomalies.
- **Details:**
  - **Switch:** Defaulted to ON; when paused, new data is stored in the browser cache and updates upon reactivation.

- **Preview:**



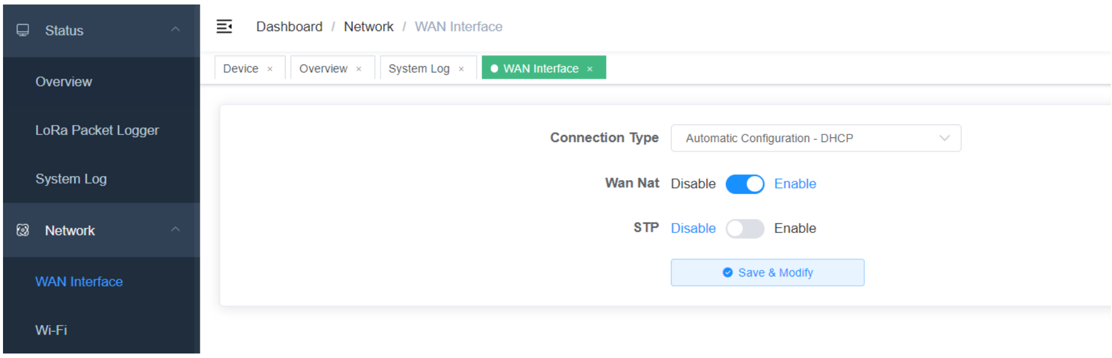
Update log: ●

```
time="2023-08-16 16:59:19" level=INFO msg="send to gateway, addr = 10.168.1.123:54882, type = PullACK"
time="2023-08-16 16:59:19" level=INFO msg="rcv from gateway: addr = 10.168.1.123:54882, type = PullData"
time="2023-08-16 16:59:18" level=INFO msg="send to gateway, addr = 10.168.1.123:43403, type = PushACK"
time="2023-08-16 16:59:18" level=INFO msg="rcv from gateway: addr = 10.168.1.123:43403, type = PushData"
time="2023-08-16 16:59:18" level=DEBUG msg="Uplink PhyPayload = 40cbc7e3000003001528de8c3fe8aa5b3db8, GatewayId = 54d0b4fffe9b006c, Frequency = 868300000, Rssi = -66, LoraSnr = 9.5"
time="2023-08-16 16:59:18" level=INFO msg="send to gateway, addr = 10.168.1.123:43403, type = PushACK"
time="2023-08-16 16:59:18" level=INFO msg="rcv from gateway: addr = 10.168.1.123:43403, type = PushData"
```

4.1.3.2 Network

1. Network

- **Path:** Network → WAN Interface
- **Function:** Used to configure network parameters, such as setting up static IP or DHCP.
- **Details:**
  - ➢ Configure various modes based on mode parameters.
- **Preview:**:



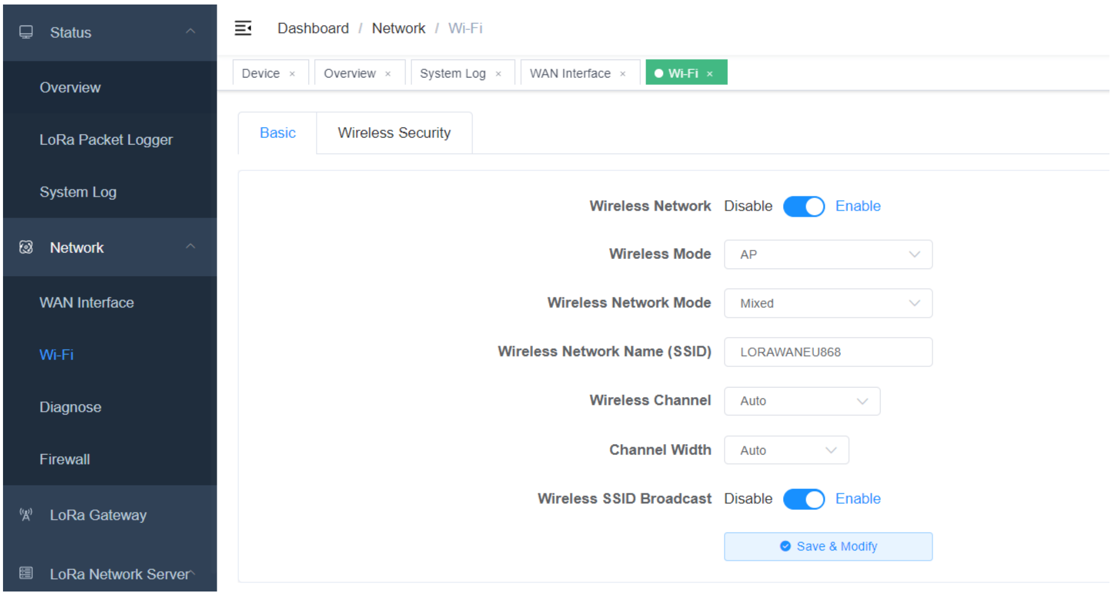
Connection Type: Automatic Configuration - DHCP

Wan Nat: Disable [✓ Enable]

STP: [✓ Disable] Enable

2. WiFi

- **Path:** Network → WiFi
- **Function:** Configures WiFi parameters and security settings.
- **Details:**
  - ➢ Configure various modes based on mode parameters.
- **Preview:**:



Wireless Network: Disable ● Enable

Wireless Mode: AP

Wireless Network Mode: Mixed

Wireless Network Name (SSID): LORAWANEU868

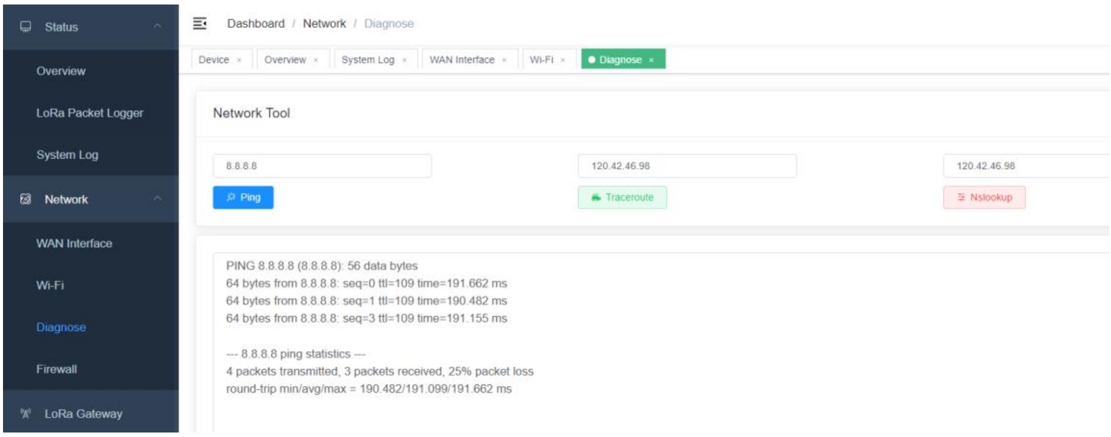
Wireless Channel: Auto

Channel Width: Auto

Wireless SSID Broadcast: Disable ● Enable

3. Network Diagnostics

- **Path:** Network → Network Diagnostics
- **Function:** Supports Ping, Traceroute, and NsLookup commands.
- **Details:**
  - **Ping:** A program used to test network connectivity.
  - **Traceroute:** A command that uses ICMP protocol to locate all routers between your computer and the target computer.
  - **NsLookup:** A command-line tool to monitor whether DNS servers in the network can correctly perform domain name resolution.
- **Preview:**



8.8.8.8

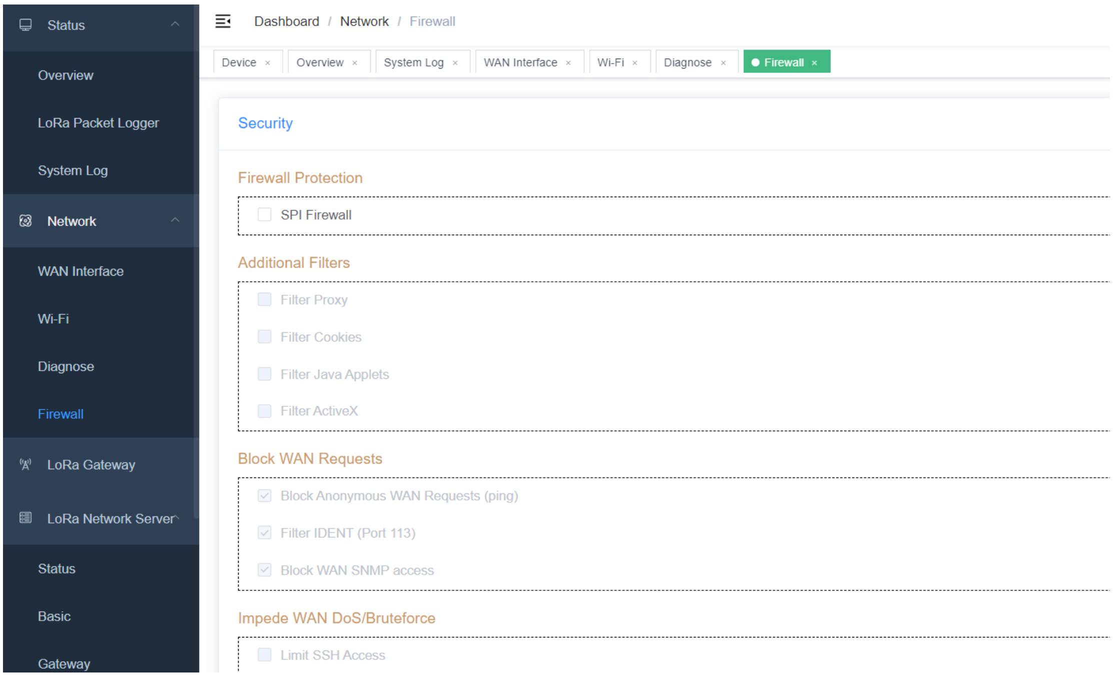
120.42.46.98

120.42.46.98

```
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: seq=0 ttl=109 time=191.662 ms
64 bytes from 8.8.8.8: seq=1 ttl=109 time=190.482 ms
64 bytes from 8.8.8.8: seq=3 ttl=109 time=191.155 ms

--- 8.8.8.8 ping statistics ---
4 packets transmitted, 3 packets received, 25% packet loss
round-trip min/avg/max = 190.482/191.099/191.662 ms
```

4. Firewall

- **Path:** Network → Firewall
- **Function:** Configuration of firewall-related parameters.
- **Details:**
  - Configure parameters based on the displayed page.
- **Preview:**



Security

Firewall Protection

☐ SPI Firewall

Additional Filters

☐ Filter Proxy

☐ Filter Cookies

☐ Filter Java Applets

☐ Filter ActiveX

Block WAN Requests

☑ Block Anonymous WAN Requests (ping)

☑ Filter IDENT (Port 113)

☑ Block WAN SNMP access

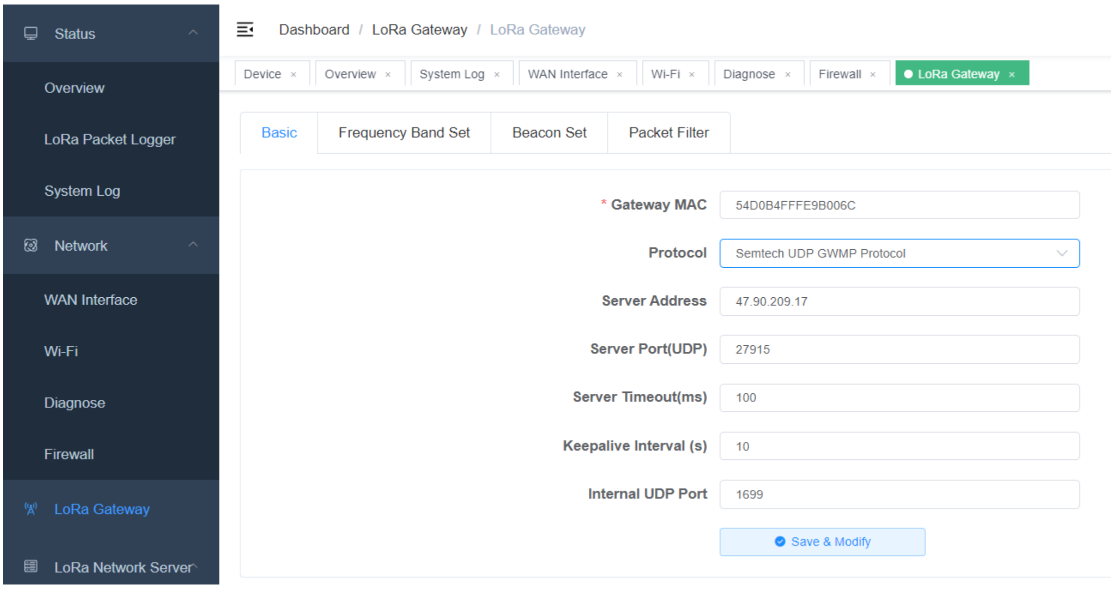
Impede WAN DoS/Bruteforce

☐ Limit SSH Access

4.1.3.3 LoRa Gateway

1. Basic Settings

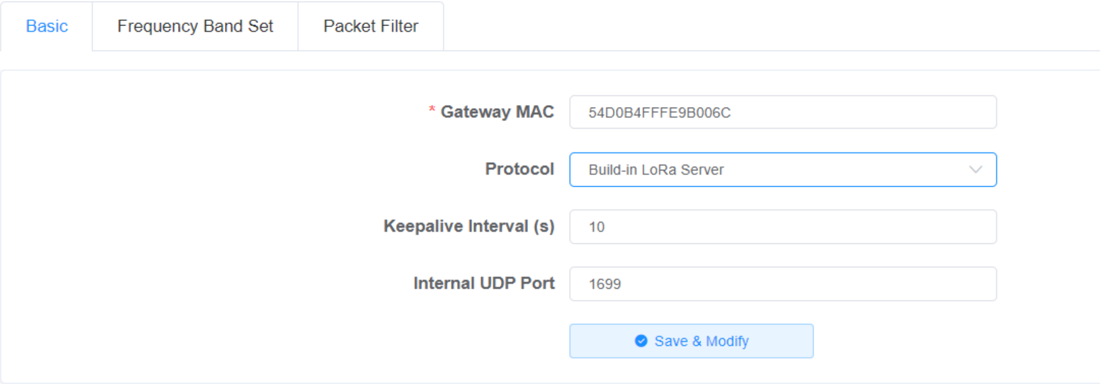
- **Path:** LoRa Gateway → Basic Settings
- **Function:**Configuration of gateway protocols, allowing for settings such as Build-in LoRa Server, Semtech UDP GWMP Protocol, and Basics Station modes.
- **Details:**
  - **Semtech UDP GWMP Protocol** - GWMP Forwarding Mode
    - Gateway MAC: Unique identifier for the gateway, 8 bytes in length (16 bits), typically not modified.
    - Protocol: UDP GWMP protocol, connecting to an external NS server, with the gateway acting as a data forwarding role.
    - Server Address: IP or domain name.
    - Server Port: Port number (e.g., 1700).
    - Server Timeout (ms): Timeout duration for waiting for responses to data reporting, typically not modified.
    - Keepalive Interval (s): Interval for the pull_data command in the protocol, typically not modified.
    - Internal UDP Communication Port: In a cascaded application where the gateway acts as a server, this port number is configured as the server port for the sub-gateway.



* Gateway MAC	54D0B4FFFE9B006C
Protocol	Semtech UDP GWMP Protocol
Server Address	47.90.209.17
Server Port(UDP)	27915
Server Timeout(ms)	100
Keepalive Interval (s)	10
Internal UDP Port	1699

➢ **Build-in LoRa Server** - Internal NS Mode

- Gateway MAC: Unique identifier for the gateway, 8 bytes in length (16 bits), typically not modified.
- Protocol: Internal NS mode, equivalent to having the NS deployed within the gateway.
- Keepalive Interval (s): Interval for the pull_data command in the protocol, typically not modified.
- Internal UDP Communication Port: In a cascaded application where the gateway acts as a server, this port number is configured as the server port for the sub-gateway.

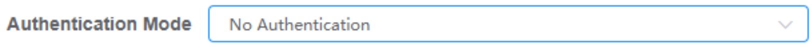


* Gateway MAC	54D0B4FFFE9B006C
Protocol	Build-in LoRa Server
Keepalive Interval (s)	10
Internal UDP Port	1699

➢ **Basics Station** - More Secure and Reliable Protocol (Connected to NS via WebSocket or HTTP)

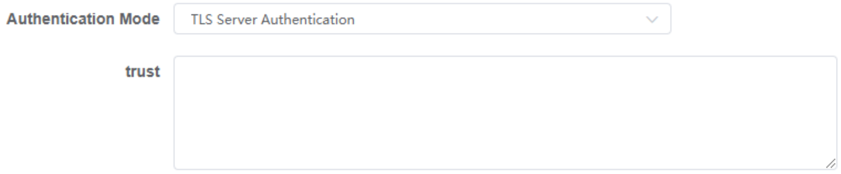
- **Gateway MAC:** Unique identifier for the gateway, 8 bytes in length (16 bits), typically not modified.
- **Protocol:** Basicstation mode.
- **Server:** LNS protocol (select this option for regular data communication) or CUPS protocol (adds gateway upgrade-related protocols).
- **URI:** Server address for connection (IP or domain name).

- **Port:** Corresponding port for the server.
- **Authentication Mode:** Security authentication mode (detailed scenarios for each mode will be explained when connecting to the platform). The following are brief introductions to each mode:
  - **No Authentication:** Establishes a regular WebSocket or HTTP connection without requiring identity verification (e.g., ChirpStack is configured in this mode for integration with the TTN platform).



Authentication Mode: No Authentication

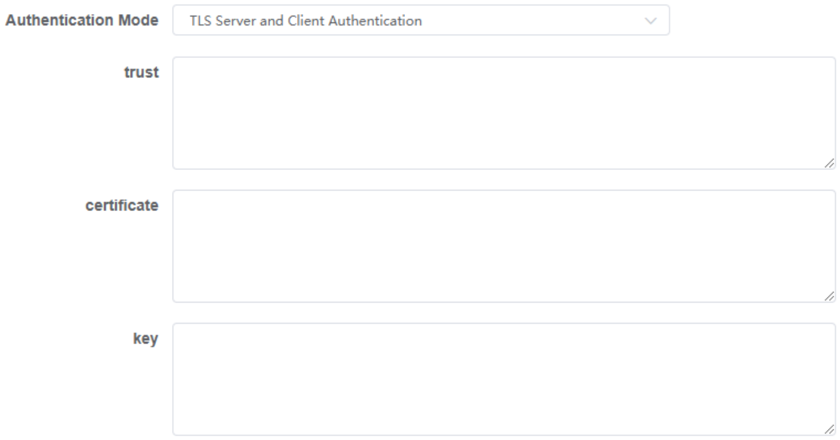
  - **TLS Server Authentication:** Authenticates the server (LNS or CUPS) through establishing a TLS connection (wss, https). （e.g. ChirpStack is configured in this mode)



Authentication Mode: TLS Server Authentication

trust

  - **TLS Server and Client Authentication:** Authenticates both the server (LNS or CUPS) and the client (gateway) through establishing a TLS connection (wss, https). The gateway authenticates the server by verifying its certificate, and the server authenticates the gateway by requesting its certificate along with a signature using a private key. (e.g. This mode is used when interfacing with the AWS platform.)



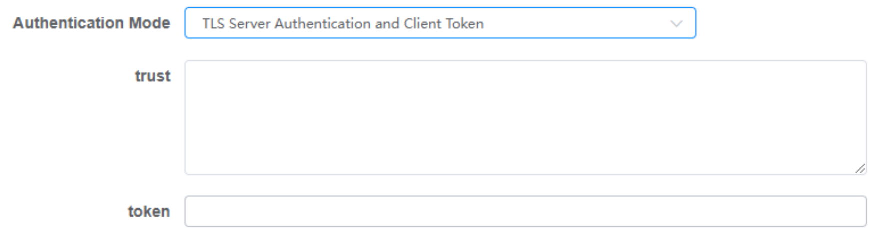
Authentication Mode: TLS Server and Client Authentication

trust

certificate

key

  - **TLS Server Authentication and Client Token:** In this mode, the gateway authenticates the server (LNS or CUPS) by establishing a TLS connection (wss, https). The server, on the other hand, verifies the identity of the gateway by examining the security token provided by the gateway.(e.g. This mode is used when interfacing with The Things Network (TTN) platform.)

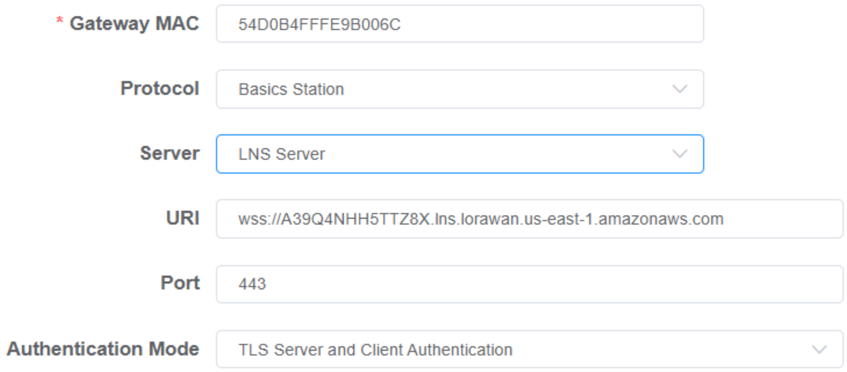


Authentication Mode: TLS Server Authentication and Client Token

trust:

token:

- **Preview:**



* Gateway MAC: 54D0B4FFFE9B006C

Protocol: Basics Station

Server: LNS Server

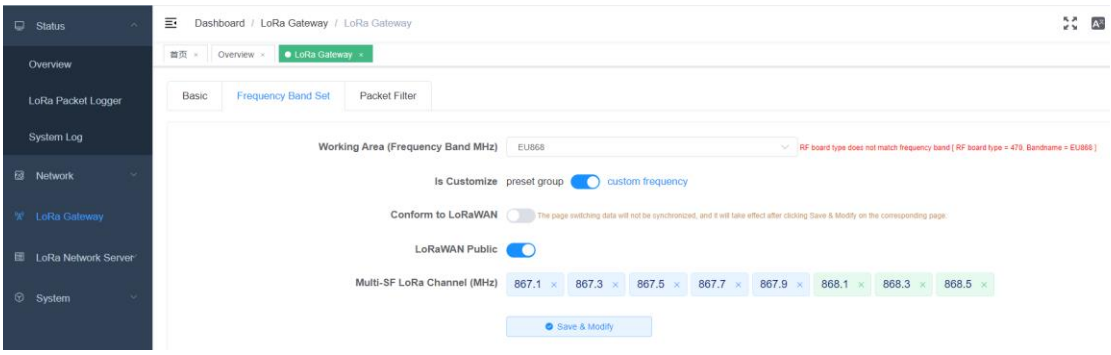
URI: wss://A39Q4NHH5TTZ8X.lns.lorawan.us-east-1.amazonaws.com

Port: 443

Authentication Mode: TLS Server and Client Authentication

2. Frequency Configuration

- **Path:** LoRa Gateway → Frequency Configuration
- **Function:** Configuration of gateway frequencies, supporting modes such as Semtech UDP GWMP Protocol or Build-in LoRa Server. For Basics Station mode, frequency settings are configured on the NS server.
- **Details:**
  - Frequency configuration supports three main methods.
    - **Custom Frequency Method:** This method provides a simple and intuitive view of allocated frequencies. In the example below, frequencies on the left (e.g., 867.1) can be deleted, while those on the right (e.g., 868.1) are essential and cannot be removed. Clicking the '×' next to a frequency deletes it, and clicking the '+ Add' on the far right adds a new frequency.



Working Area (Frequency Band MHz): EU868 RF board type does not match frequency band [RF board type = 470, Bandname = EU868]

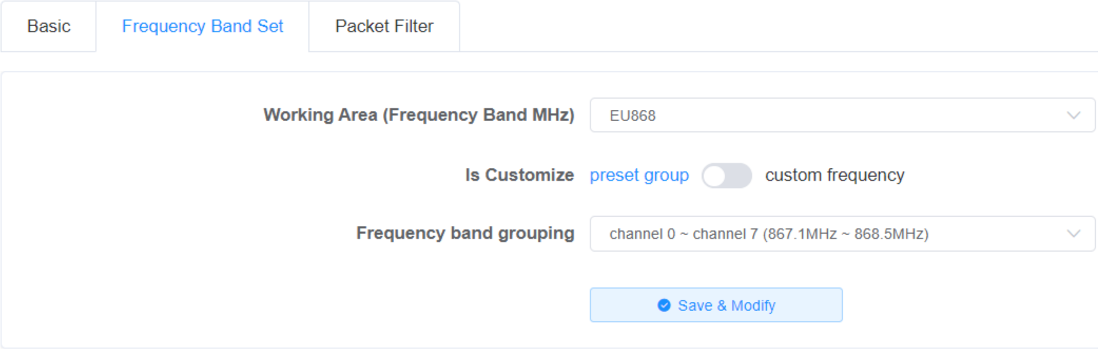
Is Customize: preset group / custom frequency

Conform to LoRaWAN: The page switching data will not be synchronized, and it will take effect after clicking Save & Modify on the corresponding page.

LoRaWAN Public

Multi-SF LoRa Channel (MHz): 867.1 × 867.3 × 867.5 × 867.7 × 867.9 × 868.1 × 868.3 × 868.5 ×

    - **Pre-set Group Method:** This method is the most convenient. Choose the corresponding group as needed, and it will display the starting and ending frequencies. Typically, there is a 0.2MHz interval between each, totaling 8 frequencies.

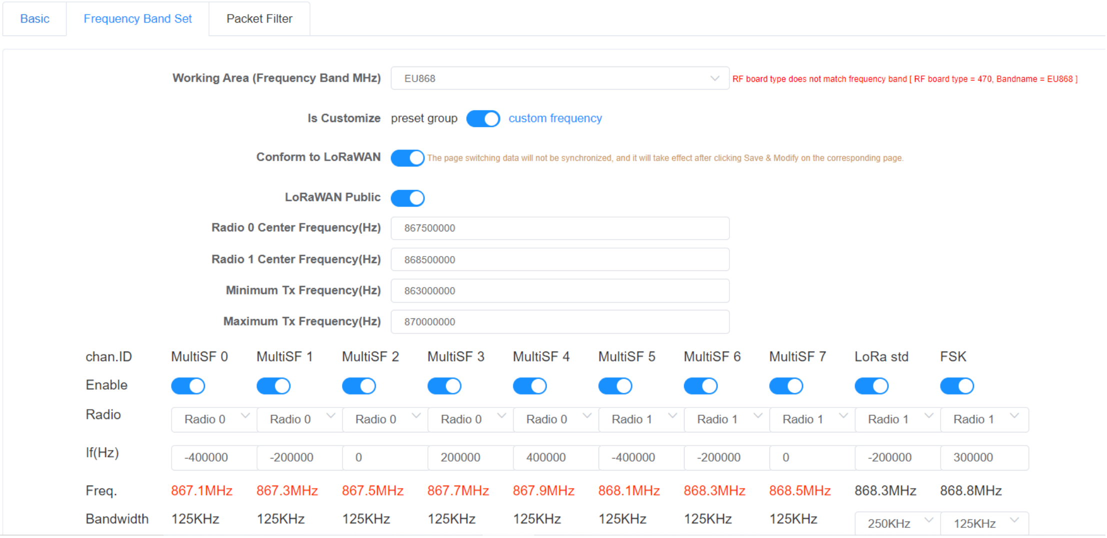


Working Area (Frequency Band MHz): EU868

Is Customize: preset group ○ custom frequency

Frequency band grouping: channel 0 ~ channel 7 (867.1MHz ~ 868.5MHz)

❖ **Custom Frequency + Conform to LoRaWAN Method:** This method is the most in line with the gateway configuration file structure and is the most comprehensive configuration method. Use this method when the other two methods cannot meet the requirements.



Working Area (Frequency Band MHz): EU868 RF board type does not match frequency band [RF board type = 470, Bandname = EU868]

Is Customize: preset group ● custom frequency

Conform to LoRaWAN: ● The page switching data will not be synchronized, and it will take effect after clicking Save & Modify on the corresponding page.

LoRaWAN Public: ●

Radio 0 Center Frequency(Hz): 867500000

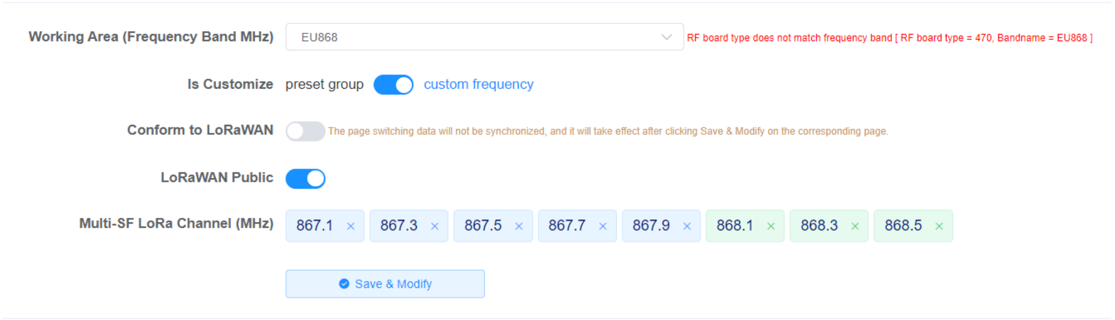
Radio 1 Center Frequency(Hz): 868500000

Minimum Tx Frequency(Hz): 863000000

Maximum Tx Frequency(Hz): 870000000

chan.ID	MultiSF 0	MultiSF 1	MultiSF 2	MultiSF 3	MultiSF 4	MultiSF 5	MultiSF 6	MultiSF 7	LoRa std	FSK
Enable	●	●	●	●	●	●	●	●	●	●
Radio	Radio 0	Radio 0	Radio 0	Radio 0	Radio 0	Radio 1	Radio 1	Radio 1	Radio 1	Radio 1
If(Hz)	-400000	-200000	0	200000	400000	-400000	-200000	0	-200000	300000
Freq.	867.1MHz	867.3MHz	867.5MHz	867.7MHz	867.9MHz	868.1MHz	868.3MHz	868.5MHz	868.3MHz	868.8MHz
Bandwidth	125KHz	125KHz	125KHz	125KHz	125KHz	125KHz	125KHz	125KHz	250KHz	125KHz

- **Preview:**



Working Area (Frequency Band MHz): EU868 RF board type does not match frequency band [RF board type = 470, Bandname = EU868]

Is Customize: preset group ● custom frequency

Conform to LoRaWAN: ○ The page switching data will not be synchronized, and it will take effect after clicking Save & Modify on the corresponding page.

LoRaWAN Public: ●

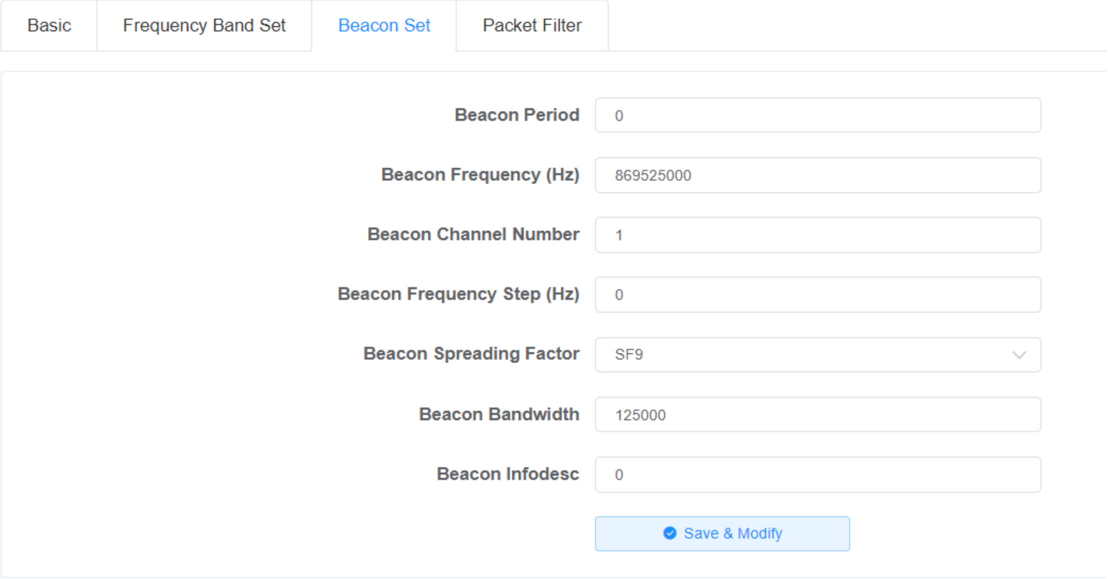
Multi-SF LoRa Channel (MHz): 867.1 × 867.3 × 867.5 × 867.7 × 867.9 × 868.1 × 868.3 × 868.5 ×

3. Beacon Set

- **Path:** LoRa Gateway → Beacon Set
- **Function:** Configure the gateway's ClassB parameters, available in Semtech UDP GWMP Protocol mode.
- **Details:**
  - ➢ **Beacon Period:** Period, set to 0 means it is turned off.

- **Beacon Frequency (Hz):** Frequency point.
- **Beacon Spreading Factor:** Spreading factor.
- **Beacon Bandwidth:** Beacon packet bandwidth.
- **Beacon Tx Power:** Transmit power.

- **Preview:**



Beacon Period: 0

Beacon Frequency (Hz): 869525000

Beacon Channel Number: 1

Beacon Frequency Step (Hz): 0

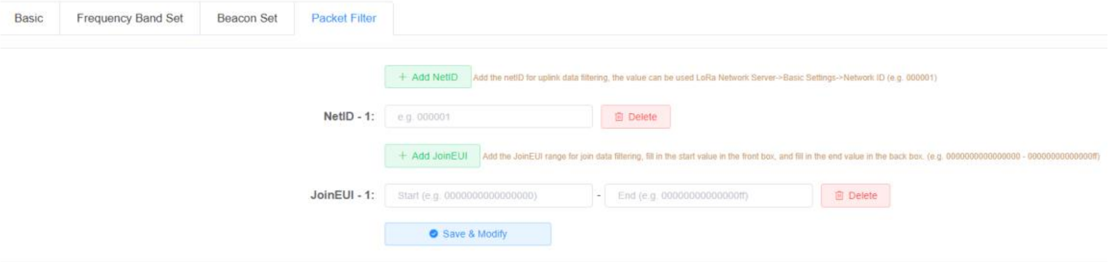
Beacon Spreading Factor: SF9

Beacon Bandwidth: 125000

Beacon Infodesc: 0

4. Packet Filter

- **Path:** LoRa Gateway → Packet Filter
- **Function:** On the gateway side, filter out some packets based on configured rules to reduce the amount of invalid data transmitted to the NS server, alleviate the processing pressure on the NS. Modes available: Semtech UDP GWMP Protocol or Build-in LoRa Server.
- **Details:**
  - **Supports configuration of NetID and JoinEUI.**
  - **NetID:** Network number filtering. The short address assigned during device joining is associated with the network number. By configuring this value, it can effectively filter out non-join packets and interference data, especially in embedded mode. This value can be configured as the network ID of the gateway to avoid interference from other device data.
  - **JoinEUI (AppEUI):** JoinEUI filtering, one of the triplets of the terminal. Multiple range values can be set here. Once set, JoinEUI outside the range will be filtered.
- **Preview:**



Add the netID for uplink data filtering, the value can be used LoRa Network Server->Basic Settings->Network ID (e.g. 000001)

NetID - 1: e.g. 000001

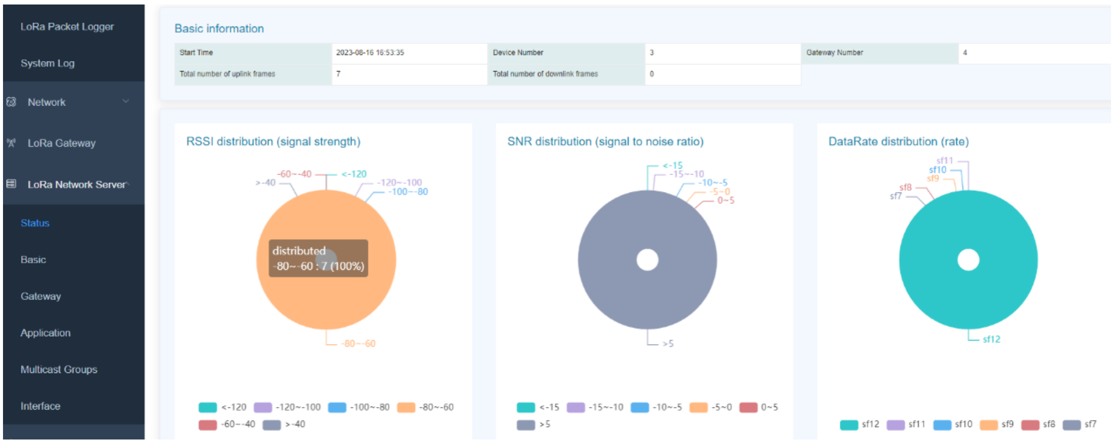
Add the JoinEUI range for join data filtering, fill in the start value in the front box, and fill in the end value in the back box. (e.g. 0000000000000000 - 00000000000000ff)

JoinEUI - 1: Start (e.g. 0000000000000000) - End (e.g. 00000000000000ff)

4.1.3.4 LoRa Network Server

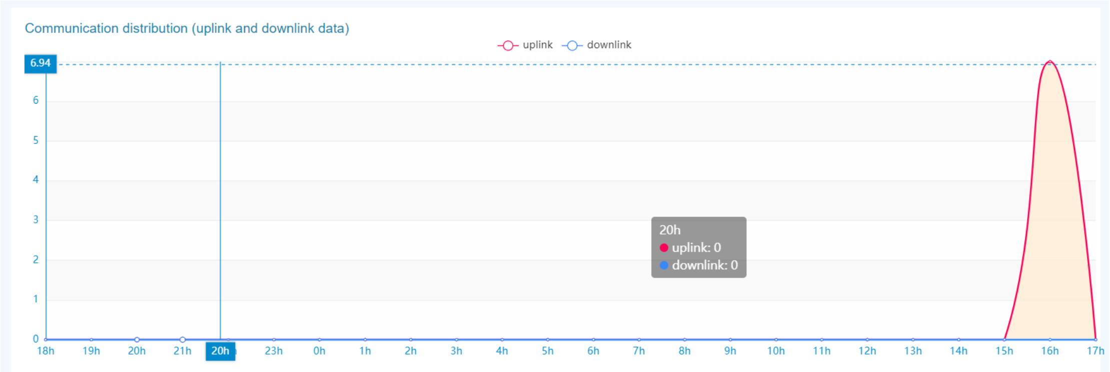
1. Status

- **Path:** LoRa Network Server → Status
- **Function:** Displays statistics flowing into the built-in NS server.
- **Details:**
  - **Basic Information:** Includes the number of gateways, devices, and statistics for device uplink and downlink data.
  - **RSSI, SNR, DataRate Distribution:** Used to analyze the communication quality between gateways and nodes.
  - **Communication Distribution:** Curve graph of uplink and downlink communication, analyzing whether the distribution of uplink and downlink data matches expectations.
- **Preview:**



Basic information

Start Time	2023-08-16 16:53:35	Device Number	3	Gateway Number	4
Total number of uplink frames	7	Total number of downlink frames	0		

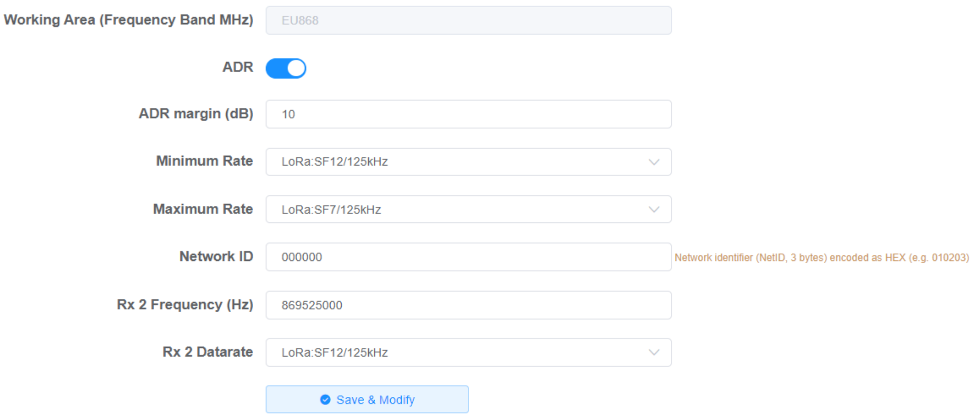

Communication distribution (uplink and downlink data)
uplink
downlink
6.94
6
5
4
3
2
1
0
20h
uplink: 0
downlink: 0
18h 19h 20h 21h 20h 23h 0h 1h 2h 3h 4h 5h 6h 7h 8h 9h 10h 11h 12h 13h 14h 15h 16h 17h

2. Basic Settings

- **Path:** LoRa Network Server → Basic Settings
- **Function:** Configures parameters related to the NS server.
- **Details:**
  - **Work Area:** Corresponds to the region parameter table's frequency band; it cannot be configured here and should match the configuration in LoRa Gateway → Frequency Configuration → Work Area.
  - **Enable Dynamic Data Rate Adjustment (ADR):** Indicates whether ADR functionality is enabled.

- ➢ **ADR Margin:** This value affects the sensitivity of ADR adjustments. A higher value makes adjustments less aggressive, while a lower value makes adjustments more aggressive.
- ➢ **Minimum Data Rate:** The minimum data rate for ADR adjustments.
- ➢ **Maximum Data Rate:** The maximum data rate for ADR adjustments.
- ➢ **Network ID:** Parameter for producing device short addresses, can be configured in filtering parameters to avoid interference.
- ➢ **Rx2 Frequency:** Frequency corresponding to the Rx2 window.
- ➢ **Rx2 Datarate:** Data rate corresponding to the Rx2 window.
- ➢ **Downlink Transmit Power (dBm):** Configures the transmit power. When set to -1, it will follow the specifications in the region parameter table.

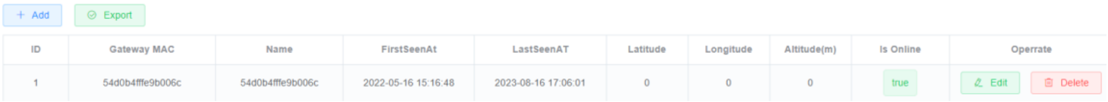
- **Preview:**



Working Area (Frequency Band MHz): EU868
ADR: ●
ADR margin (dB): 10
Minimum Rate: LoRa:SF12/125kHz
Maximum Rate: LoRa:SF7/125kHz
Network ID: 000000 — Network identifier (NetID, 3 bytes) encoded as HEX (e.g. 010203)
Rx 2 Frequency (Hz): 869525000
Rx 2 Datarate: LoRa:SF12/125kHz

3. Gateway

- **Path:** LoRa Network Server → Gateway
- **Function:** Manages the addition, deletion, modification, and viewing of gateways integrated with NS. Gateways generally do not need to be manually added; they will be added automatically when connected.
- **Details:**
  - ➢ Displays the list of gateways with detailed information, including online status.
- **Preview:**

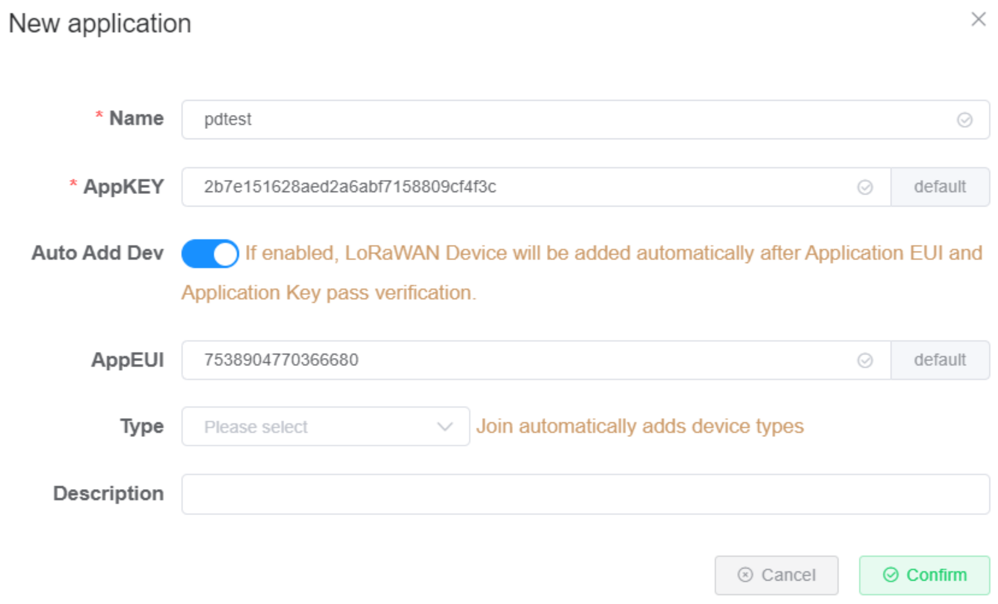


ID	Gateway MAC	Name	FirstSeenAt	LastSeenAT	Latitude	Longitude	Altitude(m)	Is Online	Operrate
1	54d0b4fffe9b006c	54d0b4fffe9b006c	2022-05-16 15:16:48	2023-08-16 17:06:01	0	0	0	true	Edit Delete

4. Application

- **Path:** LoRa Network Server → Application
- **Function:** Functions similarly to grouping, where different groups correspond to different application scenarios for easier management.
- **Details:**
  - ➢ **Clicking the add button opens the following page.**
    - ❖ **Name:** Equivalent to the group name for easy identification.
    - ❖ **AppKEY:** Corresponds to the AppKEY of the terminal; this value needs to be verified when adding devices automatically (clicking on the default on the right will change it to the default value of Four-Faith).

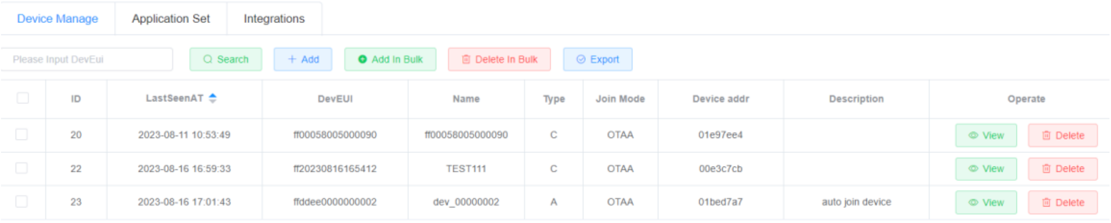
  - **Auto-add Devices**: When checked, devices can be added without prior manual addition. After AppKEY and AppEUI validation, devices will be added automatically.
  - **AppEUI (JoinEUI):** One of the triplets; configuration is required when enabling automatic device addition (clicking on the default on the right will change it to the default value of Four-Faith).
  - **Type:** Device type corresponding to the automatic addition of devices: ClassA or ClassC.
  - **Description:** Descriptive information.



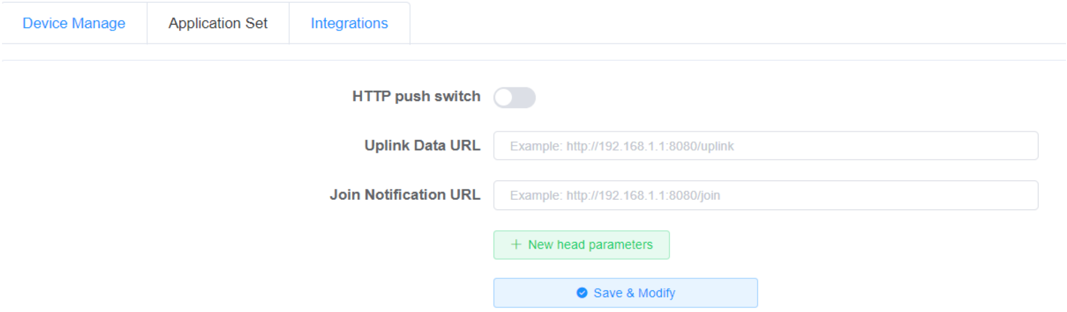
New application

* Name	pdtest	
* AppKEY	2b7e151628aed2a6abf7158809cf4f3c	default
Auto Add Dev	● If enabled, LoRaWAN Device will be added automatically after Application EUI and Application Key pass verification.	
AppEUI	7538904770366680	default
Type	Please select	Join automatically adds device types
Description		

- **Delete:** Cannot be deleted if there are devices associated with the application; devices must be deleted first.
- **View:** Entering the application allows access to the device list and more.
  - **Device Management:** Detailed explanation of the device's add, delete, modify, and query functionalities.
  - **Application Settings:** Similar to the initial creation, this allows modifications to existing applications.
  - **Interface Management:** Configuration for HTTP POST; enabling this function will push data from all devices under this application to a specified address using the HTTP POST method.



□	ID	LastSeenAT	DevEUI	Name	Type	Join Mode	Device addr	Description	Operate
□	20	2023-08-11 10:53:49	ff00058005000090	ff00058005000090	C	OTAA	01e97ee4		View Delete
□	22	2023-08-16 16:59:33	ff20230816165412	TEST111	C	OTAA	00e3c7cb		View Delete
□	23	2023-08-16 17:01:43	ffddee0000000002	dev_00000002	A	OTAA	01bed7a7	auto join device	View Delete

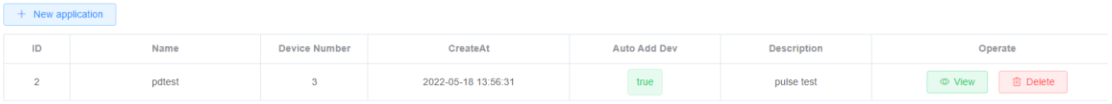


HTTP push switch

Uplink Data URL: Example: http://192.168.1.1:8080/uplink

Join Notification URL: Example: http://192.168.1.1:8080/join

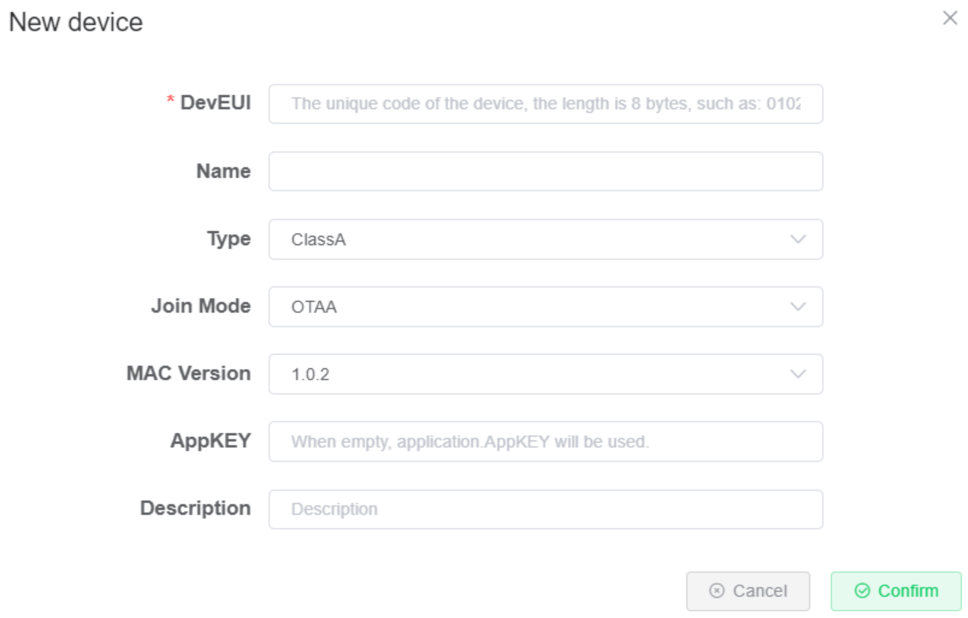
- **Preview:**



ID	Name	Device Number	CreateAt	Auto Add Dev	Description	Operate
2	pdtest	3	2022-05-18 13:56:31	true	pulse test	View Delete

5. Device

- **Path:** LoRa Network Server → Device
- **Function:** Adding, deleting, modifying, and querying devices. Web entry: LoRa Network Server → Application → View → Device Management
- **Details:**
  - **Add:** Setting basic parameters for the device. Network entry methods include OTAA (device initiates network joining) or ABP (no need for network joining). Specific AppKEY can be entered here when it is different from the AppKEY of the application.



New device

* DevEUI: The unique code of the device, the length is 8 bytes, such as: 010

Name:

Type: ClassA

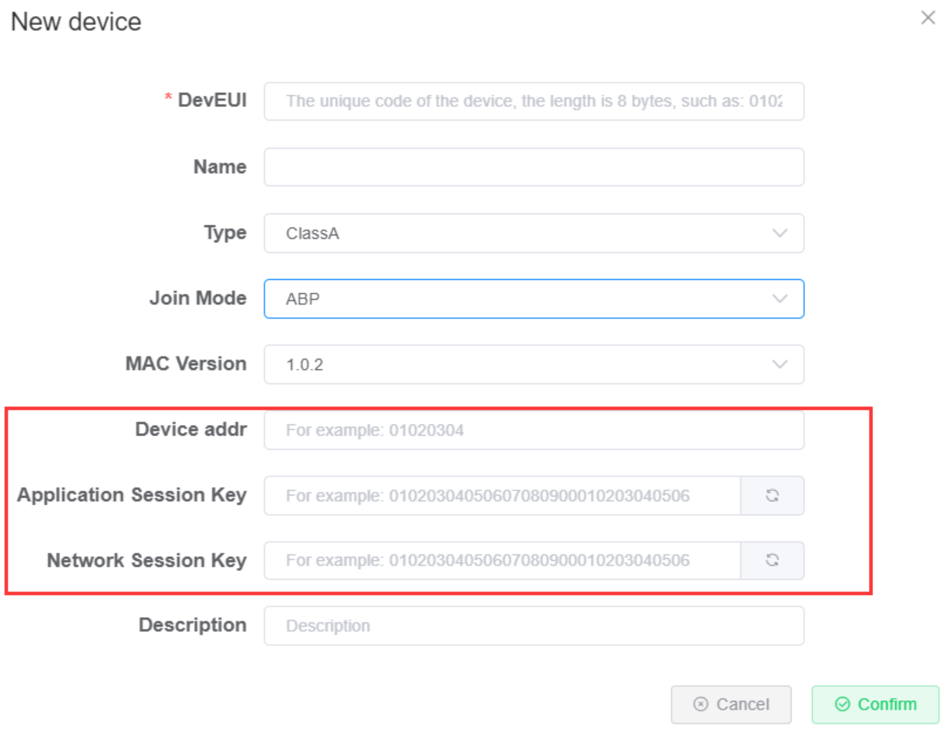
Join Mode: OTAA

MAC Version: 1.0.2

AppKEY: When empty, application.AppKEY will be used.

Description: Description

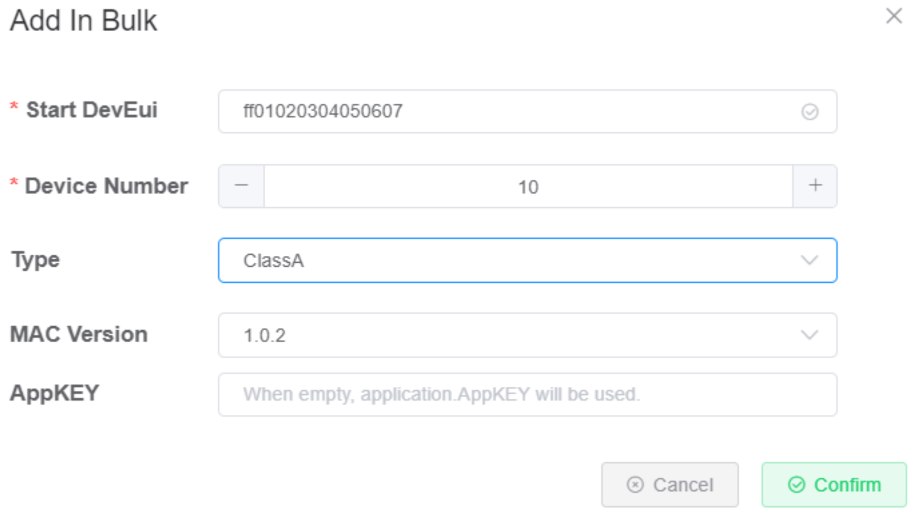
- ❖ **ABP Mode:** In this mode, you need to add the short address and session key information in the specified fields.



New device

* DevEUI	The unique code of the device, the length is 8 bytes, such as: 010:
Name	
Type	ClassA
Join Mode	ABP
MAC Version	1.0.2
Device addr	For example: 01020304
Application Session Key	For example: 01020304050607080900010203040506
Network Session Key	For example: 01020304050607080900010203040506
Description	Description

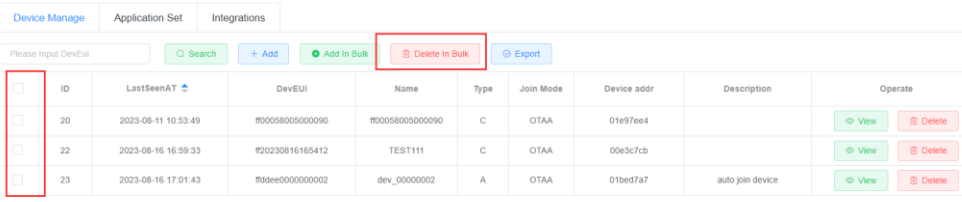
- **Bulk Add:** The parameters for bulk addition are essentially the same as for adding a single device. However, bulk addition is only applicable for OTAA devices.



Add In Bulk

* Start DevEui	ff01020304050607
* Device Number	10
Type	ClassA
MAC Version	1.0.2
AppKEY	When empty, application.AppKEY will be used.

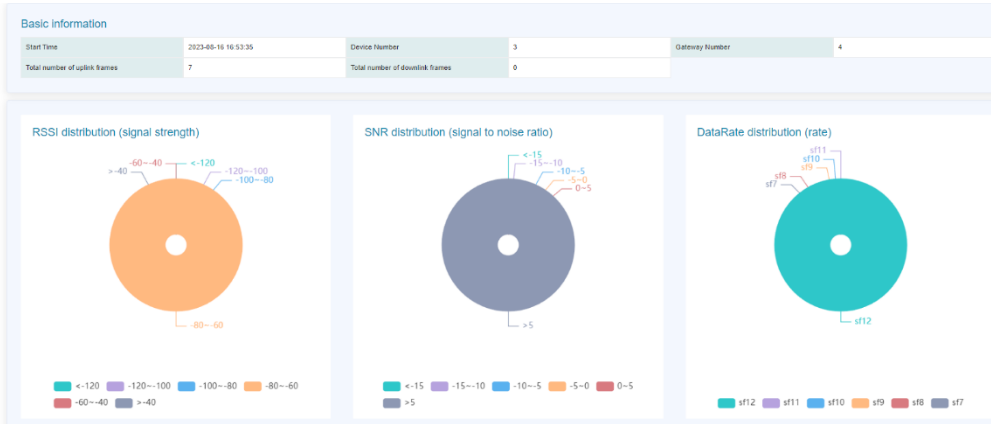
- **Bulk Delete:** First, select the devices you want to delete, then click "Bulk Delete" to remove them in batches.



□	ID	LastSeenAT	DevEUI	Name	Type	Join Mode	Device addr	Description	Operate
□	20	2023-08-11 10:53:49	ff00058005000090	ff00058005000090	C	OTAA	01e97ee4		View Delete
□	22	2023-08-16 16:59:33	ff20230816165412	TEST111	C	OTAA	00e3c7cb		View Delete
□	23	2023-08-16 17:01:43	ffddee0000000002	dev_00000002	A	OTAA	01bed7a7	auto join device	View Delete

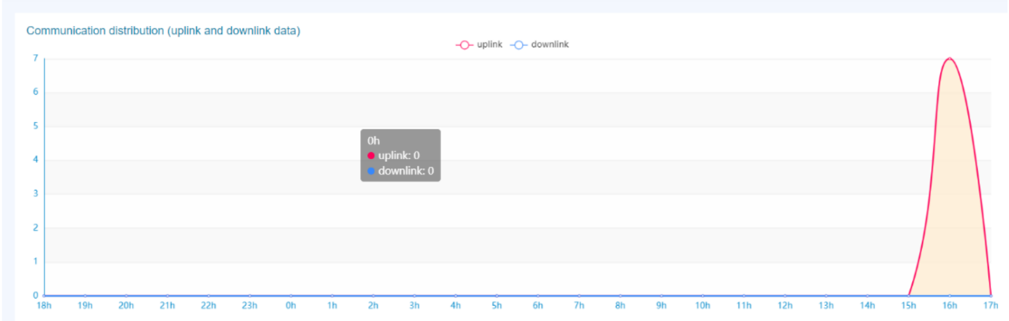
- **Export:** Export data in Excel format for backup and management purposes.

- **Device Details:** Click on "View" on the right side of the corresponding device to enter the details page.
  - ❖ **Overview:** Displays the uplink information and relevant statistics for the device, useful for analyzing packet loss and other issues.

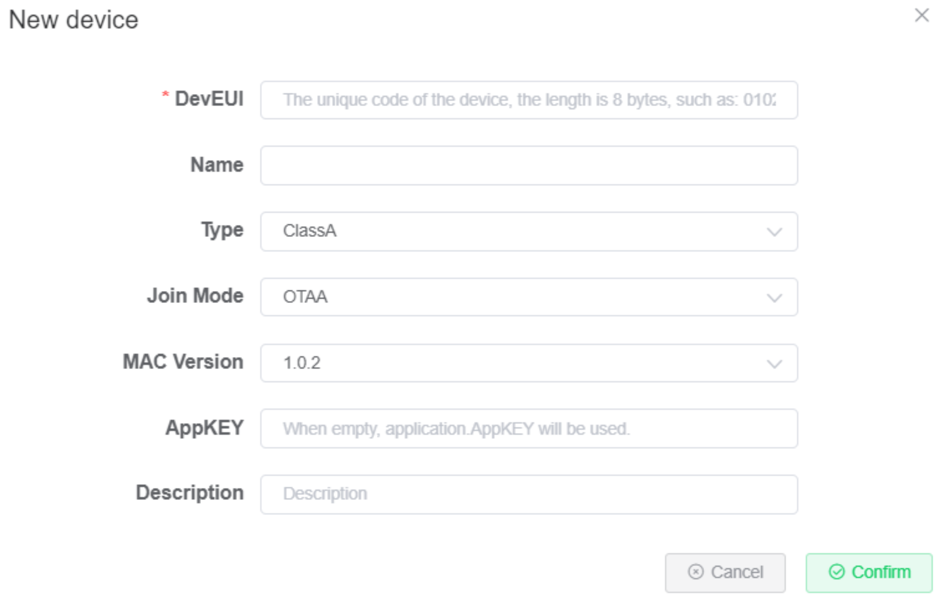


Basic information

Start Time	2023-08-16 16:53:35	Device Number	3	Gateway Number	4
Total number of uplink frames	7	Total number of downlink frames	0		



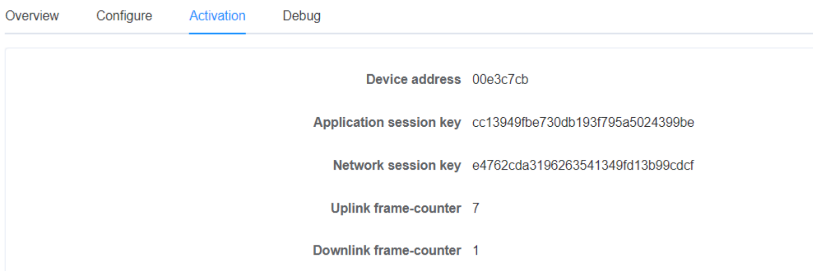
  - ❖ **Configuration:** Adjust the parameters of the device.



New device

* DevEUI	The unique code of the device, the length is 8 bytes, such as: 010:
Name	
Type	ClassA
Join Mode	OTAA
MAC Version	1.0.2
AppKEY	When empty, application AppKEY will be used.
Description	Description

  - ❖ **Activation Information:** Display of parameters after device activation.



Device address 00e3c7cb

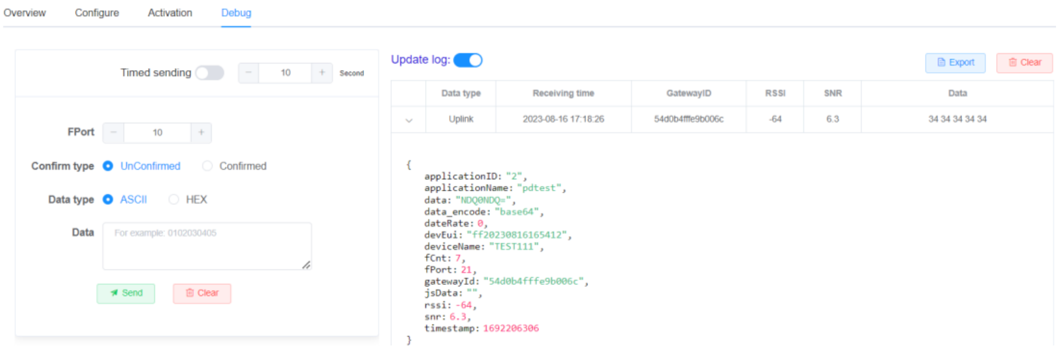
Application session key cc13949fbe730db193f795a5024399be

Network session key e4762cda3196263541349fd13b99cdcf

Uplink frame-counter 7

Downlink frame-counter 1

- ❖ **Online Debugging:** Allows for data downlink (scheduled), and displays uplink data, etc.



Timed sending ○ 10 Second

FPort 10

Confirm type ● UnConfirmed ○ Confirmed

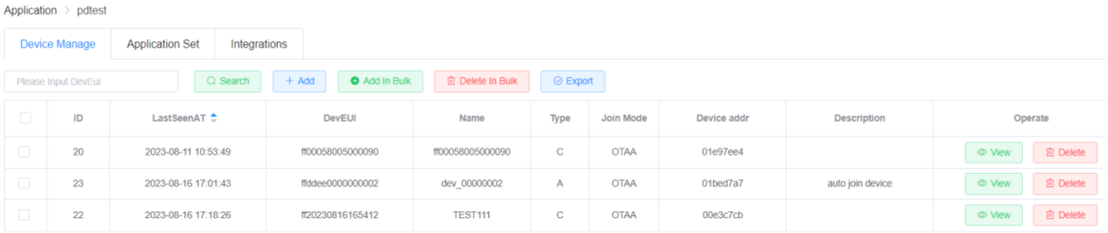
Data type ● ASCII ○ HEX

Data: For example: 0102030405

	Data type	Receiving time	GatewayID	RSSI	SNR	Data
⌄	Uplink	2023-08-16 17:18:26	54d0b4fffe9b006c	-64	6.3	34 34 34 34 34

```
{
    applicationID: "2",
    applicationName: "pdtest",
    data: "NDQ0NDQ=",
    data_encode: "base64",
    dateRate: 0,
    devEui: "ff20230816165412",
    deviceName: "TEST111",
    fCnt: 7,
    fPort: 21,
    gatewayId: "54d0b4fffe9b006c",
    jsData: "",
    rssi: -64,
    snr: 6.3,
    timestamp: 1692206306
}
```

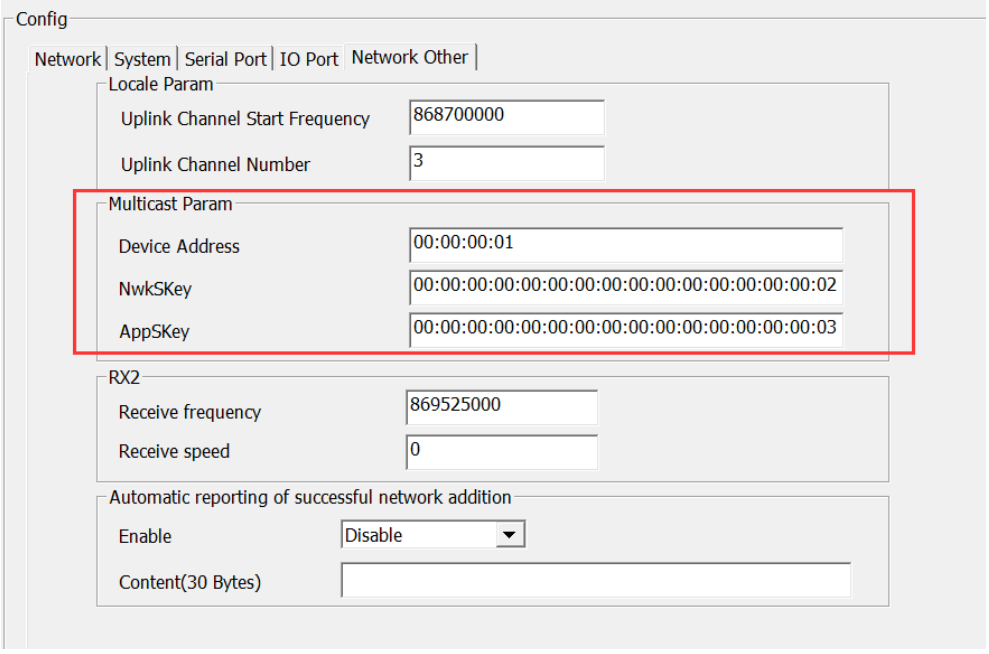
- Preview:



□	ID	LastSeenAT	DevEUI	Name	Type	Join Mode	Device addr	Description	Operate
□	20	2023-08-11 10:53:49	ff00058005000090	ff00058005000090	C	OTAA	01e97ee4		View Delete
□	23	2023-08-16 17:01:43	ffddee0000000002	dev_00000002	A	OTAA	01bed7a7	auto join device	View Delete
□	22	2023-08-16 17:18:26	ff20230816165412	TEST111	C	OTAA	00e3c7cb		View Delete

6. Multicast

- **Path:** LoRa Network Server → Multicast
- **Functionality:** Multicast here refers to all terminals with the same parameters corresponding to this NS. The downlink of multicast data can be sent via MQTT (the webpage here also supports sending tests).
- **Details:**
  - ➢ **Add Multicast:** The values corresponding to Four-Faith terminals are shown below.



Config

Locale Param

Uplink Channel Start Frequency	868700000
Uplink Channel Number	3

Multicast Param

Device Address	00:00:00:01
NwkSKey	00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:02
AppSKey	00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:03

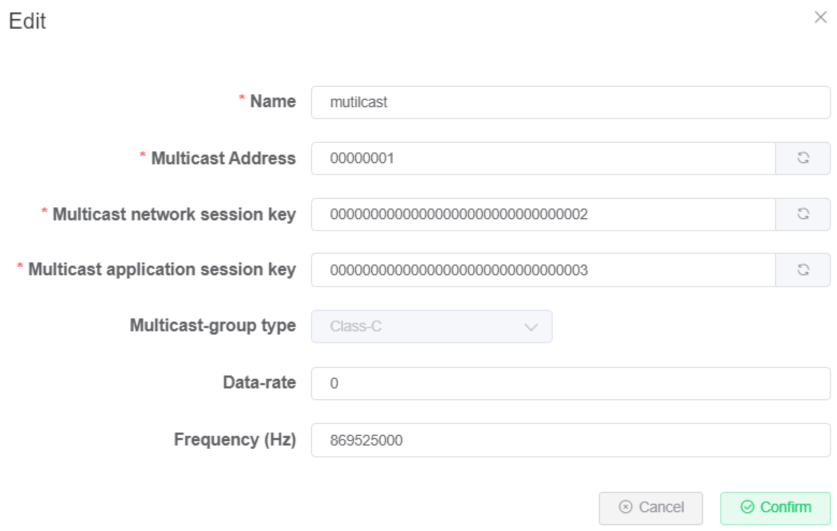
RX2

Receive frequency	869525000
Receive speed	0

Automatic reporting of successful network addition

Enable	Disable
Content(30 Bytes)	

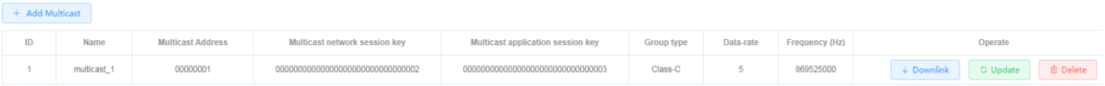
- Configure multicast parameters based on the values above.



Edit

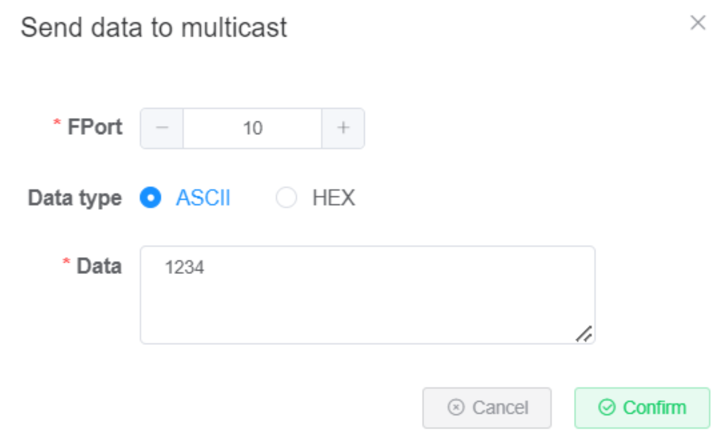
* Name	mutilcast
* Multicast Address	00000001
* Multicast network session key	00000000000000000000000000000002
* Multicast application session key	00000000000000000000000000000003
Multicast-group type	Class-C
Data-rate	0
Frequency (Hz)	869525000

- After creation, you will be able to see the following multicast list information.



ID	Name	Multicast Address	Multicast network session key	Multicast application session key	Group type	Data-rate	Frequency (Hz)	Operate
1	multicast_1	00000001	00000000000000000000000000000002	00000000000000000000000000000003	Class-C	5	869525000	Downlink Update Delete

- Sending Data Test: Click "Downlink" to open the data transmission page.

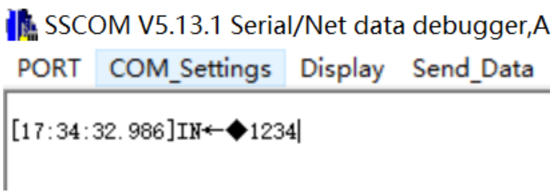


Send data to multicast

* FPort: 10

Data type: ● ASCII ○ HEX

* Data: 1234



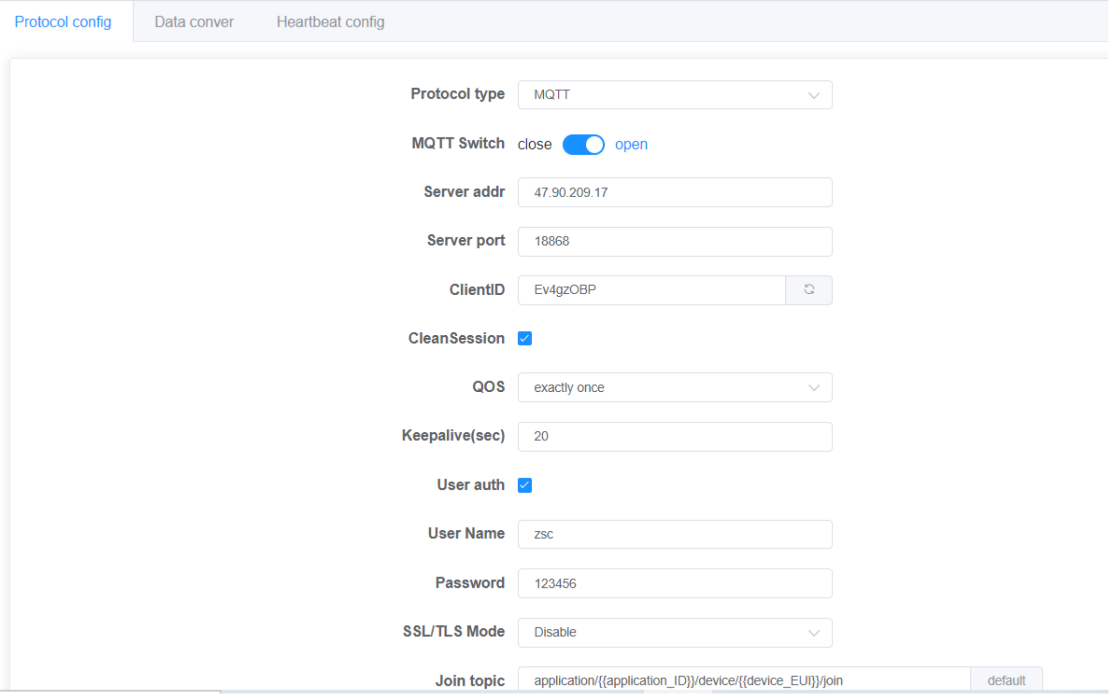
SSCOM V5.13.1 Serial/Net data debugger,A

```
[17:34:32.986]IN←◆1234
```

- ➢ When officially in use, multicast data can be sent via MQTT or TCP. Refer to the data format for specific details.

7. Interfaces

- **Path:** LoRa Network Server → Interfaces
- **Function:** Configuration page for integrating NS with client platforms, supporting both MQTT and TCP communication methods. Data can be transformed using JavaScript functions, and heartbeat configurations are supported.
- **Details:**
  - ➢ **Protocol Configuration**
    - ❖ **NONE:** Not enabled
    - ❖ **MQTT:** Configuration of MQTT parameters. Specific topics and data formats are detailed in the data format section.



Protocol type: MQTT

MQTT Switch: close ● open

Server addr: 47.90.209.17

Server port: 18868

ClientID: Ev4gzOBP

CleanSession: ☑

QOS: exactly once

Keepalive(sec): 20

User auth: ☑

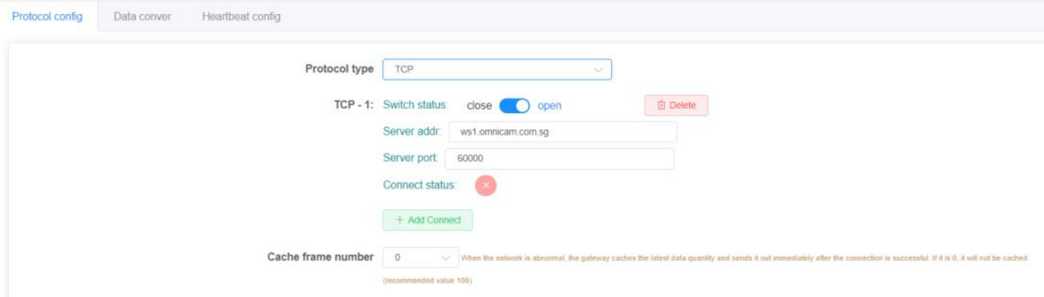
User Name: zsc

Password: 123456

SSL/TLS Mode: Disable

Join topic: application/{{application_ID}}/device/{{device_EUI}}/join default

- **TCP:** Connects to a TCP server, allowing simultaneous connections to multiple servers. Connection status can be monitored to assess the connectivity situation.



Protocol type: TCP

TCP - 1: Switch status: close ● open

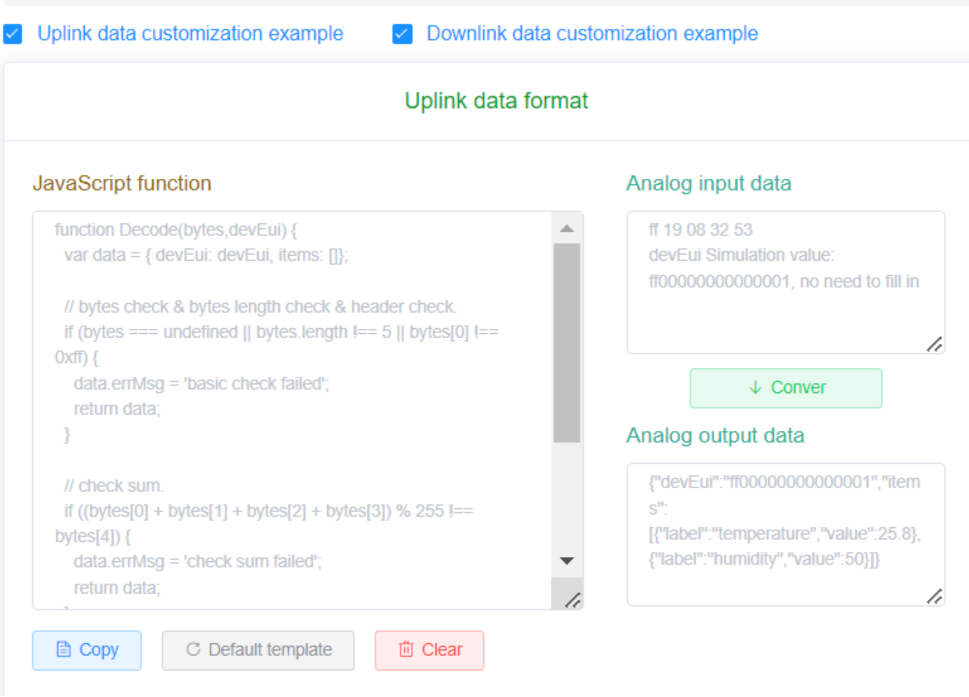
Server addr: ws1.omnicam.com.sg

Server port: 60000

Connect status: ✕

Cache frame number: 0 When the network is abnormal, the gateway caches the latest data quantity and sends it out immediately after the connection is successful. If it is 0, it will not be cached (recommended value 100)

- **Data Transformation:** If not configured here, the default data format will be used for communication. If data transformation is required, functions can be configured for conversion. Upon arrival at the gateway, both uplink and downlink data can be transformed using specified functions before forwarding.
  - **Upstream Transformation**

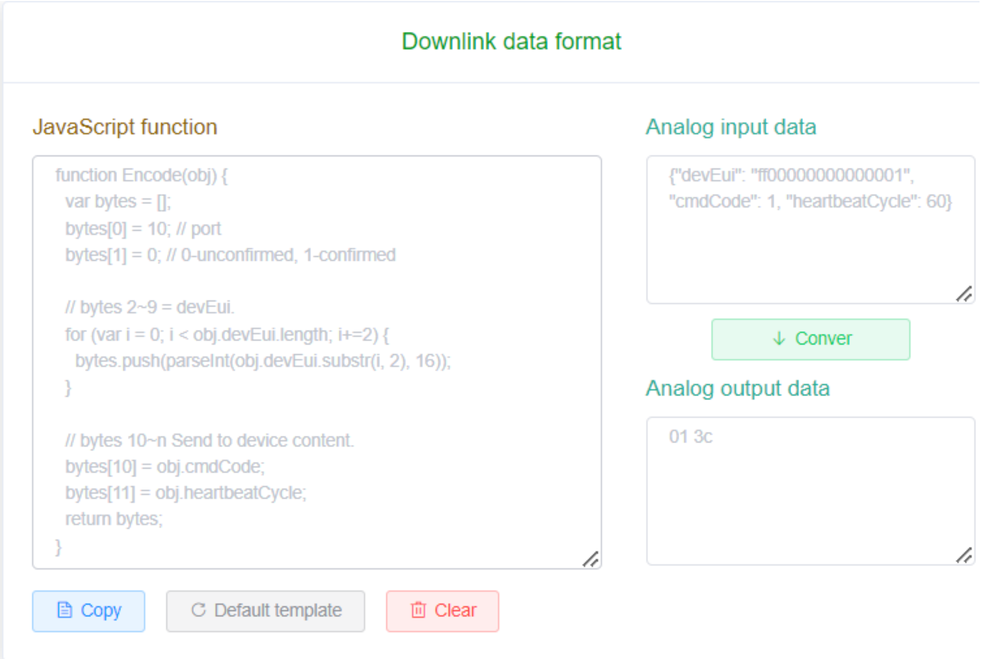


```
function Decode(bytes,devEui) {
  var data = { devEui: devEui, items: []};

  // bytes check & bytes length check & header check.
  if (bytes === undefined || bytes.length !== 5 || bytes[0] !==
0xff) {
    data.errMsg = 'basic check failed';
    return data;
  }

  // check sum.
  if ((bytes[0] + bytes[1] + bytes[2] + bytes[3]) % 255 !==
bytes[4]) {
    data.errMsg = 'check sum failed';
    return data;
```

ff 19 08 32 53
devEui Simulation value: ff00000000000001, no need to fill in

```
{"devEui":"ff00000000000001","items":[{"label":"temperature","value":25.8},{"label":"humidity","value":50}]}
```

- ❖ **Downstream Transformation**

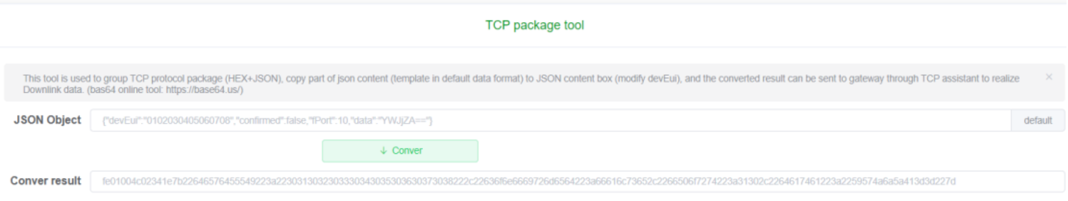


```
function Encode(obj) {
  var bytes = [];
  bytes[0] = 10; // port
  bytes[1] = 0; // 0-unconfirmed, 1-confirmed

  // bytes 2~9 = devEui.
  for (var i = 0; i < obj.devEui.length; i+=2) {
    bytes.push(parseInt(obj.devEui.substr(i, 2), 16));
  }

  // bytes 10~n Send to device content.
  bytes[10] = obj.cmdCode;
  bytes[11] = obj.heartbeatCycle;
  return bytes;
}
```

```
{"devEui": "ff00000000000001", "cmdCode": 1, "heartbeatCycle": 60}
```

```
01 3c
```

- ❖ **TCP Packet Assembly Tool:** During the testing phase when connecting to a TCP server, this tool can be used to generate corresponding data for testing through the TCP server. In a normal project, a program can be developed to generate this data.

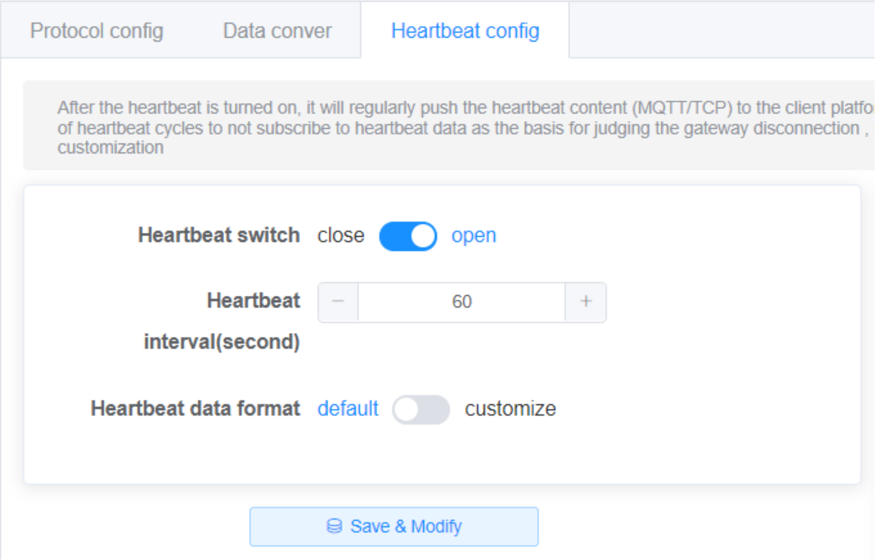


This tool is used to group TCP protocol package (HEX+JSON), copy part of json content (template in default data format) to JSON content box (modify devEui), and the converted result can be sent to gateway through TCP assistant to realize Downlink data. (bas64 online tool: https://base64.us/)

JSON Object: {"devEui":"0102030405060708","confirmed":false,"fPort":10,"data":"YWJjZA=="}

Conver result: fe01004c02341e7b226465764555 49223a2230313032303330343035303630373038222c22636f6e6669726d6564223a66616c73652c2266506f7274223a31302c2264617461223a2259574a6a5a413d3d227d

- ➢ **Heartbeat Configuration:** You can configure the heartbeat switch, heartbeat interval time, and heartbeat data format. It supports configuration as a custom string. Heartbeat is mainly used for regularly reporting status information. The gateway can also use heartbeat to determine the connection status with the MQTT server.



After the heartbeat is turned on, it will regularly push the heartbeat content (MQTT/TCP) to the client platfo
of heartbeat cycles to not subscribe to heartbeat data as the basis for judging the gateway disconnection ,
customization

Heartbeat switch close ● open

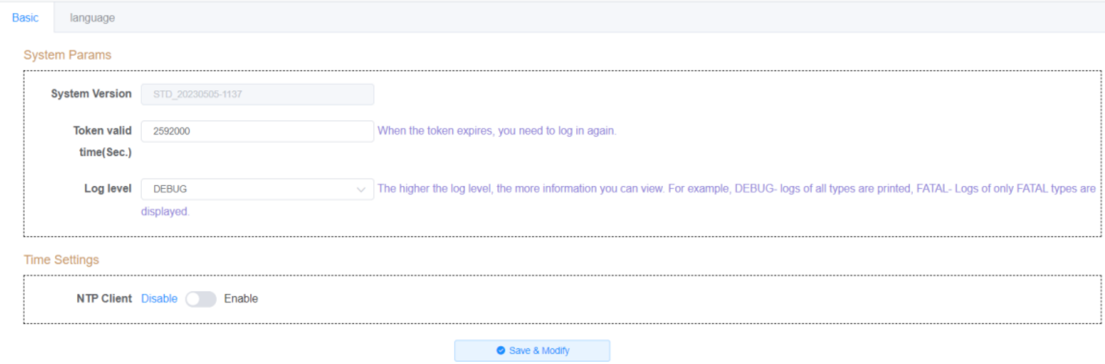
Heartbeat interval(second) − 60 +

Heartbeat data format default ○ customize

4.1.3.5 System

1. **System:**

- **Path:** System → System
- **Functionality:** View program version, configure token duration, time settings, and language switch.
- **Details:**
  - **System Program Version:** Use version information to troubleshoot related issues.
  - **Token Valid Duration:** The shorter the time, the more frequent the need to log in to the web page.
  - **NTP Time Configuration:** Configure.
- **Preview:**



System Params

System Version STD_20230505-1137

Token valid time(Sec.) 2592000 When the token expires, you need to log in again.

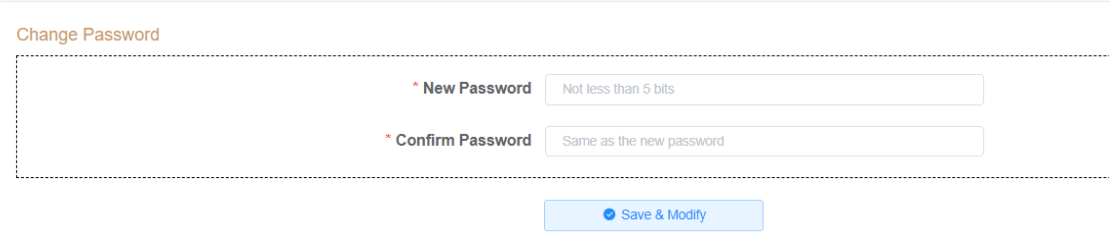
Log level DEBUG The higher the log level, the more information you can view. For example, DEBUG- logs of all types are printed, FATAL- Logs of only FATAL types are displayed.

Time Settings

NTP Client Disable ○ Enable

2. Change Password:

- **Path:** System → Change Password
- **Functionality:** Modify the password for the gateway system, with a length range of 5-32 characters.
- **Details:**
  - ➢ Enter the new password and confirm it. After modification, exit the system. When logging in again, use the new password.
- **Preview:**



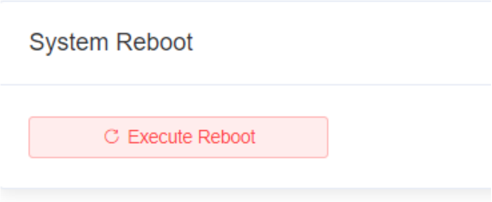
Change Password

* New Password: Not less than 5 bits

* Confirm Password: Same as the new password

3. Restart:

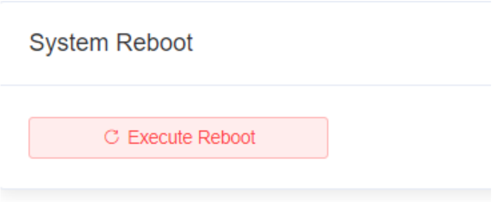
- **Path:** System → Restart
- **Functionality:** Reboot the gateway.
- **Details:**
  - ➢ Click to initiate the restart of the gateway.
- **Preview:**



System Reboot

4. Factory Reset:

- **Path:** System → Factory Reset
- **Functionality:** Clicking this button will restore the gateway parameters, mainly router-related parameters such as network settings (LoRa-related parameters like device lists and network information will not be deleted).
- **Details:**
  - ➢ Click to execute the factory reset.
- **Preview:**



System Reboot

4.1.4 Data Format

4.1.4.1 Data Explanation

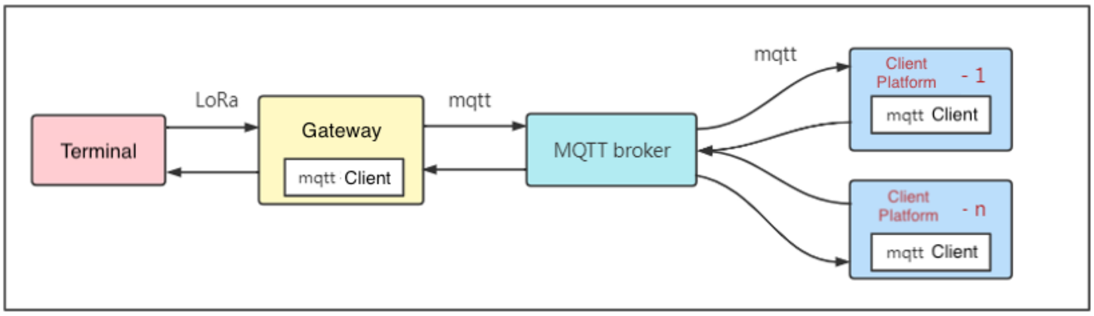
1. Data Format Explanation:

To communicate with the client, the protocols include MQTT, TCP, and HTTP, where MQTT and TCP support bidirectional communication. However, HTTP only supports the gateway pushing data to the client via the POST method and does not support downstream communication. The data formats for each protocol are as follows:

- **MQTT Data:** Topic + JSON Content
- **TCP Data:** Data Header + JSON Content
- **HTTP Data:** URL + JSON Content

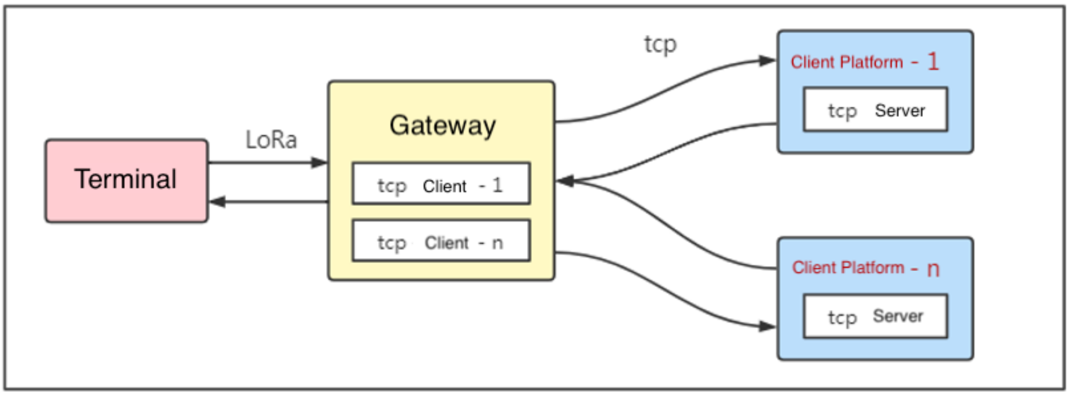
Note: The JSON content data format is entirely consistent within the same type. If JavaScript function conversion is applied, it will be universally applied to all formats.

2. MQTT Data Flow:



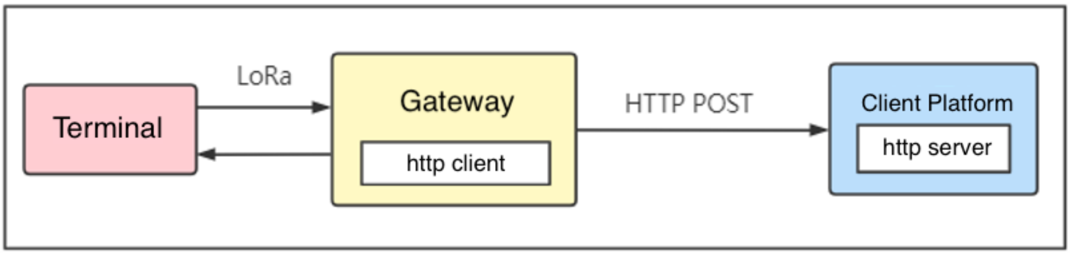
As shown in the above diagram, in this mode, you need to deploy an MQTT Broker first. Both the gateway and the client platform establish connections with it and subscribe to corresponding topics according to the topic format. If the client needs multiple sets of data, this can be achieved by connecting multiple clients and subscribing. In comparison to the TCP mode, this method involves an additional step of deploying an external MQTT server.

3. TCP



As shown in the diagram above, in this mode, the client platform opens a TCP server, and the gateway is configured in TCP mode, pointing to the IP and port of the corresponding server. This allows multiple TCP connections to be established simultaneously.

4. HTTP

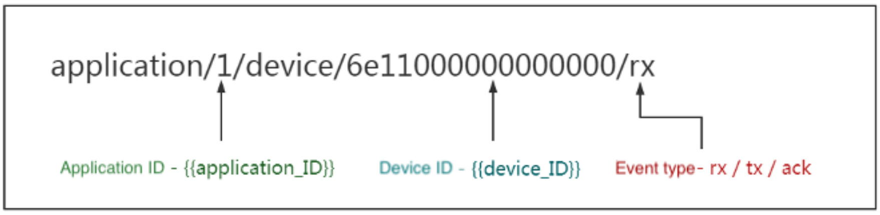


As shown in the diagram above, the configuration for this mode is in the interface settings of each application. This mode only supports data push and does not support downstream data.

4.1.4.2 MQTT Data Format

1. MQTT Topic and Data Format

- Default MQTT Topic Format


application/1/device/6e11000000000000/rx
Application ID - {{application_ID}}
Device ID - {{device_ID}}
Event type - rx / tx / ack

- MQTT format primarily includes topics and data content, which can be displayed and modified in the interface.
- The default topic contains {{application_ID}} and {{device_EUI}}.
  - ➢ {{application_ID}}: Application ID, which will be replaced with the corresponding application ID when reporting data (e.g., application/1/device/6e11000000000000/rx). It also needs to be replaced with the actual application ID when sending downlink data (e.g., application/1/device/6e11000000000000/tx).
  - ➢ {{device_EUI}}: Device's unique identifier, which will be replaced with the device's EUI when reporting data (e.g., application/1/device/6e11000000000000/rx). It also needs to be replaced with the actual device EUI when sending downlink data (e.g., application/1/device/6e11000000000000/tx). If this field is present in the topic, the JSON content of the downlink data may omit the device's unique identifier.

- Modification Instructions:

  - Topics can be modified, such as changing it to lorawan/uplink.
  - {{application_ID}}: Can be deleted. If removed, only the application ID will be excluded.
  - {{device_EUI}}: If removed, the topic won't recognize the corresponding device. In this case, the JSON content of downlink data must include the device's unique identifier (as explained in the subsequent content).

- Examples of Subscribing to Topics:

  - Subscribe to a specific event for a single device: application/1/device/6e11000000000000/rx
  - Subscribe to all events for a single device: application/1/device/6e11000000000000/+
  - Subscribe to a specific event for all devices under an application: application/1/device/+/rx
  - Subscribe to all events for all devices under an application: application/1/device/#
  - Subscribe to a specific event for all devices under all applications: application/+/device/+/rx
  - Subscribe to all events for all devices under all applications: application/+/device/+/+ or application/##

- The data content follows the JSON format, and the specific format is detailed later. It's essential to note that if the {{device_EUI}} is removed from the downlink topic, the topic won't recognize the specific device. In such cases, you need to look for the "devEui" field in the data content. If it also doesn't exist, the device-specific data will be lost.

2. **Uplink Data:**

- **Execution Condition:** Forwarded upon receiving business data reports from devices that have joined the network.
- **Default Topic Format:** application/{{application_ID}}/device/{{device_EUI}}/rx
- **Example Default Topic:** application/1/device/6e11000000000000/rx
- **Example Default JSON Data Content:**

```
{

  "applicationID": "1",

  "applicationName": "temperature",

  "deviceName": "dev_00000000",

  "devEui": "6e11000000000000",

  "rxInfo": [{
        "gatewayID": "ff0000000000000a",
        "name": "ff0000000000000a",
        "time": "", // Only applicable when the gateway can receive GPS signals to provide actual values.
        "rssi": -76,
        "loRaSNR": 7.5,
        "location": {
            "latitude": 0,
            "longitude": 0,
            "altitude": 0
        }
    }],
    "txInfo": {
        "frequency": 868100000,
        "dr": 0
    },
    "adr": false,
    "fCnt": 6,
    "fPort": 32,
"data": "MTQ1OTYzNTgy" // Base64 encoding, as explained in the later section "Base64 Encoding and Decoding."
}
```

3. Join Data:

- **Execution Condition:** Pushed upon receiving a device's join request and responding to the join accept packet.
- **Default Topic Format:** application/{{application_ID}}/device/{{device_EUI}}/join
- **Example Default Topic:** application/1/device/6e11000000000000/join
- **Example Default Data Content:**

```
{
    "applicationID": "1",
    "applicationName": "temperature",
    "deviceName": "dev_00000000",
    "devEui": "6e11000000000000",

    "devAddr": "01b0e489"

  }
```

4. **Downlink Data:**

- **Execution Condition:** Sent when business data needs to be delivered to the device.
- **Default Topic Format:** application/{{application_ID}}/device/{{device_EUI}}/tx
- **Example Default Topic:** application/1/device/6e11000000000000/tx
- **Example Default Data Content:**

```
  {

    "devEui": "6e11000000000000",

    "confirmed": true,

    "fPort": 12,

    "data": "MTIzNA==" // Base64 encoding, as explained in the later section "Base64 Encoding and

Decoding." In this context, it corresponds to "1234."

  }
```

- **Convenient Test Data**(The data above contains spaces which may lead to sending failures)

```
  {"devEui":"6e11000000000000","confirmed":true,"fPort":12,"data":"MTIzNA=="}
```

5. **Downlink Acknowledgment Response:**

- **Execution Condition:** After receiving the downlink acknowledgment, push pending device responses.
- **Default Topic Format:** application/{{application_ID}}/device/{{device_EUI}}/ack
- **Default Topic Example:** application/1/device/6e11000000000000/ack
- **Default JSON Data Content Example:**

```
  {

    "applicationID": "1",

    "applicationName": "temperature",

    "deviceName": "dev_00000000",

    "devEui": "6e11000000000000",

    "acknowledged": true

  }
```

6. **Downlink Multicast Data:**

- **Execution Condition:** To send multicast information to devices with the same triplets in the multicast group.

- **Default Topic Format:** mcast_group/{{mcast_ID}}/tx
- **Default Topic Example:** mcast_group/1/tx
- **Default JSON Data Content Example:**

```
{
  "multicastGroupId": 1,
  "fPort": 10,
  "data": "YWJjZA==" // base64 Encoding
}
```

- **Convenient Test Data Example:**

```
{"multicastGroupId":1,"fPort":10,"data":"YWJjZA=="}
```

7. Heartbeat Data:

- **Execution Condition:** Heartbeat switch is turned on, heartbeat interval > 0, and heartbeat content is non-empty.
- **Default Topic:** lorawan/heartbeat
- **Default JSON Data Content Example:**

```
{
  "gateways": [{
    "gatewayID": "ff0000000000000a",
    "gatewayName": "ff0000000000000a",
    "lastSeenAt": "2022-04-29 14:18:36",
    "isOnline": true,
    "longitude": 0,
    "latitude": 0
  }],
  "applications": [{
    "applicationID": 1,
    "name": "app",
    "deviceNum": 1,
    "activatNum": 1,
    "isAutoJoin": false
  }]
}
```

4.1.4.3 TCP Data Format

1. TCP Data Format

Offset	Bytes	Function	Identifier	Value Example
0	1	Frame Header	header	0xFE
1	1	Version Number (Current V1)	version	0x01
2	2	JSON Data Length (Big Endian)	length	0x0001
4	1	Data Type	type	0x00- Heartbeat Packet
5	2	Random Key (Big Endian)	random	0x1234
7	n	JSON Content	JSON Object	{...}

- The first 7 bytes represent the TCP data header, and starting from the 7th byte is the JSON content, which is the same as MQTT and HTTP.

2. Uplink Data

Offset	Bytes	Function	Value or Description
0	1	header	0xFE
1	1	version	0x01
2	2	length	0x018A
4	1	type	0x01
5	2	random	0x1234
7	394	JSON object	{ "applicationID": "2", "applicationName": "app1", "deviceName": "dev_00000001", "devEui": "ff00000000000001", "rxInfo": [

```
    {
      "gatewayID": "54c345fffed5a1e3",
      "name": "54c345fffed5a1e3",
      "time": "2021-11-19T01:51:01.136686Z",
      "rssi": -107,
      "loRaSNR": 7.5,
      "location": {
           "longitude": 118.03394,
           "latitude": 24.48405,
           "altitude": 89
}
    }
  ],
  "txInfo": {
    "frequency": 923400000,
    "dr": 4
  },
  "adr": false,
  "fCnt": 4,
  "fPort": 32,
  "data": "YWJjZA=="
}
```

	Description	Type
applicationID	Application ID	string
applicationName	Application Name	string
deviceName	Device Name	string
devEui	Device EUI	string
rxInfo	Gateway Information for Received Data	Struct Array

			- gatewayID	Gateway EUI	string
			- name	Gateway Name	string
			- time	GPS Time	string
			- rssi	Signal Strength	float64
			- loRaSNR	Signal-to-Noise Ratio	float64
			- location	GPS Location (Empty when no GPS signal)	
			- longitude	Longitude	float64
			- latitude	Latitude	float64
			- altitude	Altitude	float64
			TxInfo	Device Data Transmission Parameters	
			- frequency	Frequency	uint32
			- dr	Rate	uint8
			adr	ADR Request Status	bool
			fCnt	Uplink Frame Counter	uint32
			fPort	Uplink Port	uint8
			data	Business Data (Base64 encoded format)	string

3. Join Data

Offset	Bytes	Function	Value or Description
0	1	header	0xFE
1	1	version	0x01
2	2	length	0x007B
4	1	type	0x03
5	2	random	0x1234
7	123	JSON object	{ "applicationID": "2", "applicationName": "app1", "deviceName": "dev_00000001",

			"devEui": "ff00000000000001", "devAddr": "032013ac" }

	Description	Type
applicationID	Application ID	string
applicationName	Application Name	string
deviceName	Device Name	string
devEui	Device Unique Identifier	string
devAddr	Short Address Assigned by Device Joining	string

4. Downlink Data

Offset	Bytes	Function	Value or Description
0	1	header	0xFE
1	1	version	0x01
2	2	length	0x004D
4	1	type	0x02
5	2	random	0x1234

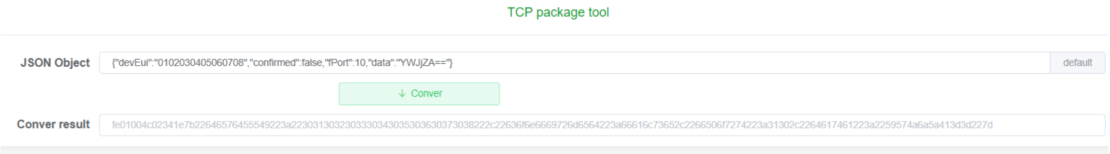
7	77	JSON object	{
			"devEui": "ff00000000000001",
			"confirmed": false,
			"fPort": 10,
			"data": "YWJjZA=="
			}

	Description	Type
devEui	Device EUI	string
confirmed	Whether to acknowledge the packet (default is false)	bool
fPort	port (default is 10)	uint8
data	business data sent (base64 encoded)	string

- Convenient test data (the data above contains spaces, which may cause transmission failures at times)

```
{"devEui":"ff00000000000001","confirmed":true,"fPort":10,"data":"MTIzNA=="}
```

Note: You can use the TCP Packet Tool on the webpage (Path: LoRa Network Server -> Interface -> Data Transformation -> TCP Packet Tool) to generate corresponding data for testing, as shown below:



TCP package tool

JSON Object: {"devEui":"0102030405060708","confirmed":false,"fPort":10,"data":"YWJjZA=="}

Conver result: fe01004c02341e7b22646576455549223a2230313032303330343035303630373038222c22636f6e6669726d6564223a66616c73652c2266506f7274223a31302c2264617461223a2259574a6a5a413d3d227d

The data that can be used for testing with TCP server (devEui may need to be modified during testing) includes:：

fe01004b0204427b22646576455549223a2266663030303030303030303030303031222c22636f6e6669726d6564223a747275652c2266506f7274223a31302c2264617461223a224d54497a4e413d3d227d

5. Downlink Acknowledgment Packet Response

<table>
<tr><th>Offset</th><th>Bytes</th><th>Function</th><th>Description or Value</th></tr>
<tr><td>0</td><td>1</td><td>header</td><td>0xFE</td></tr>
<tr><td>1</td><td>1</td><td>version</td><td>0x01</td></tr>
<tr><td>2</td><td>2</td><td>length</td><td>0x0000</td></tr>
<tr><td>4</td><td>1</td><td>type</td><td>0x05</td></tr>
<tr><td>5</td><td>2</td><td>random</td><td>0x1234</td></tr>
<tr><td>7</td><td>77</td><td>JSON object</td><td>{
"applicationID": "1",
"applicationName": "app1",
"deviceName": "dev_00000000",
"devEui": "6e00000000000000",
"acknowledged": true
}
<table>
<tr><th></th><th>Description</th><th>Type</th></tr>
<tr><td>applicationID</td><td>Application ID</td><td>String</td></tr>
<tr><td>applicationName</td><td>Application Name</td><td>String</td></tr>
</table></td></tr>
</table>

deviceName	Device Name	String
devEui	Device Unique Identifier	String
acknowledged	Response Status (Success: true)	Bool

6. Downlink Multicast Data

Offset	Bytes	Function	Value or Description
0	1	header	0xFE
1	1	version	0x01
2	2	length	0x0000
4	1	type	0x04
5	2	random	0x1234

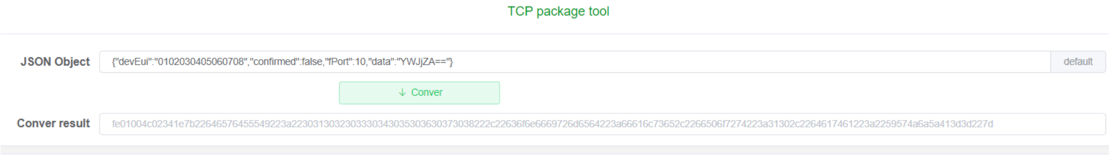
7	77	JSON object	{
			"multicastGroupId": 1,
			"fPort": 10,
			"data": "YWJjZA=="
			}

	Description	Type
multicastGroupId	Multicast ID	int
fPort	Port（Default 10）	uint8
data	Sending business data (Base64 encoded format)	string

● Convenient test data

```
{"multicastGroupId":1,"fPort":10,"data":"YWJjZA=="}
```

Note: You can utilize the TCP Packet Tool on the web page (Path: LoRa Network Server → Interface → Data Conversion → TCP Packet Tool) to generate corresponding data for testing, as shown below:



TCP package tool

JSON Object: {"devEui":"0102030405060708","confirmed":false,"fPort":10,"data":"YWJjZA=="}

Conver result: fe01004c02341e7b22646576455549223a2230313032303330343035303630373038222c22636f6e6669726d6564223a66616c73652c2266506f7274223a31302c2264617461223a2259574a6a5a413d3d227d

The converted result can be used for testing by sending it to the TCP server. The content of the example above is:

```
fe01003304341e7b226d756c74696361737447726f75704964223a312c2266506f7274223a31302c2264617461
223a2259574a6a5a413d3d227d
```

7. Heartbeat Data

<table>
<tr><th>Offset</th><th>Bytes</th><th>Functions</th><th>Value or Description</th></tr>
<tr><td>0</td><td>1</td><td>header</td><td>0xFE</td></tr>
<tr><td>1</td><td>1</td><td>version</td><td>0x01</td></tr>
<tr><td>2</td><td>2</td><td>length</td><td>0x01BC</td></tr>
<tr><td>4</td><td>1</td><td>type</td><td>0x00</td></tr>
<tr><td>5</td><td>2</td><td>random</td><td>0x1234</td></tr>
<tr><td>7</td><td>n</td><td>JSON object</td><td>
<pre>
{
        "gateways": [{
                "gatewayID": "54D0B4FFFE3AB6CE",
                "lastSeenAt": "2021-11-18 15:34:02",
                "isOnline": true,
                "longitude": 118.03394,
                "latitude": 24.48405
        }],
        "applications": [{
                "applicationID": 1,
                "name": "烟感",
                "deviceNum": 10,
                "activatNum": 7,
                "isAutoJoin": false
        }]
}
</pre>
<table>
<tr><th></th><th>Description</th><th>Types</th></tr>
<tr><td>gateways</td><td>Gateway information array</td><td></td></tr>
<tr><td>- gatewayID</td><td>Gateway unique code</td><td>string</td></tr>
<tr><td>- lastSeenAt</td><td>Last uplink time of the gateway</td><td>string</td></tr>
<tr><td>- isOnline</td><td>Online status (true: online, false: offline)</td><td>bool</td></tr>
<tr><td>- longitude</td><td>Longitude</td><td>float64</td></tr>
<tr><td>- latitude</td><td>Latitude</td><td>float64</td></tr>
<tr><td>applications</td><td>Application information array</td><td></td></tr>
<tr><td>- applicationID</td><td>Application ID</td><td>int</td></tr>
<tr><td>- name</td><td>Application name</td><td>string</td></tr>
</table>
</td></tr>
</table>

				- deviceNum	Total number of devices under this application	int	
				- activatNum	Number of activated (joined) devices	int	
				- isAutoJoin	Whether the application allows automatic device addition during network activation	bool	

4.1.4.4 HTTP Push Data Format

- HTTP is configured for each application at the path: LoRa Network Server → Application → View (corresponding APP) → Interface Management.
- The data content pushed by HTTP is in JSON format, consistent with the JSON content of MQTT and TCP methods (please refer to the previous two chapters).
- When JavaScript function parsing is configured, the JSON data will no longer use the default data format but will use the converted data format.
- HTTP only supports pushing and does not support downstream data.

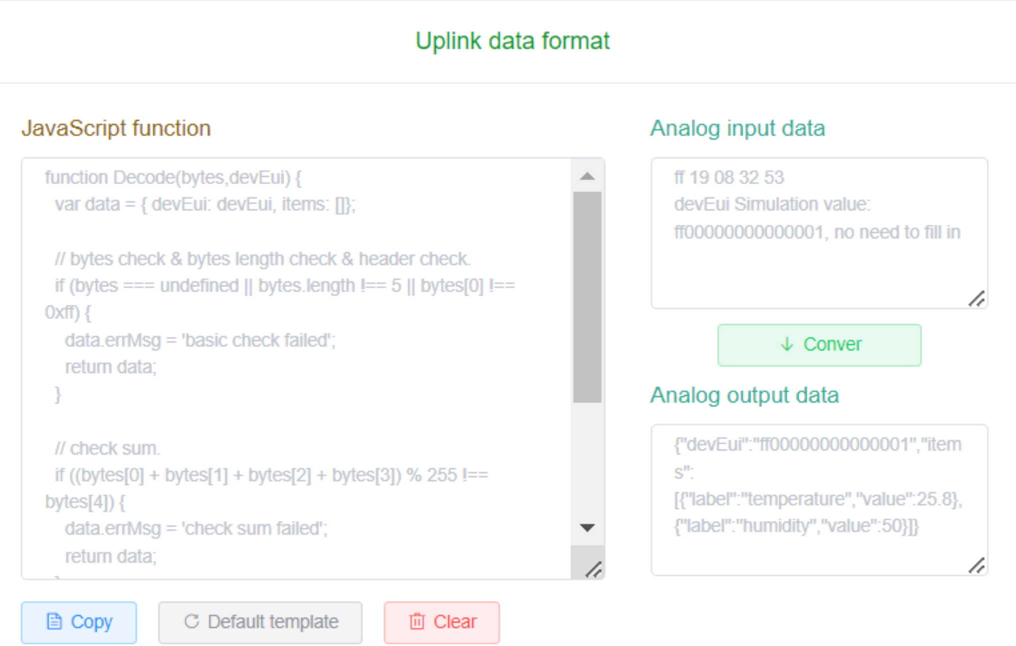
4.1.4.5 JavaScript Function Transformation Method

- The function of this transformation method:
  - When receiving uplink data, it converts the hexadecimal data (or string) reported by the device into JSON format corresponding field data. This allows integration with specific platforms without the need for customization.
  - When receiving downlink data, it converts the JSON data sent by the customer platform into corresponding hexadecimal data, and then sends this hexadecimal data to the device.
- The transformation function is not configured by default, and the default data format is used when not configured.
- The gateway supports function transformation for both uplink and downlink data, and it is disabled by default.

1. **Uplink Data Transformation**

- When the device reports data as the hexadecimal number ff 19 08 32 53, it can be transformed into the following JSON data: {"devEui":"ff00000000000001","items":[{"label":"temperature","value":25.8},{"label":"humidity","value":50}]} (where ff is the fixed header of the protocol, 19 is the integer part of the temperature value, 08 is the decimal part of the temperature value, 32 is the humidity value, and 53 is the checksum). After successful configuration, the JSON-formatted data received by the client will be as described above.

- **Configuration Path:** LoRa Network Server → Interface → Data Conversion → Upstream Data Format



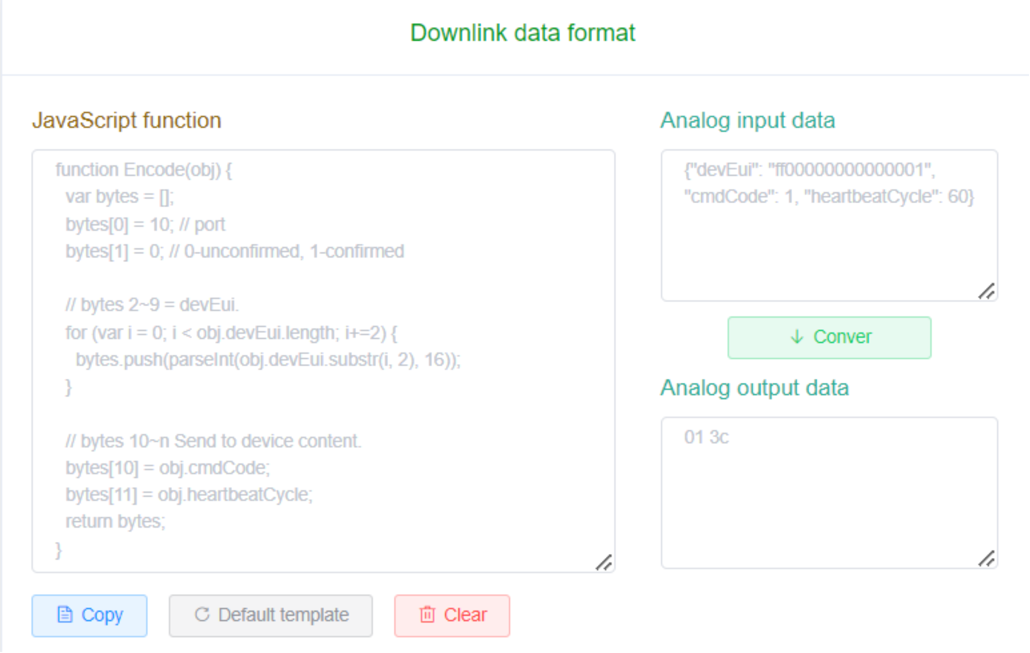
```
function Decode(bytes,devEui) {
  var data = { devEui: devEui, items: []};

  // bytes check & bytes length check & header check.
  if (bytes === undefined || bytes.length !== 5 || bytes[0] !==
0xff) {
    data.errMsg = 'basic check failed';
    return data;
  }

  // check sum.
  if ((bytes[0] + bytes[1] + bytes[2] + bytes[3]) % 255 !==
bytes[4]) {
    data.errMsg = 'check sum failed';
    return data;
```

```
ff 19 08 32 53
devEui Simulation value:
ff00000000000001, no need to fill in
```

```
{"devEui":"ff00000000000001","items":
[{"label":"temperature","value":25.8},
{"label":"humidity","value":50}]}
```

2. Downlink Data Transformation

- When the device sends data {"devEui": "ff00000000000001", "cmdCode": 1, "heartbeatCycle": 60} (after function transformation) it will be converted to 01 3c (01-command code for heartbeat cycle configuration, 3c-heartbeat cycle value), which will be sent to the terminal.
- Configuration Path: LoRa Network Server → Interface → Data Transformation → Downlink Data Format



```
function Encode(obj) {
  var bytes = [];
  bytes[0] = 10; // port
  bytes[1] = 0; // 0-unconfirmed, 1-confirmed

  // bytes 2~9 = devEui.
  for (var i = 0; i < obj.devEui.length; i+=2) {
    bytes.push(parseInt(obj.devEui.substr(i, 2), 16));
  }

  // bytes 10~n Send to device content.
  bytes[10] = obj.cmdCode;
  bytes[11] = obj.heartbeatCycle;
  return bytes;
}
```

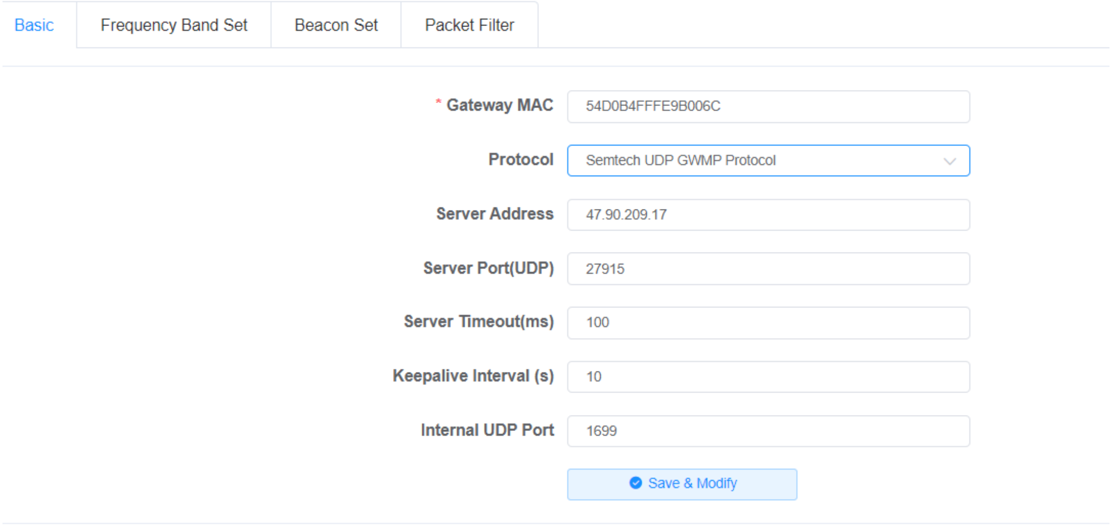
```
{"devEui": "ff00000000000001",
"cmdCode": 1, "heartbeatCycle": 60}
```

```
01 3c
```

4.1.5 Common Platform Integration

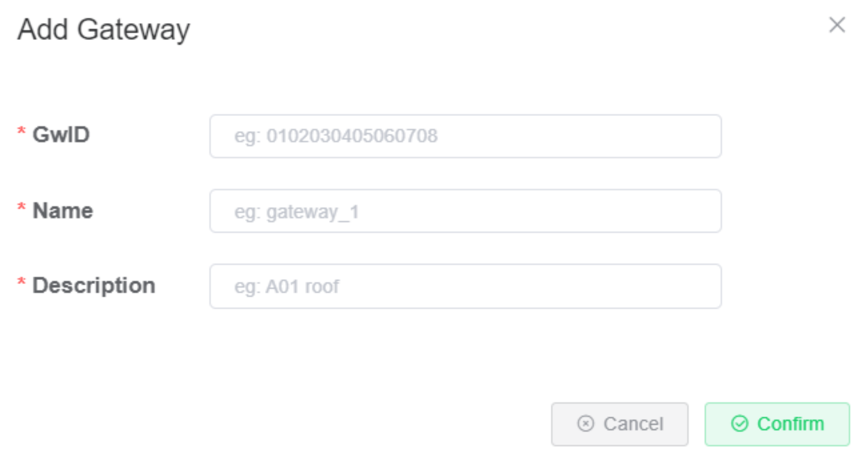
4.1.5.1 Four-Faith Cloud NS

- The standard NS used by Four-Faith Cloud adopts the Semtech UDP GWMP Protocol.
- In this mode, the gateway implements data forwarding functionality.
- Configuration path: LoRa Gateway → Basic Settings. The main configurations include protocol, server address, and server port (UDP). The specific configurations are as follows:



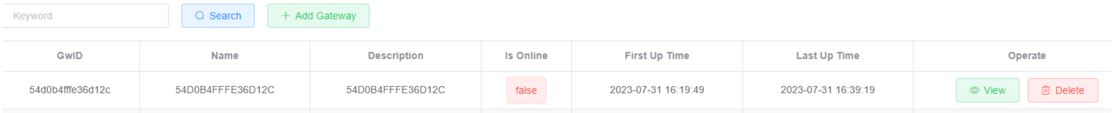
* Gateway MAC	54D0B4FFFE9B006C
Protocol	Semtech UDP GWMP Protocol
Server Address	47.90.209.17
Server Port(UDP)	27915
Server Timeout(ms)	100
Keepalive Interval (s)	10
Internal UDP Port	1699

- **Open CSTool:** http://47.90.209.17:51868/#/ns/gateways
- **Create a gateway.**


Add Gateway

* GwID	eg: 0102030405060708
* Name	eg: gateway_1
* Description	eg: A01 roof

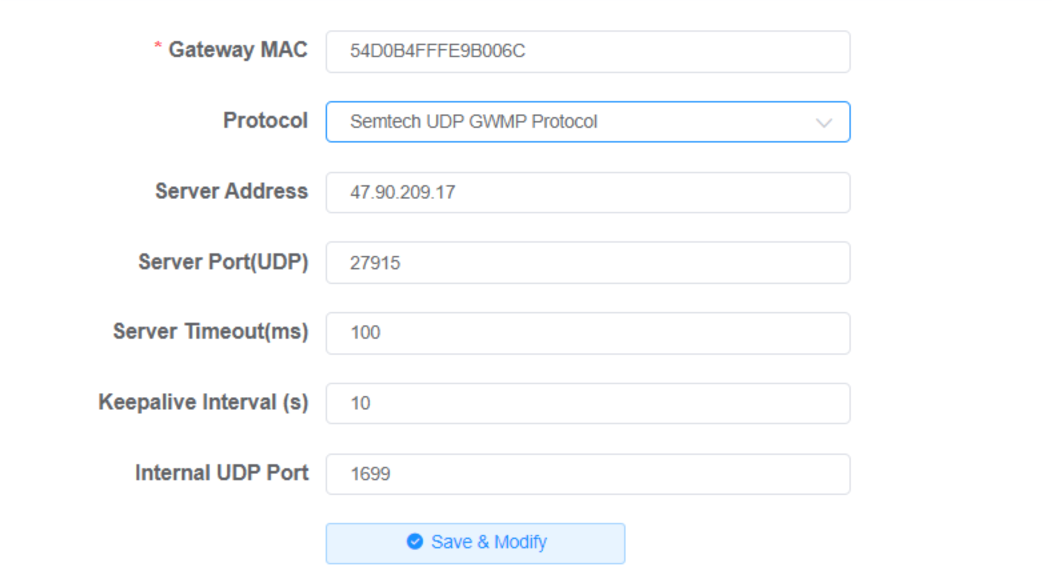
- Check the gateway status, as shown in the figure below, indicating that the gateway is already online.



GwID	Name	Description	Is Online	First Up Time	Last Up Time	Operate
54d0b4fffe36d12c	54D0B4FFFE36D12C	54D0B4FFFE36D12C	false	2023-07-31 16:19:49	2023-07-31 16:39:19	View Delete

4.1.5.2 ChirpStack Platform（GWMP）

- ChirpStack is a general open-source NS that supports multiple access methods, commonly used for GWMP protocol access.
- **Configuration path:** LoRa Gateway → Basic Settings, mainly configuring the protocol, server address, and server port (UDP). The specific configuration is as follows:



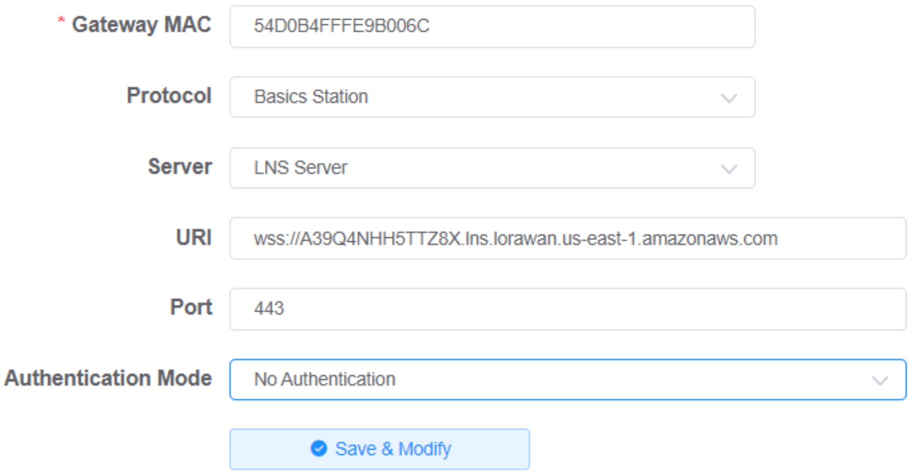
* Gateway MAC	54D0B4FFFE9B006C
Protocol	Semtech UDP GWMP Protocol
Server Address	47.90.209.17
Server Port(UDP)	27915
Server Timeout(ms)	100
Keepalive Interval (s)	10
Internal UDP Port	1699

4.1.5.3 ChirpStack Platform（LNS）

ChirpStack can be configured for Basicstation protocol access, which is generally used as LNS. It supports modes such as No Authentication or TLS Server Authentication. The following provides examples for configuring access in both ways.

1. LNS - No Authentication

- By configuring the protocol, server protocol type, URI, port, and mode selection, you can modify the settings successfully.



* Gateway MAC: 54D0B4FFFE9B006C

Protocol: Basics Station

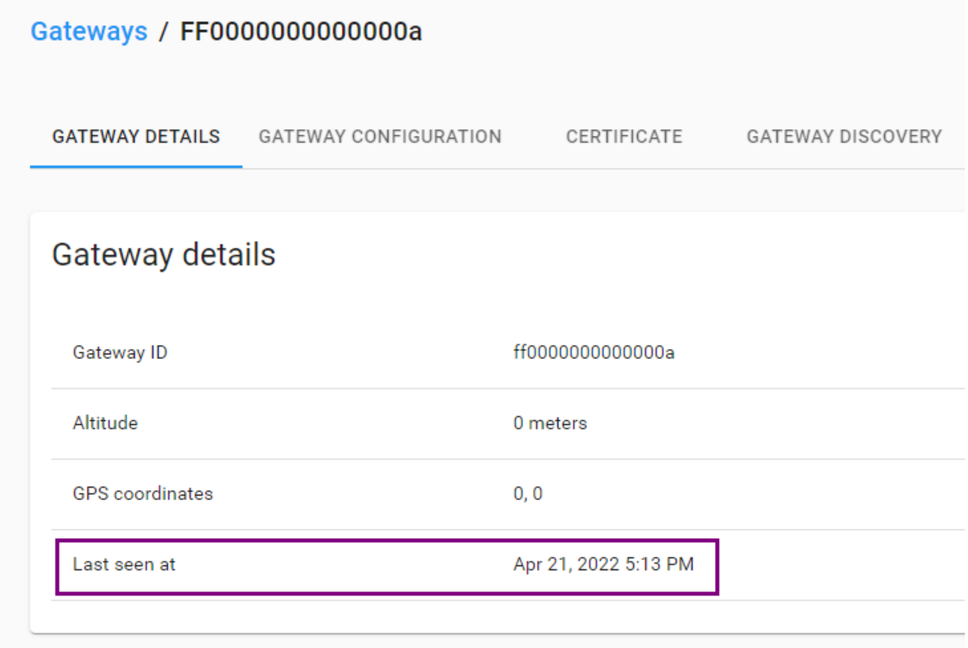
Server: LNS Server

URI: wss://A39Q4NHH5TTZ8X.lns.lorawan.us-east-1.amazonaws.com

Port: 443

Authentication Mode: No Authentication

- On the platform, the "Last seen at" for the gateway indicates the connection status of the gateway.

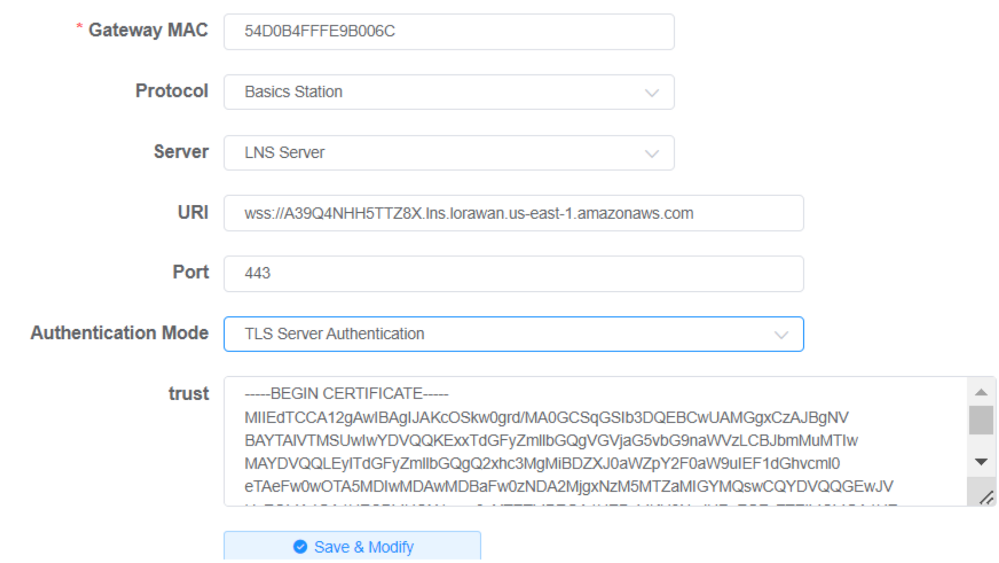


Gateway details

Gateway ID	ff0000000000000a
Altitude	0 meters
GPS coordinates	0, 0
Last seen at	Apr 21, 2022 5:13 PM

2. LNS - TLS Server Authentication

- When configuring the gateway, the URI requires the corresponding domain name of the server, and the "trust" content is derived from the server's .pem file.

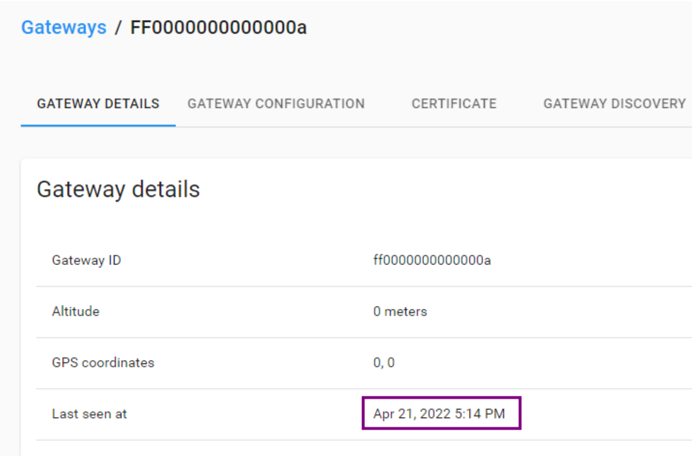


* Gateway MAC	54D0B4FFFE9B006C
Protocol	Basics Station
Server	LNS Server
URI	wss://A39Q4NHH5TTZ8X.lns.lorawan.us-east-1.amazonaws.com
Port	443
Authentication Mode	TLS Server Authentication

trust

```
-----BEGIN CERTIFICATE-----
MIIEdTCCA12gAwIBAgIJAKcOSkw0grd/MA0GCSqGSIb3DQEBCwUAMGgxCzAJBgNV
BAYTAlVTMSUwIwYDVQQKExxTdGFyZmllbGQgVGVjaG5vbG9naWVzLCBJbmMuMTIw
MAYDVQQLEylTdGFyZmllbGQgQ2xhc3MgMiBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0
eTAeFw0wOTA5MDIwMDAwMDBaFw0zNDA2MjgxNzM5MTZaMIGYMQswCQYDVQQGEwJV
```

- The "Last seen at" of the gateway on the platform indicates the connection status of the gateway.



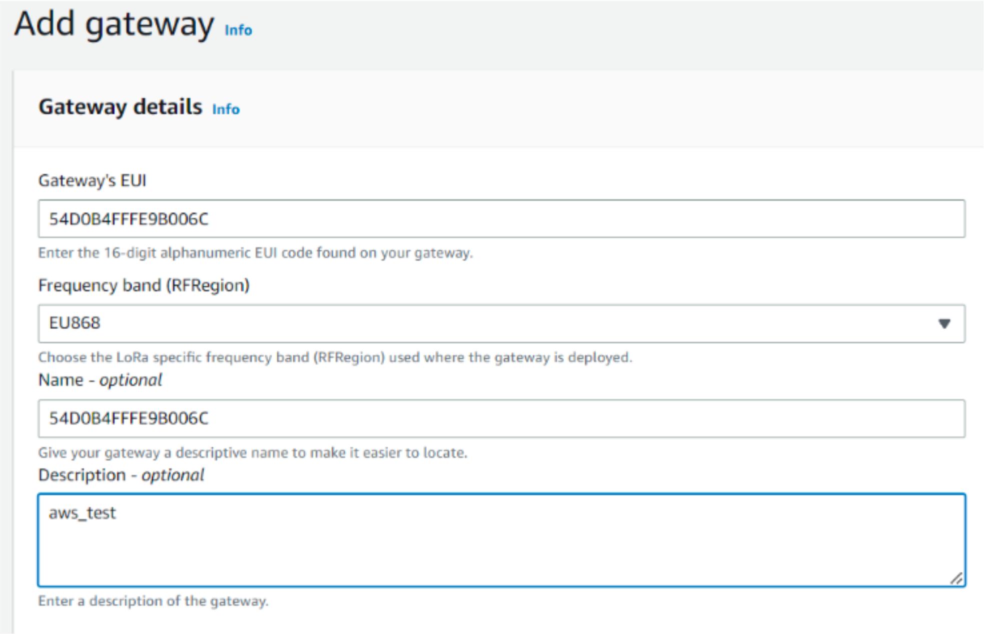
Gateways / FF0000000000000a

Gateway details

Gateway ID	ff0000000000000a
Altitude	0 meters
GPS coordinates	0, 0
Last seen at	Apr 21, 2022 5:14 PM

4.1.5.4 AWS Platform（LNS）

- Creating a gateway on the AWS platform



Add gateway Info

Gateway details Info

Gateway's EUI

54D0B4FFFE9B006C

Enter the 16-digit alphanumeric EUI code found on your gateway.

Frequency band (RFRegion)

EU868

Choose the LoRa specific frequency band (RFRegion) used where the gateway is deployed.

Name - *optional*

54D0B4FFFE9B006C

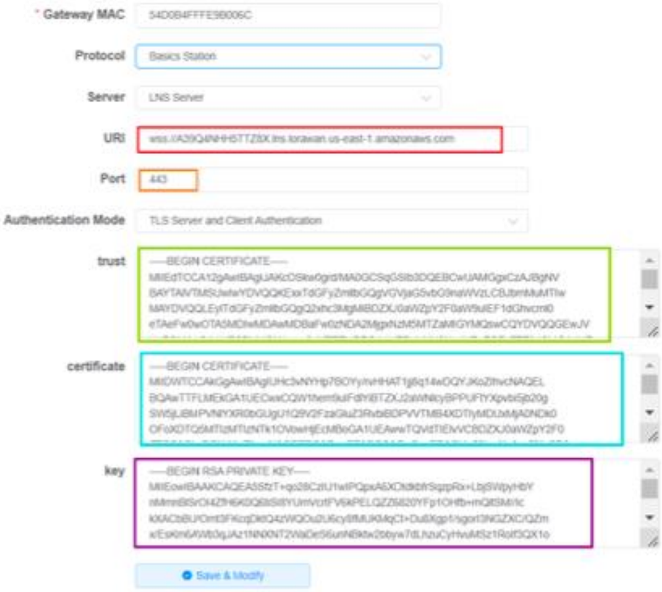
Give your gateway a descriptive name to make it easier to locate.

Description - *optional*

aws_test

Enter a description of the gateway.

- Download the corresponding key generated by the gateway and configure the corresponding parameters for the gateway. Select the mode as TLS Server and Client Authentication. Boxes of the same color in the following image represent identical content.



* Gateway MAC: 54D0B4FFFE9B006C

Protocol: Basics Station

Server: LNS Server

URI: wss://A39Q4NHH5TTZ8X.lns.lorawan.us-east-1.amazonaws.com

Port: 443

Authentication Mode: TLS Server and Client Authentication

trust:

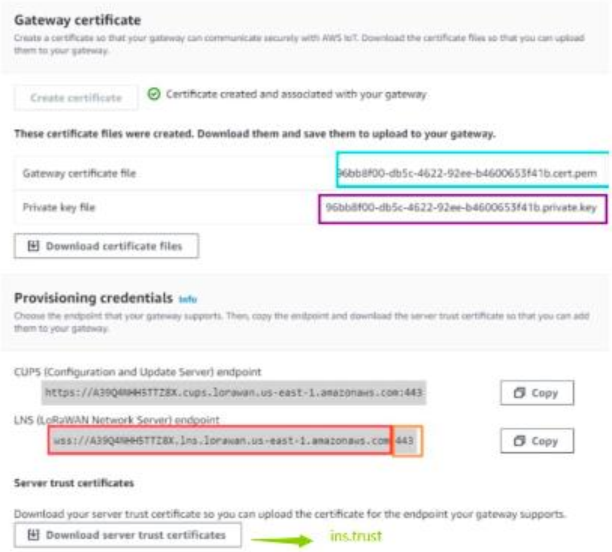
```
-----BEGIN CERTIFICATE-----
MIIEdTCCA12gAwIBAgIJAKzOSkw0grd/MA0GCSqGSIb3DQEBCwUAMGgxCzAJBgNV
BAYTAlVTMSUwIwYDVQQKExxTdGFyZmllbGQgVGVjaG5vbG9naWVzLCBJbmMuMTIw
MAYDVQQLEylTdGFyZmllbGQgQ2xhc3MgMiBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0
eTAeFw0wOTA5MDIwMDAwMDBaFw0zNDA2MjgxNzM5MTZaMGYxMQswCQYDVQQGEwJV
```

certificate:

```
-----BEGIN CERTIFICATE-----
MIIDWTCCAkGgAwIBAgIUHc3vNYHp7BOYy/nvHHAT1g6q14wDQYJKoZIhvcNAQEL
BQAwTTFLMEkGA1UECwxCQW1hem9uIFdlYiBTZXJ2aWNlcyBPPUFtYXpvbi5jb20g
SW5jLiBMPVNlYXR0bGUgU1Q9V2FzaGluZ3RvbiBDPVVTMB4XDTIyMDUxMjA0NDk0
OFoXDTQ5MTIzMTIzNTk1OVowHjEcMBoGA1UEAwwTQVdTIElvVCBDZXJ0aWZpY2F0
```

key:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEA3SfzT+qo28CztU1wIPQpxA6XCh8kbfrSqzpRx+LbjSWpyHbY
nMmnBiSrOi4ZfH6KDQ6bSi8YUmVctFV6kPELQZZ6820YFp1OHfb+mQtSMi/c
kKACbBU/Omt3FKcqDkt0Q4zWQOu2U6cy8fMUKMqCt+Du8Xgp1/sgot3NGZXC/QZm
x/EsKm6AWb0qJAz1NNXNT2WaDe56unNBktw2bbyw7dLhzuCyHwM5z1Rolf3QX1o
```



Gateway certificate

Create a certificate so that your gateway can communicate securely with AWS IoT. Download the certificate files so that you can upload them to your gateway.

Certificate created and associated with your gateway

These certificate files were created. Download them and save them to upload to your gateway.

Gateway certificate file	96bb8f00-db5c-4622-92ee-b4600653f41b.cert.pem
Private key file	96bb8f00-db5c-4622-92ee-b4600653f41b.private.key

Provisioning credentials Info

Choose the endpoint that your gateway supports. Then, copy the endpoint and download the server trust certificate so that you can add them to your gateway.

CUPS (Configuration and Update Server) endpoint

```
https://A39Q4NHH5TTZ8X.cups.lorawan.us-east-1.amazonaws.com:443
```

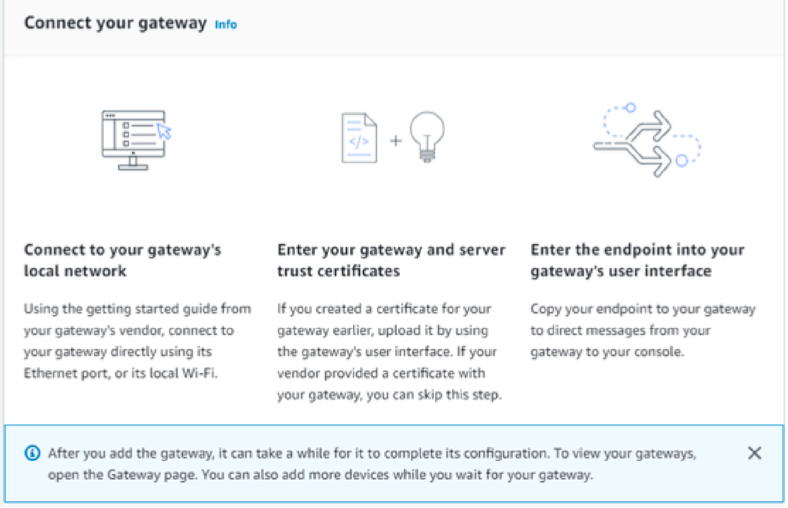
LNS (LoRaWAN Network Server) endpoint

```
wss://A39Q4NHH5TTZ8X.lns.lorawan.us-east-1.amazonaws.com:443
```

Server trust certificates

Download your server trust certificate so you can upload the certificate for the endpoint your gateway supports.

- Continue to complete the creation of the gateway.



Connect your gateway Info

Connect to your gateway's local network

Using the getting started guide from your gateway's vendor, connect to your gateway directly using its Ethernet port, or its local Wi-Fi.

Enter your gateway and server trust certificates

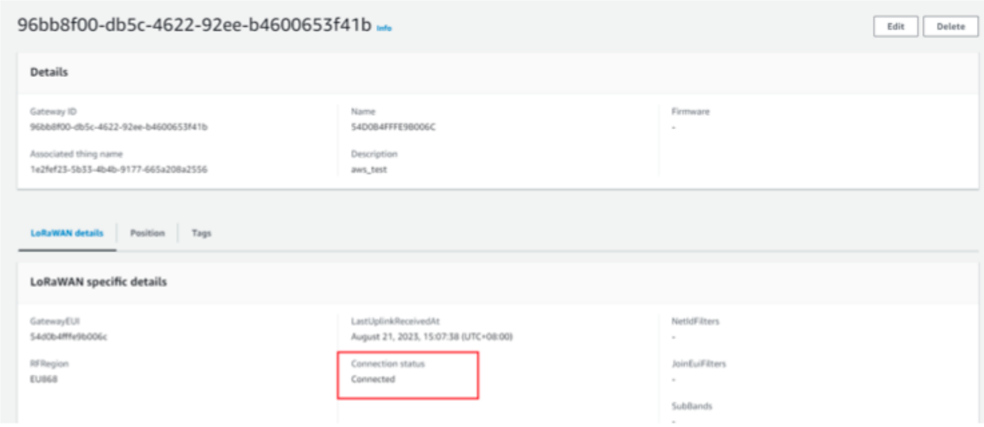
If you created a certificate for your gateway earlier, upload it by using the gateway's user interface. If your vendor provided a certificate with your gateway, you can skip this step.

Enter the endpoint into your gateway's user interface

Copy your endpoint to your gateway to direct messages from your gateway to your console.

After you add the gateway, it can take a while for it to complete its configuration. To view your gateways, open the Gateway page. You can also add more devices while you wait for your gateway.

- After successfully configuring the gateway, you can see the connection status of the gateway on the AWS platform.



96bb8f00-db5c-4622-92ee-b4600653f41b Info

Details

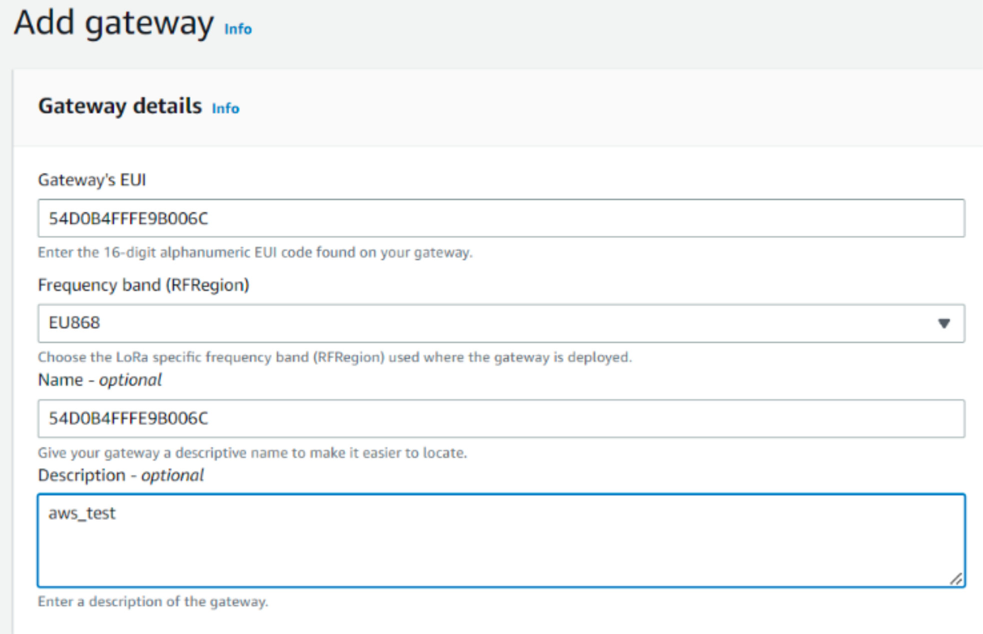
Gateway ID 96bb8f00-db5c-4622-92ee-b4600653f41b	Name 54D0B4FFFE9B006C	Firmware -
Associated thing name 1e2fef23-5b33-4b4b-9177-665a208a2556	Description aws_test	

LoRaWAN specific details

GatewayEUI 54d0b4fffe9b006c	LastUplinkReceivedAt August 21, 2023, 15:07:38 (UTC+08:00)	NetIdFilters -
RFRegion EU868	Connection status Connected	JoinEuiFilters -
		SubBands -

4.1.5.5 AWS Platform（CPUS）

- Create a gateway on the AWS platform.



Add gateway Info

Gateway details Info

Gateway's EUI: 54D0B4FFFE9B006C

Enter the 16-digit alphanumeric EUI code found on your gateway.

Frequency band (RFRegion): EU868

Choose the LoRa specific frequency band (RFRegion) used where the gateway is deployed.

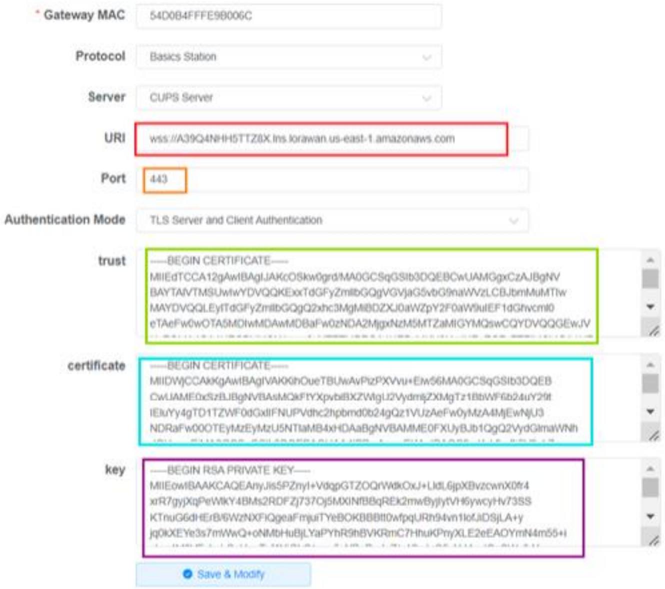
Name - *optional*: 54D0B4FFFE9B006C

Give your gateway a descriptive name to make it easier to locate.

Description - *optional*: aws_test

Enter a description of the gateway.

- Download the corresponding key generated by the gateway and configure the corresponding parameters for the gateway. Select the mode as TLS Server and Client Authentication. The boxes with the same color in the following image represent identical content.



* Gateway MAC: 54D0B4FFFE9B006C

Protocol: Basics Station

Server: CUPS Server

URI: wss://A39Q4NHH5TTZ8X.lns.lorawan.us-east-1.amazonaws.com

Port: 443

Authentication Mode: TLS Server and Client Authentication

trust:

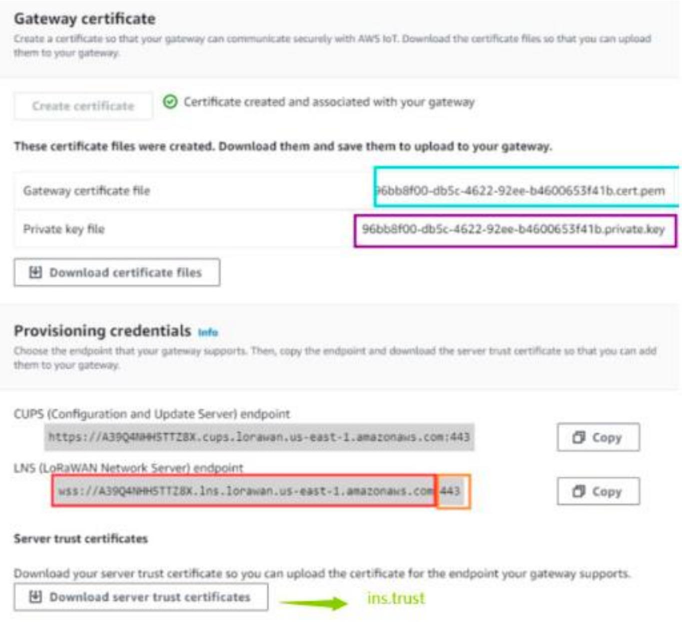
```
-----BEGIN CERTIFICATE-----
MIIEdTCCA12gAwIBAgIJAKcOSkw0grd/MA0GCSqGSIb3DQEBCwUAMGgxCzAJBgNV
BAYTAlVTMSUwIwYDVQQKExxTdGFyZmllbGQgVGVjaG5vbG9naWVzLCBJbmMuMTIw
MAYDVQQLEylTdGFyZmllbGQgQ2xhc3MgMiBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0
eTAeFw0wOTA5MDIwMDAwMDBaFw0zNDA2MjgxNzM5MTZaMIGYMQswCQYDVQQGEwJV
```

certificate:

```
-----BEGIN CERTIFICATE-----
MIIDWjCCAkKgAwIBAgIVAKKhOueTBUwAvPizPXVvu+Ew56MA0GCSqGSIb3DQEB
CwUAME0xSzBJBgNVBAsMQkFtYXpvbiBXZWIgU2VydmljZXMgTz1BbWF6b24uY29t
IEluYy4gTD1TZWF0dGxlIFNUPVdhc2hpbmd0b24gQz1VUzAeFw0yMzA4MjEwNjU3
NDRaFw00OTEyMzEyMzU5NTlaMB4xHDAaBgNVBAMME0FXUyBJb1QgQ2VydGlmaWNh
```

key:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAnyJis5PZnyl+VdqpGTZOQrWdkOxJ+LIdL6jpXBvzcwnX0fr4
xrR7gyjXqPeWlkY4BMs2RDFZj737Oj5MXINfBBqREk2mwByjlytVH6ywcyHv73SS
KTnuG6dHErB/6WzNXFiQgeaFmjuiTYeBOKBBBtI0wfpqURh94vn1lofJDSjLA+y
jq0kXEYe3s7mWwQ+oNMbHuBjLYaPYhR9hBVKRmC7HhuKPnyXLE2eEAOYmN4m55+i
```



Gateway certificate

Create a certificate so that your gateway can communicate securely with AWS IoT. Download the certificate files so that you can upload them to your gateway.

Create certificate ✓ Certificate created and associated with your gateway

These certificate files were created. Download them and save them to upload to your gateway.

Gateway certificate file	96bb8f00-db5c-4622-92ee-b4600653f41b.cert.pem
Private key file	96bb8f00-db5c-4622-92ee-b4600653f41b.private.key

Download certificate files

Provisioning credentials Info

Choose the endpoint that your gateway supports. Then, copy the endpoint and download the server trust certificate so that you can add them to your gateway.

CUPS (Configuration and Update Server) endpoint

```
https://A39Q4NHHSTTZ8X.cups.lorawan.us-east-1.amazonaws.com:443
```

LNS (LoRaWAN Network Server) endpoint

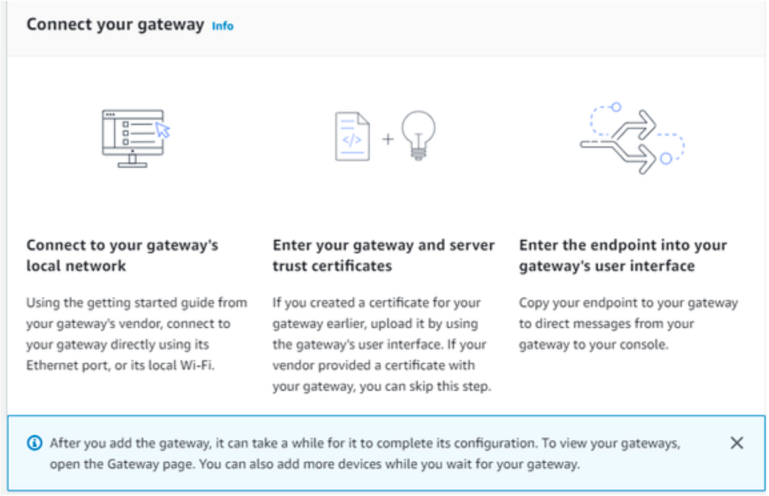
```
wss://A39Q4NHHSTTZ8X.lns.lorawan.us-east-1.amazonaws.com:443
```

Server trust certificates

Download your server trust certificate so you can upload the certificate for the endpoint your gateway supports.

Download server trust certificates

- Continue to complete the creation of the gateway.



Connect your gateway Info

Connect to your gateway's local network

Using the getting started guide from your gateway's vendor, connect to your gateway directly using its Ethernet port, or its local Wi-Fi.

Enter your gateway and server trust certificates

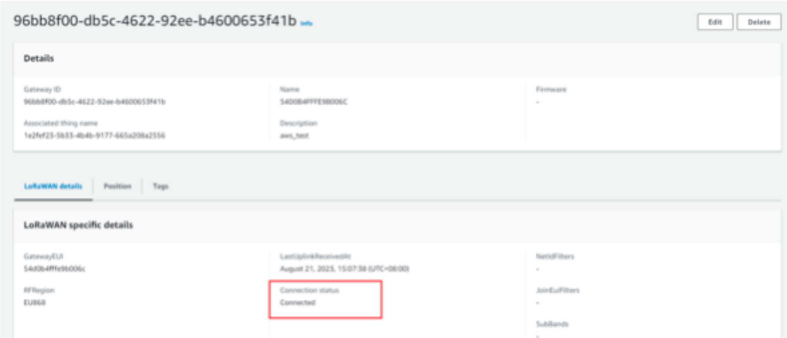
If you created a certificate for your gateway earlier, upload it by using the gateway's user interface. If your vendor provided a certificate with your gateway, you can skip this step.

Enter the endpoint into your gateway's user interface

Copy your endpoint to your gateway to direct messages from your gateway to your console.

After you add the gateway, it can take a while for it to complete its configuration. To view your gateways, open the Gateway page. You can also add more devices while you wait for your gateway.

- After successfully configuring the gateway, you can view the gateway's connection status on the AWS platform.



96bb8f00-db5c-4622-92ee-b4600653f41b Info

Details

Gateway ID 96bb8f00-db5c-4622-92ee-b4600653f41b	Name 54D0B4FFFE9B006C	Firmware -
Associated thing name 1e2fef23-5b33-4b4b-9177-665a208a2556	Description aws_test	

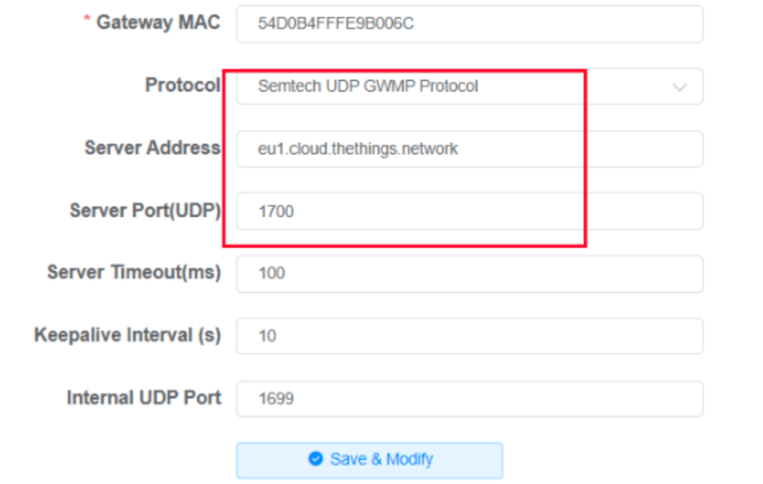
LoRaWAN specific details

GatewayEUI 54d0b4fffe9b006c	LastUplinkReceivedAt August 21, 2023, 15:07:38 (UTC+08:00)	NetIdFilters -
RFRegion EU868	Connection status Connected	JoinEuiFilters -
		SubBands -

4.1.5.6 TTN Platform（GWMP）

- The TTN platform supports both GWMP and Basicstation modes of access.
- When using the GWMP protocol, the configuration is the same as other platforms,

requiring only the server IP and port settings.



* Gateway MAC: 54D0B4FFFE9B006C

Protocol: Semtech UDP GWMP Protocol

Server Address: eu1.cloud.thethings.network

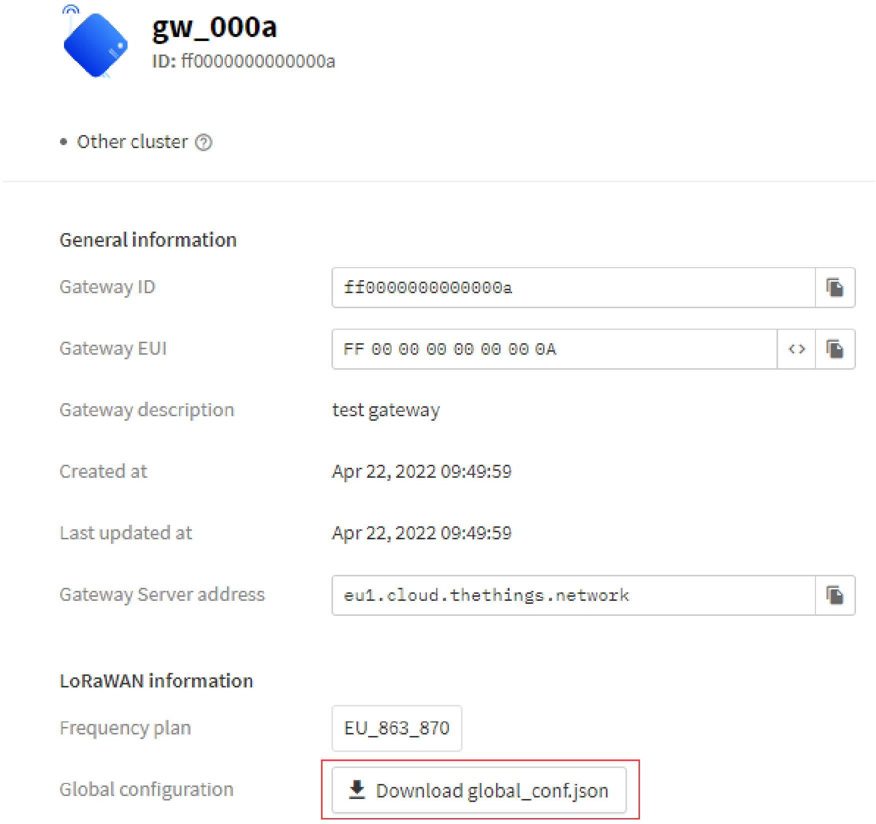
Server Port(UDP): 1700

Server Timeout(ms): 100

Keepalive Interval (s): 10

Internal UDP Port: 1699

- The server address and port information can be obtained from the global_conf.json file, which can be downloaded from the TTN platform.



gw_000a

ID: ff0000000000000a

• Other cluster

General information

Gateway ID: ff0000000000000a

Gateway EUI: FF 00 00 00 00 00 00 0A

Gateway description: test gateway

Created at: Apr 22, 2022 09:49:59

Last updated at: Apr 22, 2022 09:49:59

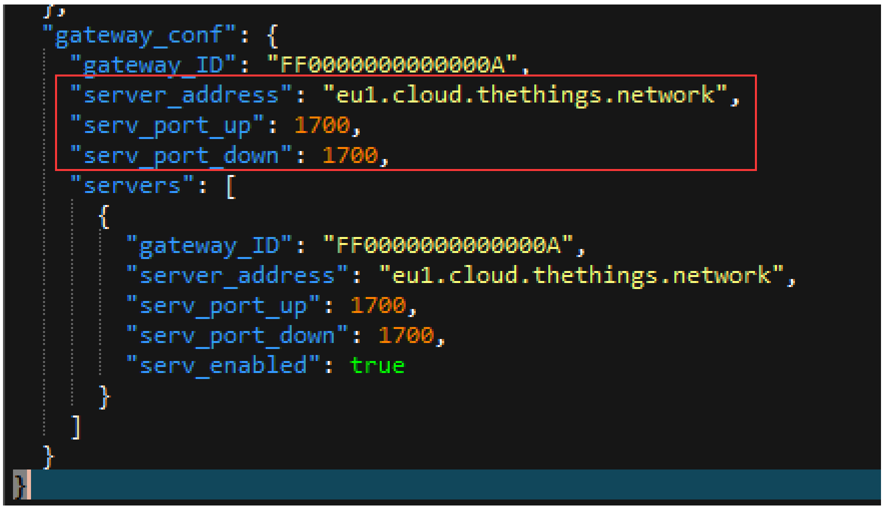
Gateway Server address: eu1.cloud.thethings.network

LoRaWAN information

Frequency plan: EU_863_870

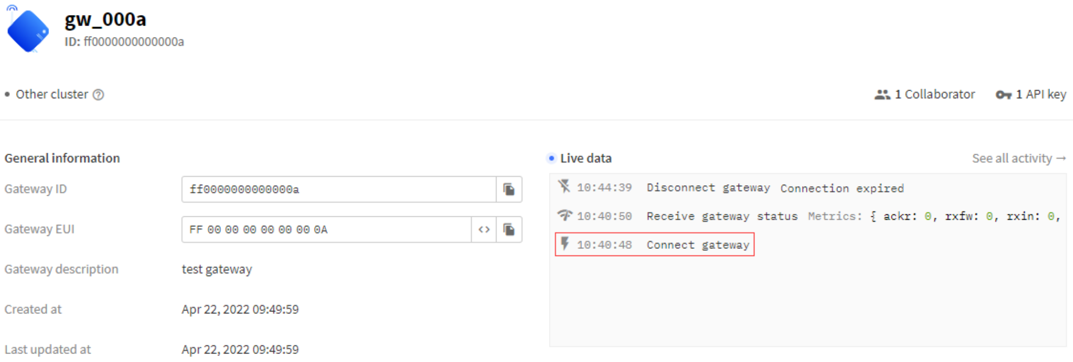
Global configuration: Download global_conf.json

- The server address and port can be found at the end of the file.



```
},
"gateway_conf": {
  "gateway_ID": "FF0000000000000A",
  "server_address": "eu1.cloud.thethings.network",
  "serv_port_up": 1700,
  "serv_port_down": 1700,
  "servers": [
    {
      "gateway_ID": "FF0000000000000A",
      "server_address": "eu1.cloud.thethings.network",
      "serv_port_up": 1700,
      "serv_port_down": 1700,
      "serv_enabled": true
    }
  ]
}
}
```

- After successfully configuring the gateway, you can view the connection information on the TTN platform.



gw_000a
ID: ff0000000000000a

• Other cluster

1 Collaborator   1 API key

General information

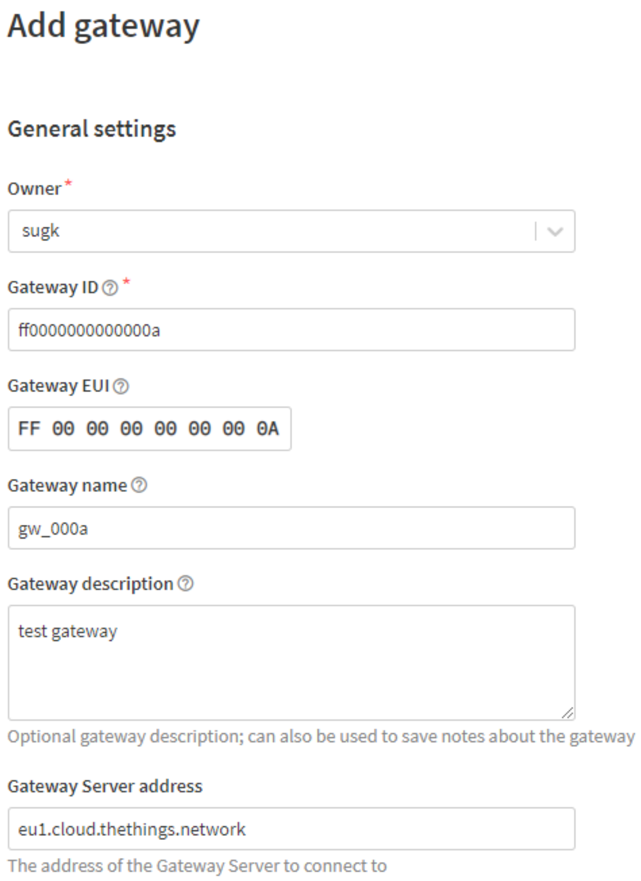
Gateway ID	ff0000000000000a
Gateway EUI	FF 00 00 00 00 00 00 0A
Gateway description	test gateway
Created at	Apr 22, 2022 09:49:59
Last updated at	Apr 22, 2022 09:49:59

• Live data   See all activity →

```
10:44:39  Disconnect gateway  Connection expired
10:40:50  Receive gateway status  Metrics: { ackr: 0, rxfw: 0, rxin: 0,
10:40:48  Connect gateway
```

4.1.5.7 TTN Platform（LNS）

- The TTN platform also supports connection using the Basicstation's LNS protocol, with the mode set to TLS Server Authentication and Client Token.
- Add the gateway.



Add gateway

General settings

Owner *

sugk

Gateway ID *

ff000000000000a

Gateway EUI

FF 00 00 00 00 00 00 0A

Gateway name

gw_000a

Gateway description

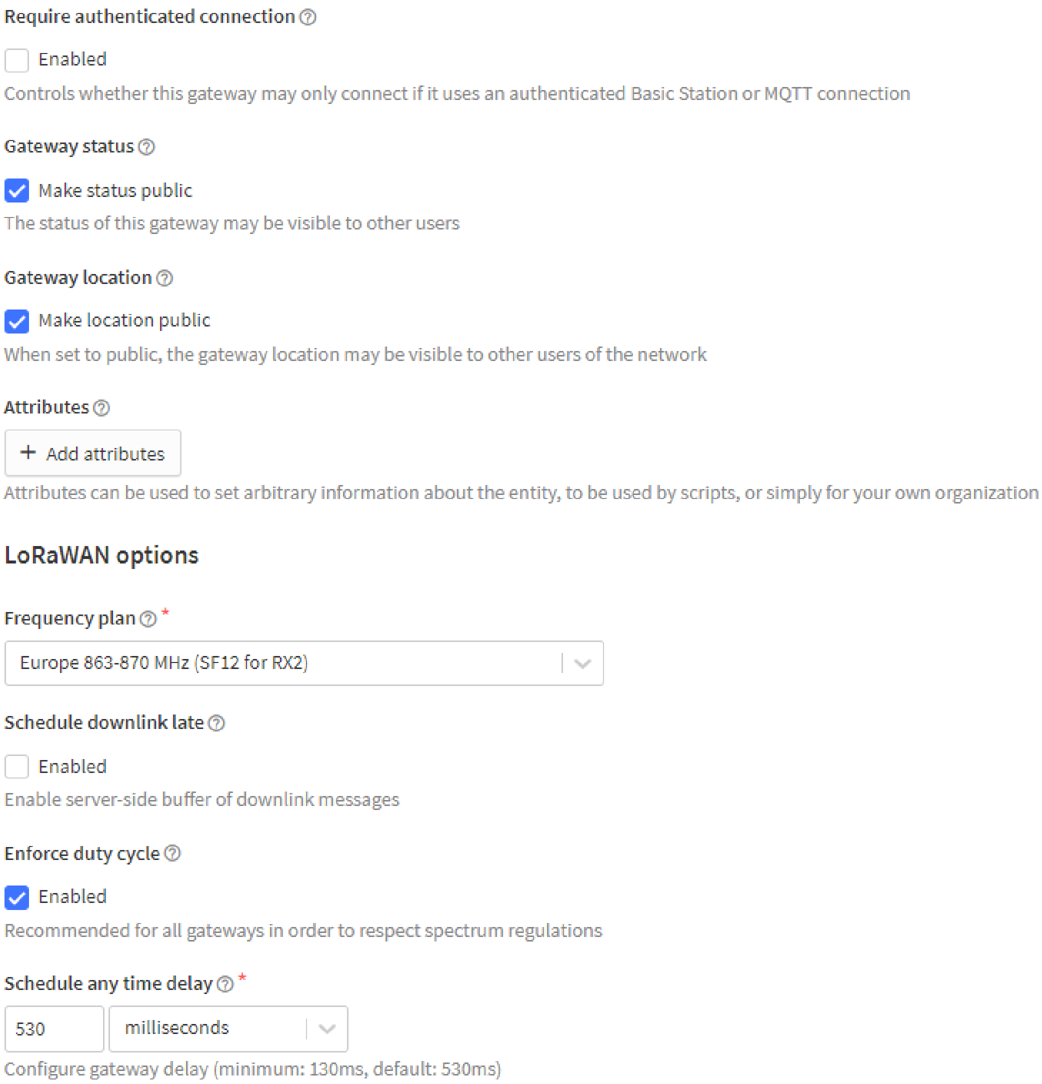
test gateway

Optional gateway description; can also be used to save notes about the gateway

Gateway Server address

eu1.cloud.thethings.network

The address of the Gateway Server to connect to



Require authenticated connection ⓘ

□ Enabled

Controls whether this gateway may only connect if it uses an authenticated Basic Station or MQTT connection

Gateway status ⓘ

☑ Make status public

The status of this gateway may be visible to other users

Gateway location ⓘ

☑ Make location public

When set to public, the gateway location may be visible to other users of the network

Attributes ⓘ

Attributes can be used to set arbitrary information about the entity, to be used by scripts, or simply for your own organization

LoRaWAN options

Frequency plan ⓘ *

Europe 863-870 MHz (SF12 for RX2)

Schedule downlink late ⓘ

□ Enabled

Enable server-side buffer of downlink messages

Enforce duty cycle ⓘ

☑ Enabled

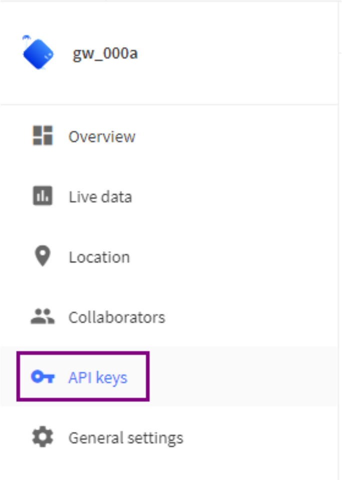
Recommended for all gateways in order to respect spectrum regulations

Schedule any time delay ⓘ *

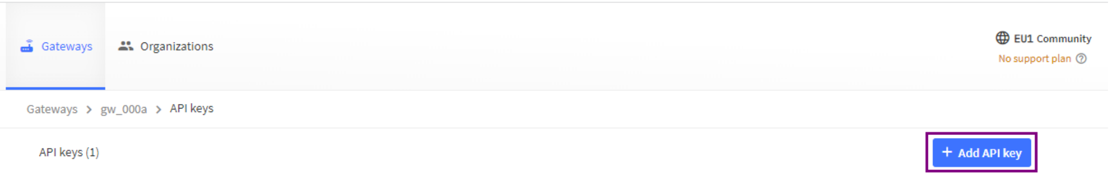
530 milliseconds

Configure gateway delay (minimum: 130ms, default: 530ms)

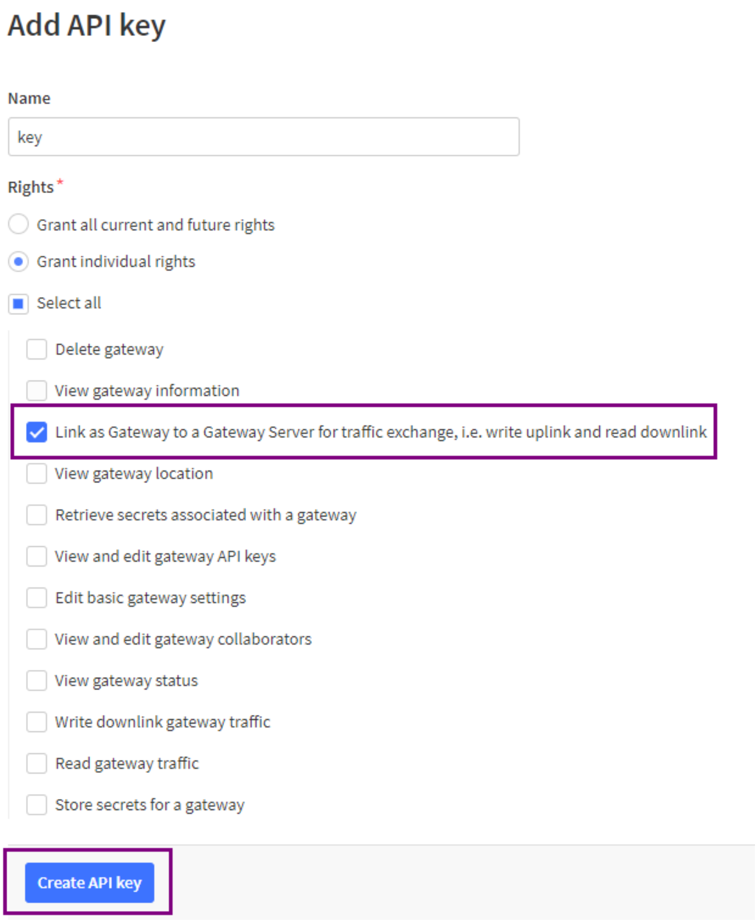
- Obtain the token value.



- Add API key



- Select according to the following image and create an API key.



Add API key

Name

key

Rights *

○ Grant all current and future rights

● Grant individual rights

■ Select all

□ Delete gateway

□ View gateway information

☑ Link as Gateway to a Gateway Server for traffic exchange, i.e. write uplink and read downlink

□ View gateway location

□ Retrieve secrets associated with a gateway

□ View and edit gateway API keys

□ Edit basic gateway settings

□ View and edit gateway collaborators

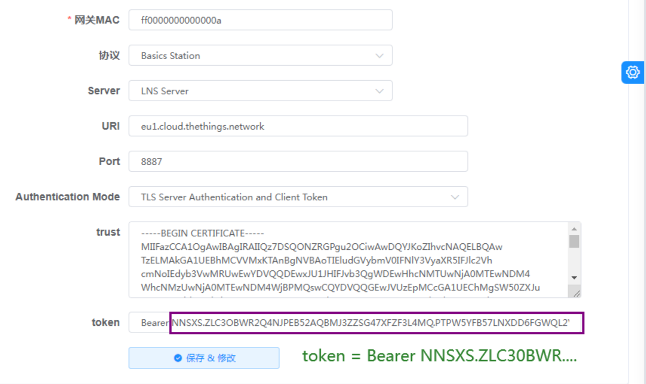
□ View gateway status

□ Write downlink gateway traffic

□ Read gateway traffic

□ Store secrets for a gateway

- Token Explanation: token = Bearer + space + API key, for example, Bearer NNSXS...



* 网关MAC: ff000000000000a

协议: Basics Station

Server: LNS Server

URI: eu1.cloud.thethings.network

Port: 8887

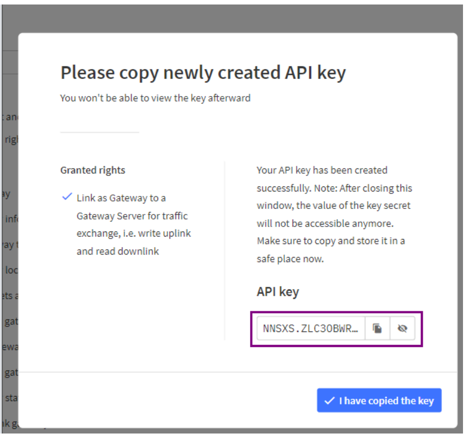
Authentication Mode: TLS Server Authentication and Client Token

trust:

```
-----BEGIN CERTIFICATE-----
MIIFazCCA1OgAwIBAgIRAIIQz7DSQONZRGPgu2OCiwAwDQYJKoZIhvcNAQELBQAw
TzELMAkGA1UEBhMCVVMxKTAnBgNVBAoTIEludGVybmV0IFNlY3VyaXR5IFJlc2Vh
cmNoIEdyb3VwMRUwEwYDVQQDEwxJU1JHIFJvb3QgWDEwHhcNMTUwNjA0MTEwNDM4
WhcNMzUwNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCcGA1UEChMgSW50ZXJu
```

token: Bearer NNSXS.ZLC3OBWR2Q4NJPEB52AQBMJ3ZZSG47XFZF3L4MQ.PTPW5YFB57LNXDD6FGWQL2'

token = Bearer NNSXS.ZLC30BWR....



Please copy newly created API key

You won't be able to view the key afterward

Granted rights

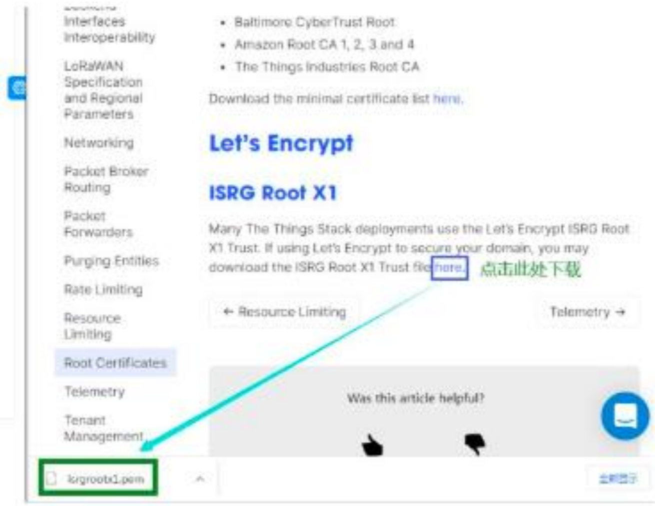
✓ Link as Gateway to a Gateway Server for traffic exchange, i.e. write uplink and read downlink

Your API key has been created successfully. Note: After closing this window, the value of the key secret will not be accessible anymore. Make sure to copy and store it in a safe place now.

API key

NNSXS.ZLC30BWR...

- **Trust Explanation:** This content is from the file isrgrootx1.pem, which can be downloaded from the TTN platform. The file download path is: https://www.thethingsindustries.com/docs/reference/root-certificates/#lets-encrypt



Interfaces Interoperability

LoRaWAN Specification and Regional Parameters

Networking

Packet Broker Routing

Packet Forwarders

Purging Entities

Rate Limiting

Resource Limiting

Root Certificates

Telemetry

Tenant Management

- Baltimore CyberTrust Root
- Amazon Root CA 1, 2, 3 and 4
- The Things Industries Root CA

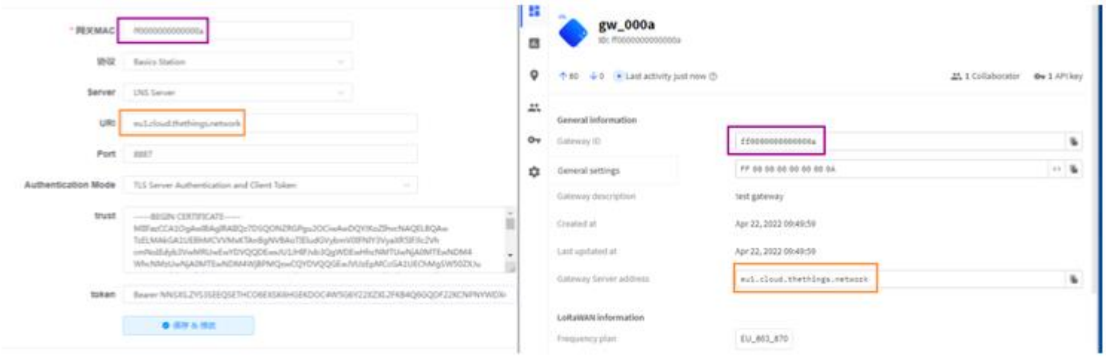
Download the minimal certificate list here.

Let's Encrypt

ISRG Root X1

Many The Things Stack deployments use the Let's Encrypt ISRG Root X1 Trust. If using Let's Encrypt to secure your domain, you may download the ISRG Root X1 Trust file here. 点击此处下载

- URI Configuration



* 网关MAC: ff0000000000000a

协议: Basics Station

Server: LNS Server

URI: eu1.cloud.thethings.network

Port: 8887

Authentication Mode: TLS Server Authentication and Client Token

trust: -----BEGIN CERTIFICATE----- [illegible]

token: Bearer NNSXS.ZYS [illegible]

gw_000a

ID: ff0000000000000a

↑ 80 ↓ 0 Last activity just now

1 Collaborator 1 API key

General information

Gateway ID: ff0000000000000a

General settings: FF 00 00 00 00 00 00 0A

Gateway description: test gateway

Created at: Apr 22, 2022 09:49:59

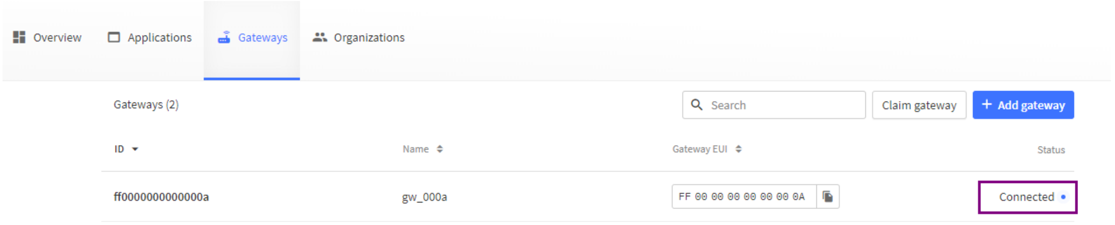
Last updated at: Apr 22, 2022 09:49:59

Gateway Server address: eu1.cloud.thethings.network

LoRaWAN information

Frequency plan: EU_863_870

- **Port Configuration:** Fixed to 8887
- After the gateway configuration is successful, you can check the gateway's connection status to determine whether the connection is successful.



Gateways (2)

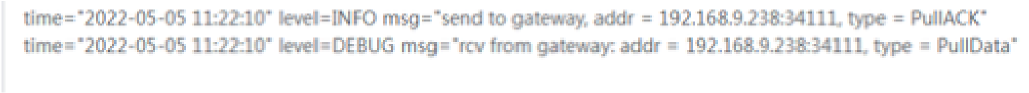
ID	Name	Gateway EUI	Status
ff0000000000000a	gw_000a	FF 00 00 00 00 00 00 0A	Connected ●

4.1.6 Common Issues:

4.1.6.1 Gateway Status

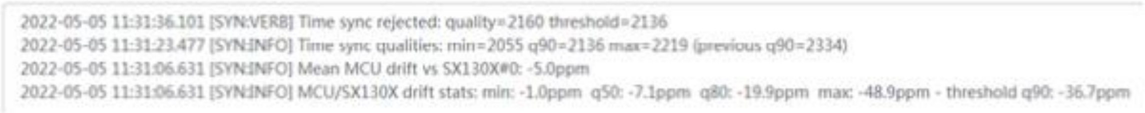
1. Internal Program Status Troubleshooting

When using Semtech UDP GWMP Protocol or Build-in LoRa Server, you can check the logs for the presence of PullData and PullACK. If there is no response after waiting for 30 seconds, it indicates an issue with the gateway.



```
time="2022-05-05 11:22:10" level=INFO msg="send to gateway, addr = 192.168.9.238:34111, type = PullACK"
time="2022-05-05 11:22:10" level=DEBUG msg="rcv from gateway: addr = 192.168.9.238:34111, type = PullData"
```

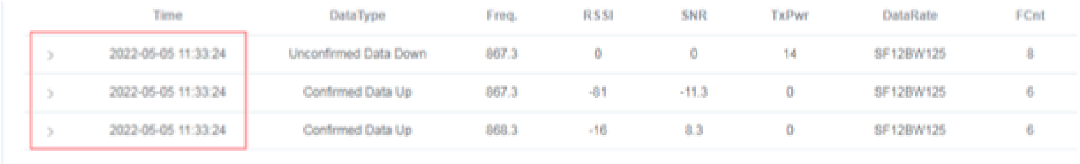
- When using Basicstation mode, you can check whether there are any logs to determine the status.



```
2022-05-05 11:31:36.101 [SYN:VERB] Time sync rejected: quality=2160 threshold=2136
2022-05-05 11:31:23.477 [SYN:INFO] Time sync qualities: min=2055 q90=2136 max=2219 (previous q90=2334)
2022-05-05 11:31:06.631 [SYN:INFO] Mean MCU drift vs SX130X#0: -5.0ppm
2022-05-05 11:31:06.631 [SYN:INFO] MCU/SX130X drift stats: min: -1.0ppm  q50: -7.1ppm  q80: -19.9ppm  max: -48.9ppm - threshold q90: -36.7ppm
```

3. Can the gateway receive RF data

- Open the LoRa Packet Logger and transmit data or initiate device activation with a device configured on the same frequency as the gateway. If the LoRa Packet Logger can capture logs, it indicates that the RF module is functioning properly.



	Time	DataType	Freq.	RSSI	SNR	TxPwr	DataRate	FCnt
›	2022-05-05 11:33:24	Unconfirmed Data Down	867.3	0	0	14	SF12BW125	8
›	2022-05-05 11:33:24	Confirmed Data Up	867.3	-81	-11.3	0	SF12BW125	6
›	2022-05-05 11:33:24	Confirmed Data Up	868.3	-16	8.3	0	SF12BW125	6

4.1.6.2 Communication Device

1. Abnormal Reception of Uplink Data

- **Antenna Verification:** Ensure that the antennas on both the gateway and the terminal are set to the correct frequency band. Are the antennas installed correctly? Is the feeder line of the gateway installed correctly?
- **Frequency Point Confirmation:** Compare the frequency points configured on the device with those configured on the gateway to ensure consistency.
- **Gateway LoRa Packet Logger:** Open the LoRa packet logger on the gateway, have the terminal send data, or initiate network joining to see if the gateway can listen to the terminal's data.

2. Not receiving downlink data

- Confirm the correctness of antennas on both the gateway and the device. Check if they are operating on the correct frequency band and if the antenna connections are secure.
- Examine the packet logger to determine if there are logs indicating downlink data.
  - For Class A devices, downlink data transmission occurs after an uplink transmission from the device.
  - For Class C devices, downlink data is sent immediately.

- Verify if the frequency and data rate of the downlink data match the frequency and data rate that the device is listening on. (For Four-Faith modules, you can set DBL=2 to observe this.)
- Ensure that the device type is consistent between the device and the server:
  - For Class A devices, if the server is Class C, the data is sent immediately, but the device may not be in the receive window, resulting in data loss.
  - For Class C devices, if the server is Class A, the sent data won't be understood by the device until it sends an uplink again. Without an uplink, the device won't receive the data.

  After adjusting the device or server device type, the device needs to be reconnected to synchronize.

4.1.6.3 Device Joining Exception

- First, check whether the gateway can receive the join request packet initiated by the device. If it cannot be received, please refer to the "Communication with Devices Troubleshooting."
- Embedded NS
  - Check whether the device has been entered into the embedded NS or if automatic device addition is enabled.
  - For automatic device addition, verify if the AppEUI and AppKey are consistent.
  - For devices that have been entered, confirm if the AppKey is consistent.
- External NS Server
  - Check whether the device has been added to the platform.
  - Verify if the AppEUI and AppKey of the device are consistent. AppKey is a mandatory verification field, and AppEUI verification depends on the platform requirements.
- If the gateway can see the Join Accept downlink packet but the terminal does not receive it, check if the device's frequency band matches the NS frequency band. Inconsistency can result in the listening frequency or rate not matching the downlink, causing data to be unable to be received properly.

Note: The failure of device joining is not related to inconsistent device types. For example, if the device is class A and the server is class C, the joining process can still be successful.

4.1.6.4 Customer Platform Integration

- You can use the gateway's network diagnostic tool to ping the server's IP and check if the gateway network is functioning properly (Path: Network → Network Diagnostics

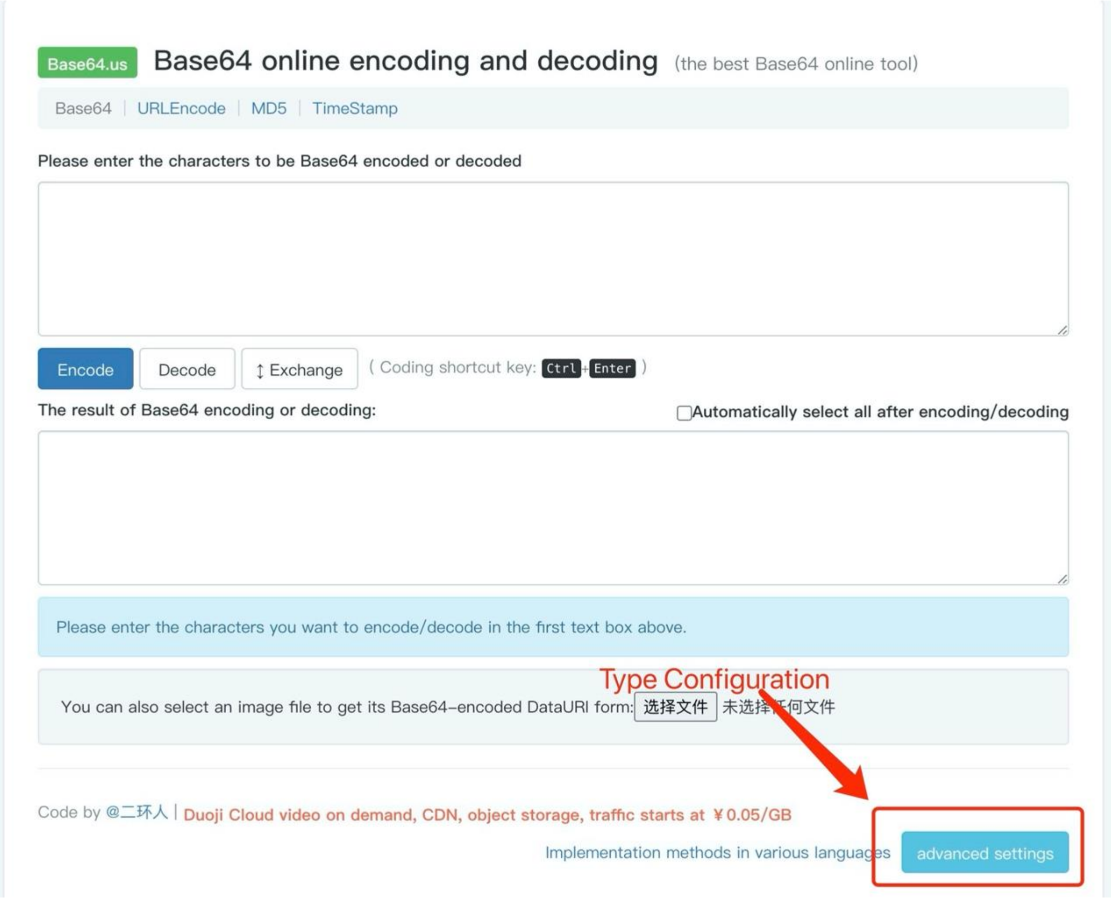
→ Ping).

- For MQTT type (Path: LoRa Network Server → Interface → Protocol Configuration):
  - Check if the MQTT switch is turned on.
  - Confirm the server's IP and port.
  - Verify the MQTT connection status.

- For TCP type:
  - Check if the corresponding TCP connection switch is turned on.
  - Verify the server's address and port.
  - Check the status of the corresponding connection.

4.1.6.5 Base64 encoding and decoding

- Online tool address: https://base64.us/



Base64.us Base64 online encoding and decoding (the best Base64 online tool)

Base64 | URLEncode | MD5 | TimeStamp

Please enter the characters to be Base64 encoded or decoded

Encode | Decode | ↕ Exchange | (Coding shortcut key: Ctrl + Enter)

The result of Base64 encoding or decoding: ☐Automatically select all after encoding/decoding

Please enter the characters you want to encode/decode in the first text box above.

You can also select an image file to get its Base64-encoded DataURI form: 选择文件 未选择任何文件

Type Configuration

Code by @二环人 | Duoji Cloud video on demand, CDN, object storage, traffic starts at ￥0.05/GB

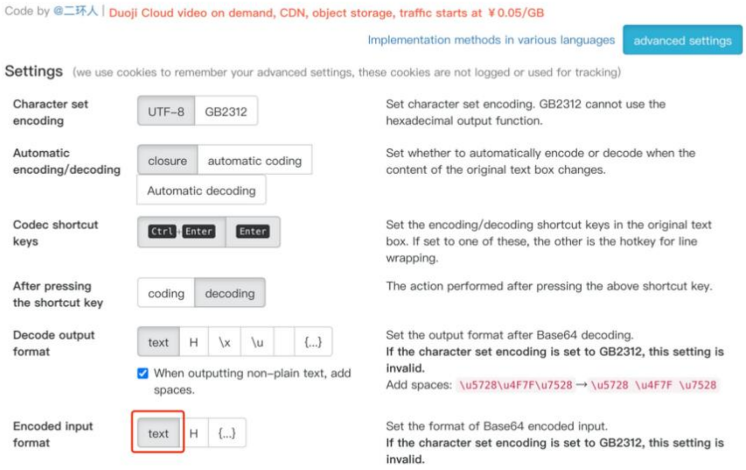
Implementation methods in various languages

advanced settings

- Mainly involves different data types with different encoding and decoding results, such as text (strings) or Hex (hexadecimal). The encoding and decoding configurations are in the advanced settings shown above.
- Encoding: (When sending downstream data, the data needs to be encoded in

base64 format)

- ➢ Text type (1234 → MTIzNA==)



Code by @二环人 | Duoji Cloud video on demand, CDN, object storage, traffic starts at ￥0.05/GB

Settings (we use cookies to remember your advanced settings, these cookies are not logged or used for tracking)

Character set encoding: UTF-8 | GB2312 — Set character set encoding. GB2312 cannot use the hexadecimal output function.

Automatic encoding/decoding: closure | automatic coding | Automatic decoding — Set whether to automatically encode or decode when the content of the original text box changes.

Codec shortcut keys: Ctrl+Enter | Enter — Set the encoding/decoding shortcut keys in the original text box. If set to one of these, the other is the hotkey for line wrapping.

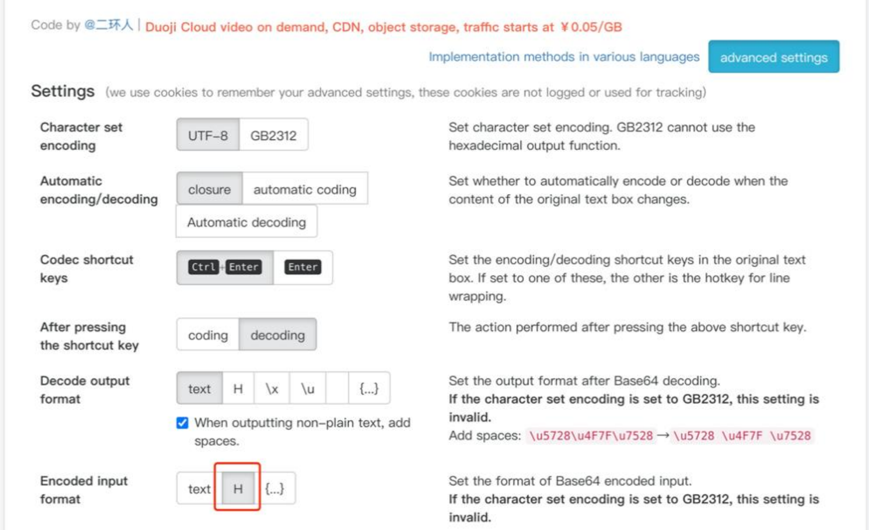
After pressing the shortcut key: coding | decoding — The action performed after pressing the above shortcut key.

Decode output format: text | H | \x | \u | {...} — Set the output format after Base64 decoding. **If the character set encoding is set to GB2312, this setting is invalid.** Add spaces: `\u5728\u4F7F\u7528` → `\u5728 \u4F7F \u7528`

☑ When outputting non-plain text, add spaces.

Encoded input format: [✓ text] | H | {...} — Set the format of Base64 encoded input. **If the character set encoding is set to GB2312, this setting is invalid.**

- ➢ Hex type (0x1234 → EjQ=)



Code by @二环人 | Duoji Cloud video on demand, CDN, object storage, traffic starts at ￥0.05/GB

Settings (we use cookies to remember your advanced settings, these cookies are not logged or used for tracking)

Character set encoding: UTF-8 | GB2312 — Set character set encoding. GB2312 cannot use the hexadecimal output function.

Automatic encoding/decoding: closure | automatic coding | Automatic decoding — Set whether to automatically encode or decode when the content of the original text box changes.

Codec shortcut keys: Ctrl+Enter | Enter — Set the encoding/decoding shortcut keys in the original text box. If set to one of these, the other is the hotkey for line wrapping.

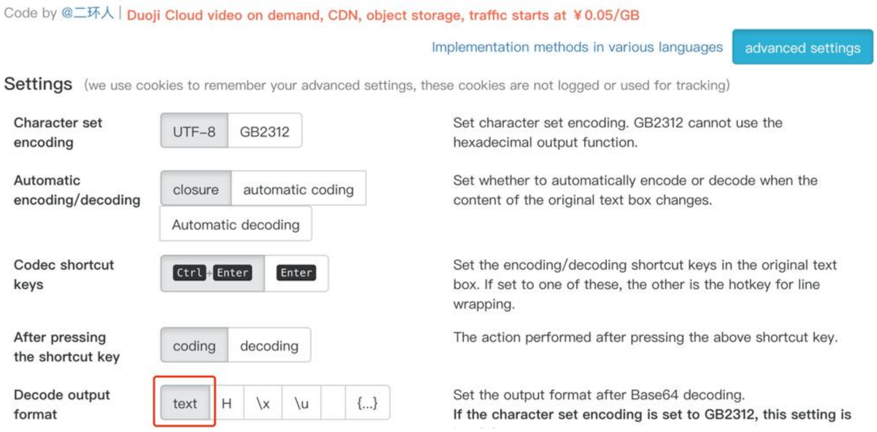
After pressing the shortcut key: coding | decoding — The action performed after pressing the above shortcut key.

Decode output format: text | H | \x | \u | {...} — Set the output format after Base64 decoding. **If the character set encoding is set to GB2312, this setting is invalid.** Add spaces: `\u5728\u4F7F\u7528` → `\u5728 \u4F7F \u7528`

☑ When outputting non-plain text, add spaces.

Encoded input format: text | [✓ H] | {...} — Set the format of Base64 encoded input. **If the character set encoding is set to GB2312, this setting is invalid.**

- Decoding: (The 'data' field content of the upstream push data needs to be decoded from base64 to actual content)
  - ➢ Text type (MTIzNA== → 1234)



Code by @二环人 | Duoji Cloud video on demand, CDN, object storage, traffic starts at ￥0.05/GB

Settings (we use cookies to remember your advanced settings, these cookies are not logged or used for tracking)

Character set encoding: UTF-8 | GB2312 — Set character set encoding. GB2312 cannot use the hexadecimal output function.

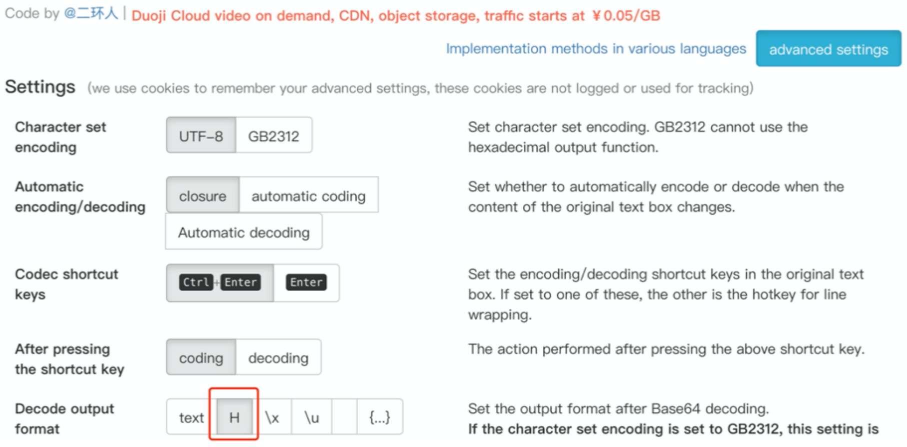
Automatic encoding/decoding: closure | automatic coding | Automatic decoding — Set whether to automatically encode or decode when the content of the original text box changes.

Codec shortcut keys: Ctrl+Enter | Enter — Set the encoding/decoding shortcut keys in the original text box. If set to one of these, the other is the hotkey for line wrapping.

After pressing the shortcut key: coding | decoding — The action performed after pressing the above shortcut key.

Decode output format: [✓ text] | H | \x | \u | {...} — Set the output format after Base64 decoding. **If the character set encoding is set to GB2312, this setting is**

- ➢ Hex tpye (EjQ= → 0x1234)



Code by @二环人 | Duoji Cloud video on demand, CDN, object storage, traffic starts at ￥0.05/GB

Implementation methods in various languages

advanced settings

Settings (we use cookies to remember your advanced settings, these cookies are not logged or used for tracking)

Character set encoding: UTF-8 | GB2312 — Set character set encoding. GB2312 cannot use the hexadecimal output function.

Automatic encoding/decoding: closure | automatic coding | Automatic decoding — Set whether to automatically encode or decode when the content of the original text box changes.

Codec shortcut keys: Ctrl+Enter | Enter — Set the encoding/decoding shortcut keys in the original text box. If set to one of these, the other is the hotkey for line wrapping.

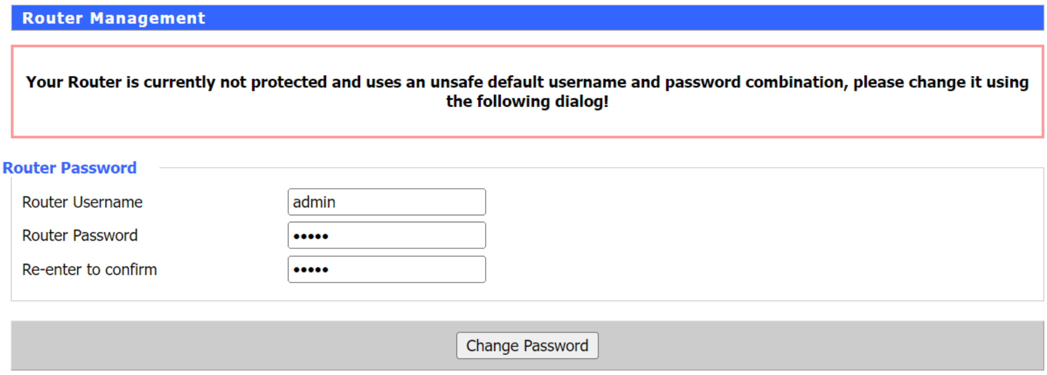
After pressing the shortcut key: coding | decoding — The action performed after pressing the above shortcut key.

Decode output format: text | H | \x | \u | | {...} — Set the output format after Base64 decoding. **If the character set encoding is set to GB2312, this setting is**

4.1.7 Management

4.1.7.1 Management

This page allows network administrators to manage specific functionalities of FBL800, ensuring access and security.



Router Management

Your Router is currently not protected and uses an unsafe default username and password combination, please change it using the following dialog!

Router Password

Router Username: admin

Router Password: •••••

Re-enter to confirm: •••••

The new password length must not exceed 32 characters and should not contain any spaces. The confirmation password should match the new password you set; otherwise, the configuration will not be successful.

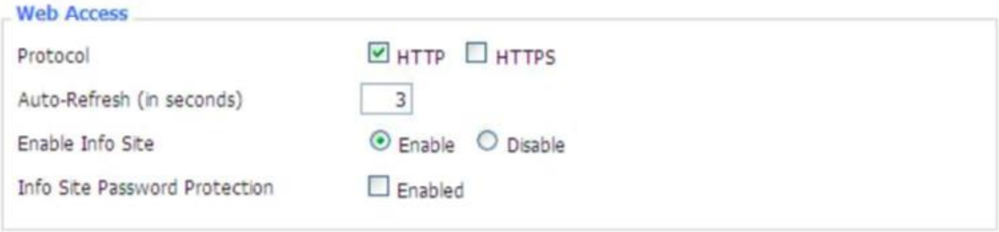
Warning

The default username is: admin.

We strongly recommend changing the factory default password, admin. This ensures that all users attempting to access and modify FBL800 must provide the correct password for access and usage.

Web Access

This feature allows you to manage FBL800 using either the HTTP or HTTPS protocol. If you choose to disable this feature, a manual restart will be required. You can also enable or disable the FBL800 information webpage, allowing password protection for this page (requiring the correct username and password input).



Web Access

Protocol ☑ HTTP □ HTTPS

Auto-Refresh (in seconds) 3

Enable Info Site ● Enable ○ Disable

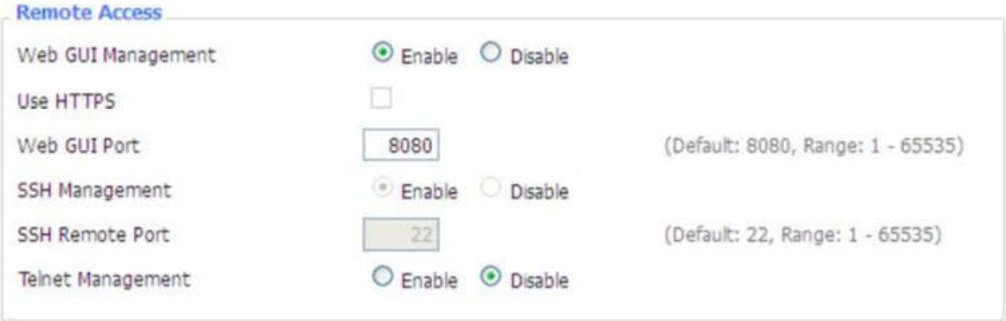
Info Site Password Protection □ Enabled

Protocol: The web page supports protocols including HTTP and HTTPS.

Auto Refresh (seconds): Adjust the time interval for the web interface to automatically refresh. 0 indicates that this feature is disabled.

Display system information webpage before login: Enable or disable displaying the system information webpage before login.

System information webpage password protection: Enable or disable the password protection feature for the system information webpage.



Remote Access

Web GUI Management ● Enable ○ Disable

Use HTTPS □

Web GUI Port 8080 (Default: 8080, Range: 1 - 65535)

SSH Management ● Enable ○ Disable

SSH Remote Port 22 (Default: 22, Range: 1 - 65535)

Telnet Management ○ Enable ● Disable

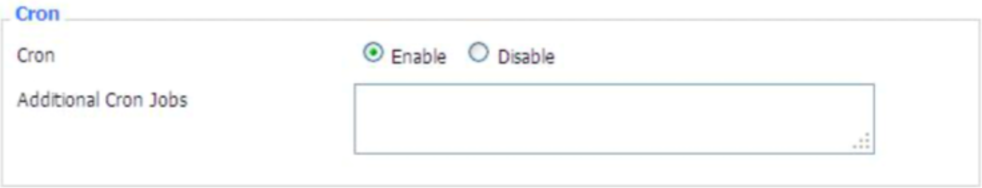
Web Interface Management: This feature allows you to manage FBL800 remotely over the Internet. To disable this feature, keep the default settings, which is disabled. To enable this feature, select enable and use the specified port on your computer (default is 8080) to remotely manage FBL800. If you haven't set a password yet, you must also set the default password for your FBL800. To remotely manage FBL800, go to http://xxx.xxx.xxx.xxx:8080 (replace 'x' with the Internet IP address of FBL800, and 8080 represents the specified port) in your web browser's address bar. You will be prompted to enter the password for FBL800. If you use HTTPS, you need to specify the URL as https://xxx.xxx.xxx.xxx:8080 (not all firmware supports SSL rebuilding).

SSH Management: You can enable SSH to remotely access FBL800 securely. Please note that to learn about the settings of the SSH daemon, you can access more information on the Services page.

Warning:

If the remote access feature of FBL800 is enabled, anyone who knows the Internet IP address and password of FBL800 will be able to change the settings of FBL800.

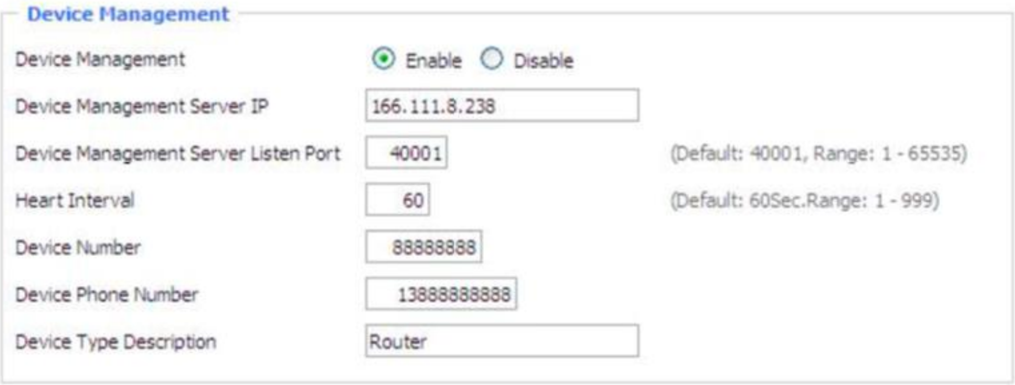
Telnet Management: Enable or disable remote Telnet functionality.



Cron

Cron	● Enable ○ Disable
Additional Cron Jobs	

Cron: The cron subsystem is for scheduling Linux commands that you plan to execute. In practice, you may need to use the command line or startup scripts.

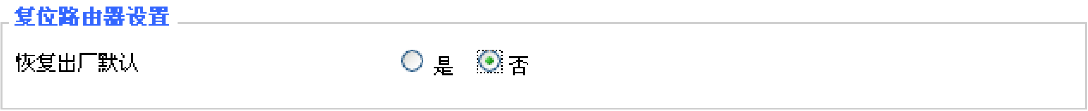


Device Management

Device Management	● Enable ○ Disable	
Device Management Server IP	166.111.8.238	
Device Management Server Listen Port	40001	(Default: 40001, Range: 1 - 65535)
Heart Interval	60	(Default: 60Sec.Range: 1 - 999)
Device Number	88888888	
Device Phone Number	13888888888	
Device Type Description	Router	

Device Management: Monitor and manage this FBL800 unit through a custom-developed remote management server, including parameter configuration, Wi-Fi advertising updates, and more.

4.1.7.2 Factory Default

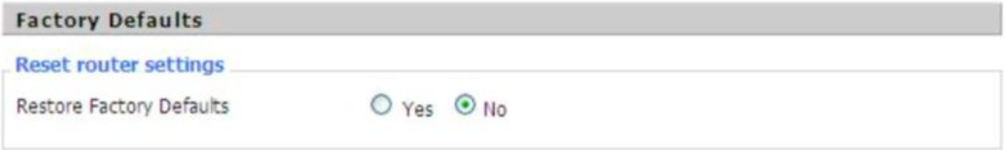


复位路由器设置

恢复出厂默认	○ 是 ● 否

Restore Factory Defaults: Clicking the "**Yes**" button and saving the settings will clear all configurations and restore them to the factory values. When restoring to default settings, all the changes you made will be lost. The default configuration for this function is set to "**No**". For more information, please click "**More**".

4.1.7.3 Firmware Upgrade



Factory Defaults

Reset router settings

Restore Factory Defaults ○ Yes ● No

Firmware Upgrade: This feature allows you to load new firmware onto the FBL800. New firmware versions will be released on www.four-faith.com and can be downloaded for free. If the FBL800 is functioning properly, there is no need to download the updated firmware version unless it includes new features you want to use.

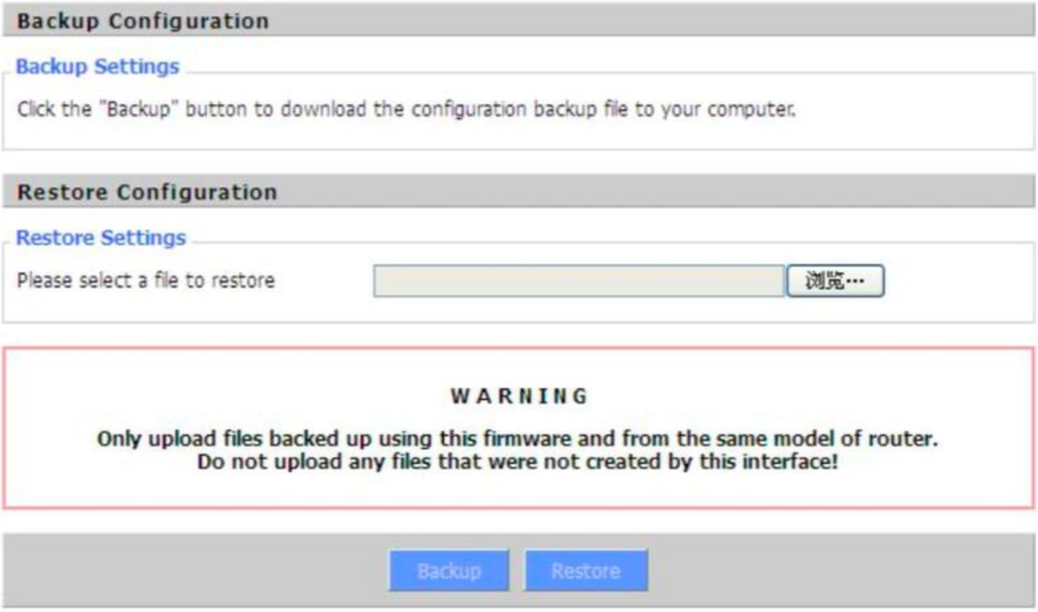
Note: When upgrading the firmware of the FBL800, configuration settings may be lost, so be sure to backup the settings of the FBL800 before upgrading the firmware.

After the upgrade, reset to: If you want to reset the firmware version of the FBL800 to default settings after the upgrade, click the "Default Settings" option.

Click "Browse," select the firmware file to be upgraded, and then click the "Upgrade" button to start the firmware upgrade. The firmware upgrade may take a few minutes, so do not power off or press the reset button during the process.

4.1.7.4 Backup

This page is used to backup or restore the configuration file of the FBL800.



Backup Configuration

Backup Settings

Click the "Backup" button to download the configuration backup file to your computer.

Restore Configuration

Restore Settings

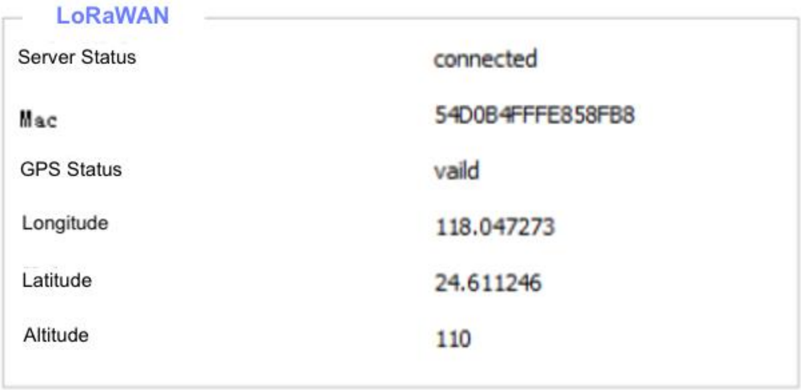
Please select a file to restore

WARNING

Only upload files backed up using this firmware and from the same model of router.
Do not upload any files that were not created by this interface!

If you want to backup the configuration file of the FBL800, click the "**Backup**" button. Follow the on-screen instructions.
If you want to restore the configuration file of the FBL800, click the "**Browse**" button, locate the backup file, and follow the on-screen instructions. After selecting the backup file, click the "**Restore**" button.

4.1.7.5 FBL800



LoRaWAN

Server Status	connected
Mac	54D0B4FFFE858FB8
GPS Status	vaild
Longitude	118.047273
Latitude	24.611246
Altitude	110

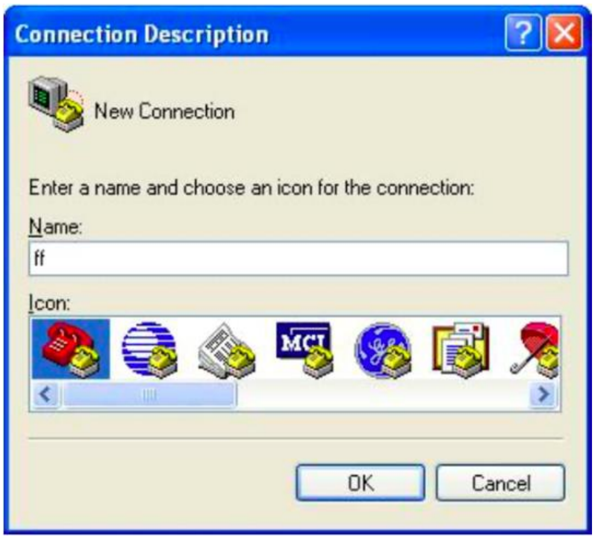
Server Status: The connection status with the specified LoRaWAN server.
Mac: The MAC address of the FBL800, serving as an identification code for different FBL800 devices recognized by the LoRaWAN server.
GPS Status: Indicates whether there is GPS signal.
Longitude, Latitude, Altitude: Information obtained from GPS.

Appendix

Capturing debug information through Console using HyperTerminal: Step-by-step guide and configuration methods (WINDOWS XP)

1. Click on "Start" -> "Programs" -> "Accessories" -> "Communications" -> "HyperTerminal" (or directly click "Start" -> "Run" and type "hypertrm" to launch HyperTerminal).

The interface after launching HyperTerminal is as follows:



Connection Description

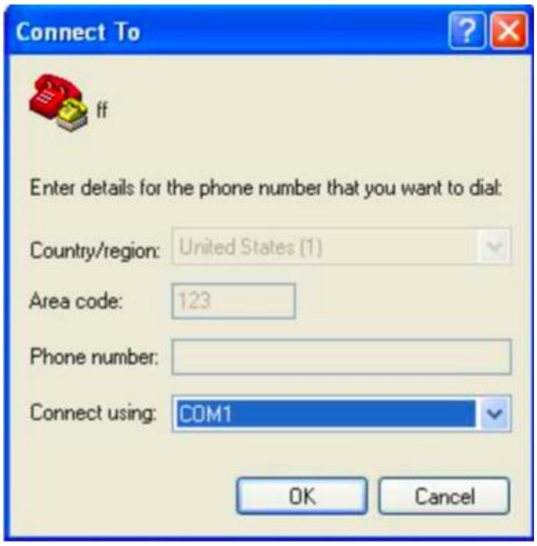
New Connection

Enter a name and choose an icon for the connection:

Name: ff

2. Enter the connection name and select "OK."

3. Select the PC's physical serial port used to connect to the FBL800 Console port and choose “Confirm."



Connect To

ff

Enter details for the phone number that you want to dial:

Country/region: United States (1)

Area code: 123

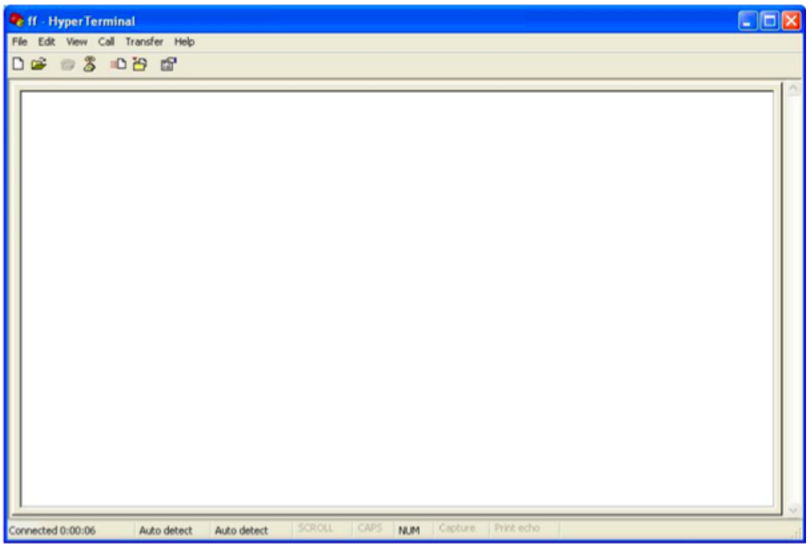
Phone number:

Connect using: COM1

4. Select the PC's physical serial port used to connect to the FBL800 Console port and choose “Confirm."

   Baud rate: 115200
   Data bits: 8
   Parity: None
   Stop bits: 1
   Flow control: None

At this point, HyperTerminal is running normally.



If the user is using the Windows 7 system, they can download a HyperTerminal for Windows 7 online. Alternatively, they can use other commonly used serial interaction software with similar usage.