

255MG-701 数据手册

内置 NS 工业级二开 LORAWAN 网关

版本 1.1

日期 2026-01-1



版权声明

南京市二五五物联科技有限公司保留所有权利。



是南京市二五五物联科技有限公司所有商标。

本指南中出现的其他商标，由商标所有者所有。

说明

本应用指南对应产品为 255MG-701:

本应用指南的使用对象为系统工程师，开发工程师及测试工程师由于产品版本升级或其它原因，本手册内容会在不预先通知的情况下进行必要的更新除非另有约定，本手册中的所有陈述、信息和建议不构成任何明示或暗示的担保。

南京市二五五物联科技有限公司为客户提供全方位的技术支持，任何垂询请直接联系您的客户经理或发送邮件至以下邮箱：

联系方式：15651028736

邮箱：yuyang@255mesh.com

公司网址：<http://www.255mesh.com>

修改历史

版本	日期	原因
V1.0	2026/01/04	创建文档

目录

目录	3
1. 产品简介	5
1.1 产品特征	6
1.2 订购信息	6
1.3 功能参数	7
2. WEB 配置	8
2.1 访问设备内置 WEB	0
2.2 系统信息	1
2.2 参数配置	1
2.2.1 网络配置	1
2.2.2 Gateway 设置	3
2.2.3 LoRaWAN NS 配置	4
2.2.4 Node-RED	6
2.2.5 设备维护页	7
2.2.6 串口连接	8
3. 功能介绍	9
3.1 多网卡管理	9
3.2 LED 指示灯状态	9
4. 测试用 NS	10
5. NS 对接	0
5.1 对接 ChirpStack	0
5.1 对接 TTN	0
6. 二次开发	0
7. ChirpStack 的使用	0
7.1 Chirpstack 简介	0
7.2 网关录入	0
南京二五五物联科技有限公司版权所有	3

- 7.3 设备录入 1
 - 7.3.1 创建 Device Profiles 2
 - 7.3.2 创建 Application 3
 - 7.3.3 录入设备 3
- 7.4 查看数据 4
 - 7.4.1 查看网关数据 4
 - 7.4.2 查看节点数据 5
- 7.5 数据下行 6

1. 产品简介

255MG-701 是南京二五五物联开发的一款智能数据采集网关，采用的 LoRaWAN/4G/WIFI/485 传输，实现了数据传输功能，方便实现远程、无线、网络化的通信方式，轻松实现与平台互联，大幅度的缩减了无线产品复杂的开发过程，使您的产品以更低的成本快速投入市场，可广泛应用于工业物联网应用中。

该系列产品提供 2 路以太网接口，8 路 LoRaWAN 无线接收通道，1 路发送通道，4G 无线互联平台，WIFI 配置、联网，另外预留了一路 RS485 接口，可方便扩展功能。

具有网络覆盖范围广（4G）、组网灵活快捷（安装即可使用）、运行成本低（按流量或时长计费）等诸多优点。可应用于电力系统、工业监控、交通管理、气象、水处理、环境监控、煤矿和石油等行业。



1.1 产品特征

- ◆ 高性能嵌入式硬件平台
- ◆ MiniPCIE 4G 模块，可更换全球各国频段
- ◆ 宽压输入 DC 9~28V，工业级稳定性
- ◆ 群脉冲：电源 $\pm 2kV$ ，通讯线 $\pm 4kV$
- ◆ 湿度范围：10%~95%，功能丰富
- ◆ 内置网页，方便查看、配置系统功能
- ◆ 支持 UDP Packet Forward 协议，Chirpstack MQTT Forwarder
- ◆ 内置 Chirpstack，单台设备即可完成 LoRaWAN 网络搭建
- ◆ 可通过 LED 查看各数据通道状态，如 无线连接、数据收发等
- ◆ WIFI/4G/以太网多网络备份，自动切换，数据链路充分冗余
- ◆ RS485 接口，丰富的可扩展性

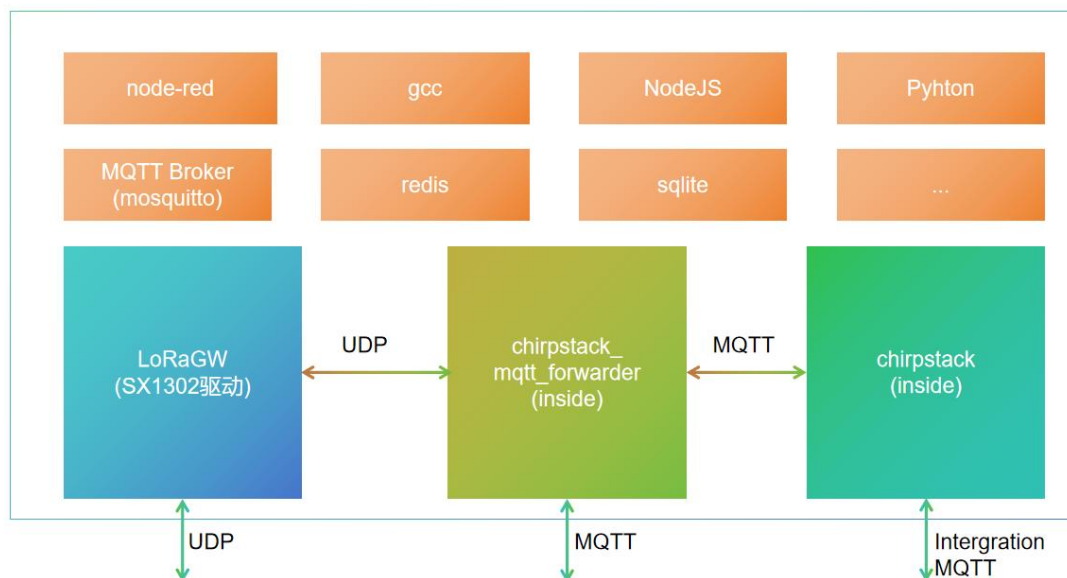
1.2 订购信息

型号	场景	4G 网络
255MG-701ETH	室内	无
255MG-701LTE	室内	有

1.3 功能参数

分类	参数	数值
硬件参数	工作电压	DC 9~28
	待机功耗	1.5W(12V)
物理特性	外壳	钣金
	尺寸	164.0*91.9*32.0(mm) (L*W*H 不含端子)
	工作温度	-20°C~70°C
	储存温度	-45°C~90°C
	工作湿度	10%~90% (无凝露)
通讯接口	联网方式	4G MiniPCle、Ethernet、WIFI(2.4G)
	LoRa 频段	433/470/868/915/923 全球可选
	扩展接口	全隔离 RS485
功能	NS 对接	UDP Packet Forward/ Chirpstack MQTT Forwarder
	配置	WEB
	升级	WEB 升级
	多网卡自动切换	4G、Ethernet、WIFI 自动切换
	支持频段	EU433/CN470/EU868/US915/AU915/AS923/IN865

1.4 设备整体框架



设备内置 ubuntu core 系统，有 apt 包管理器，除上述组件外，可自行安装各种组件。

LoRaGW、chirpstack_mqtt_forwarder、chirpstack 三个为 lorawan 相关的主服务。

1. LoRaGW 是 lorawan 的射频接口，负责频点配置、接收、发送数据。
2. chirpstack_mqtt_forwarder 负责将 loragw 的 UDP 数据转成 MQTT 数据发给 chirpstack。
3. chirpstack 是 LoRaWAN 的处理核心，LoRaWAN 相关的加解密、入网等各种机制都在这实现，intergration MQTT 是 chirpstack 的对外接口，可以从这边上下行数据。

以上各层也可以选择对接到外置的 NS 去，比如用 LoRaGW 或者 chirpstack_mqtt_forwarder 去对接 NS。

2. 设备配置

2.1 访问设备 WEB

1. 设备支持通过以太网、WIFI 访问设备

其中

- 以太网的 WAN 口的 IP 是不固定的，由上级路由器定，查看上级路由器获取
- 以太网的 LAN 口和 WIFI AP 的网络是同一个网段，默认是 192.168.43.1

WIFI 的 SSID 和密码：

- SSID 默认是 LoRaGW-XXXXXX,其中 XXXXXX 为 SN 的后 6 位
- 密码默认为 loragwwifi

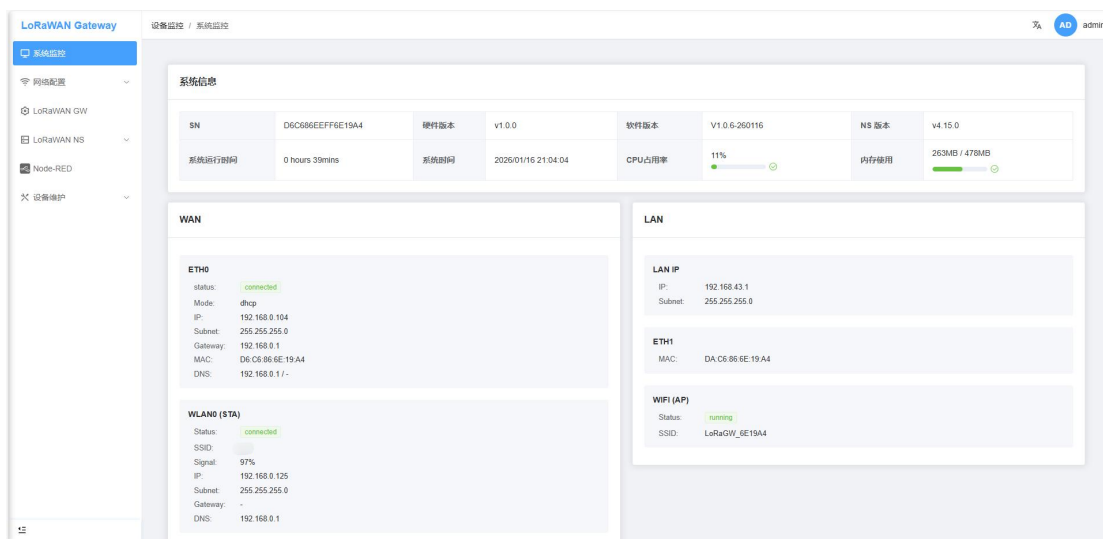
WEB 端的默认的用户名密码为

- 用户名:admin
- 密码:admin123

所以通常访问设备 WEB 的方式是：

- 1.连接设备的 LAN 口以太网或者 WIFI AP
- 2.在浏览器中输入 <http://192.168.168.1>
- 3.输入用户名密码来访问 WEB 页面

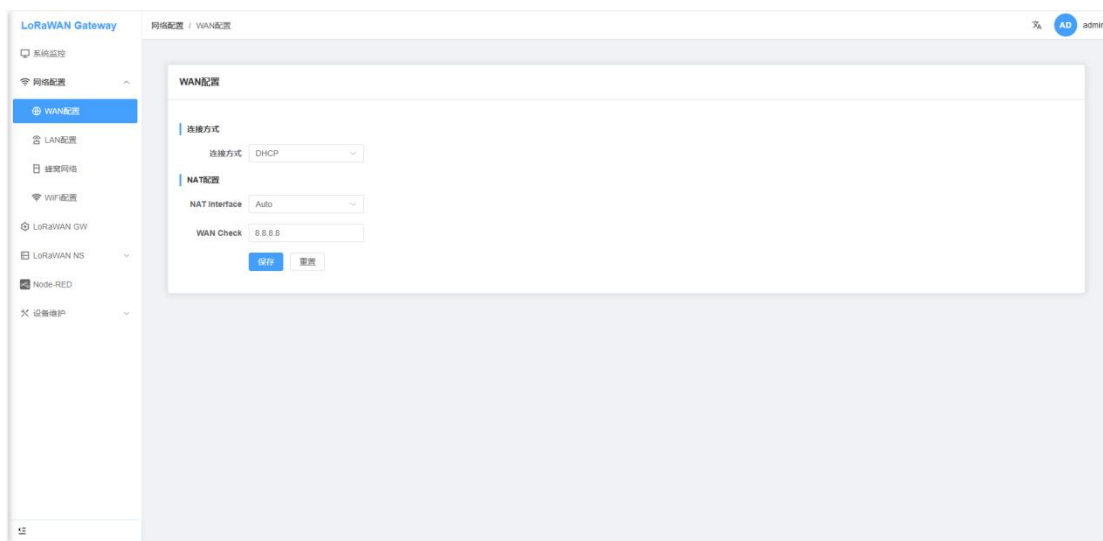
2.2 系统信息



2.2 参数配置

2.2.1 网络配置

2.2.1.1 WAN 配置



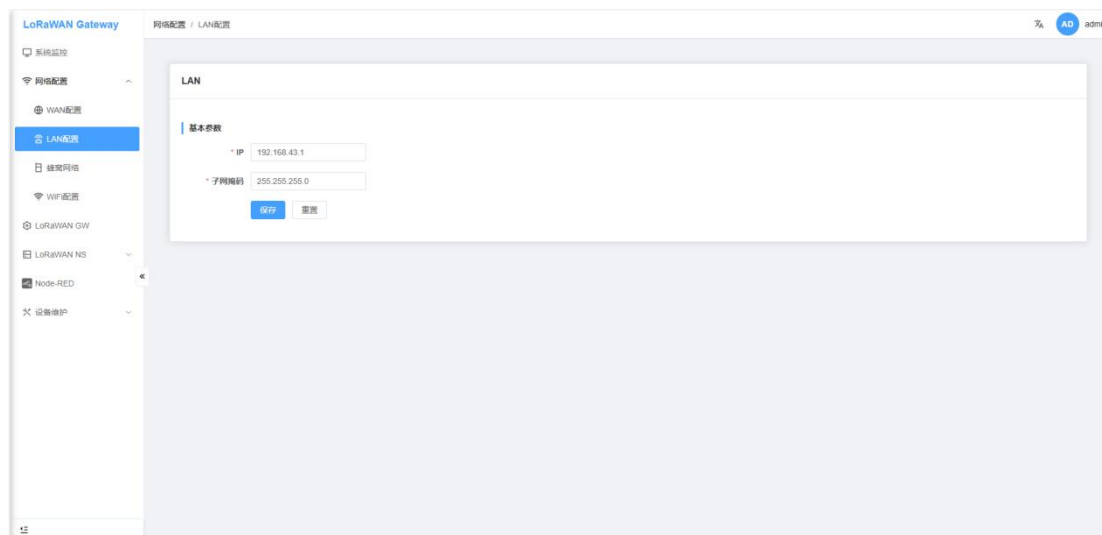
可以设置 WAN ETH 的连接方式

支持 DHCP 和静态 IP

其中 NAT 配置主要是配置设备访问外网的接口和外网监测的 IP。

从网络接口来看，WAN 口，WIFI，蜂窝都可以用来访问外网，所以用户可以选择固定走哪个接口，或者自动选择，自动选择的优先级是 WAN 口> WIFI> 蜂窝。另外，为了测试外网的连通性以自动选择合适的接口来访问外网，设置了 **WAN Check 地址**来检测网卡对于外网的连通性。

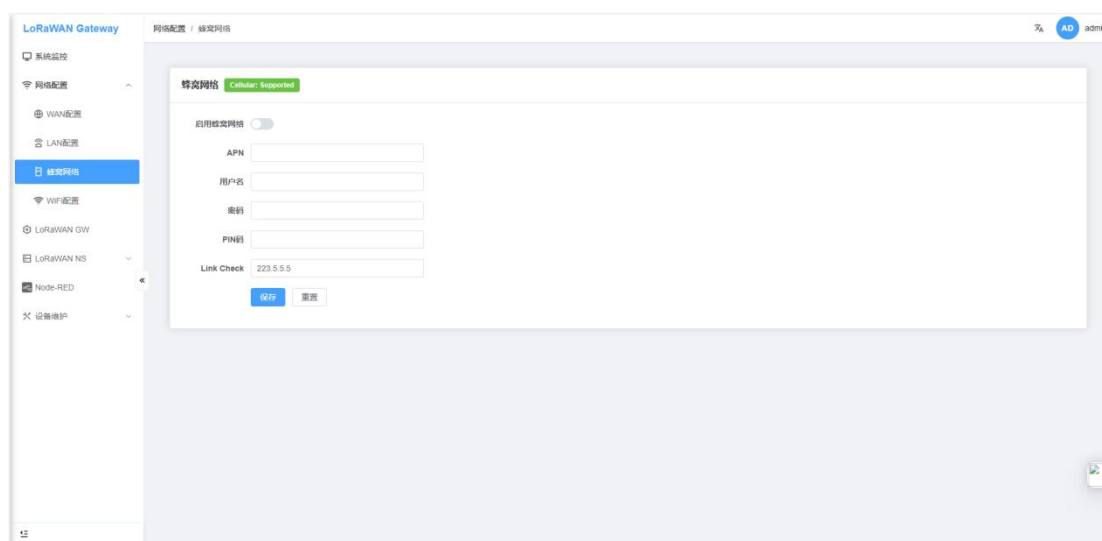
2.2.1.2 LAN 配置



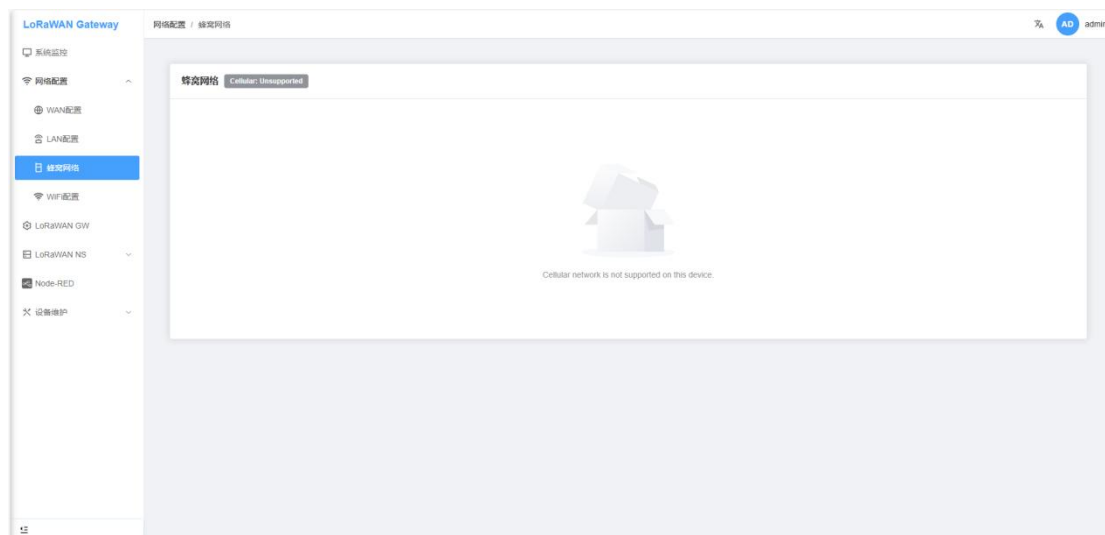
支持配置 LAN 的网段，此设置会应用于所有 LAN 口的网卡，包括 LAN ETH 和 WIFI AP

LAN 默认是 192.168.168.1 地址，也就是是 192.168.168.0 这个网段的

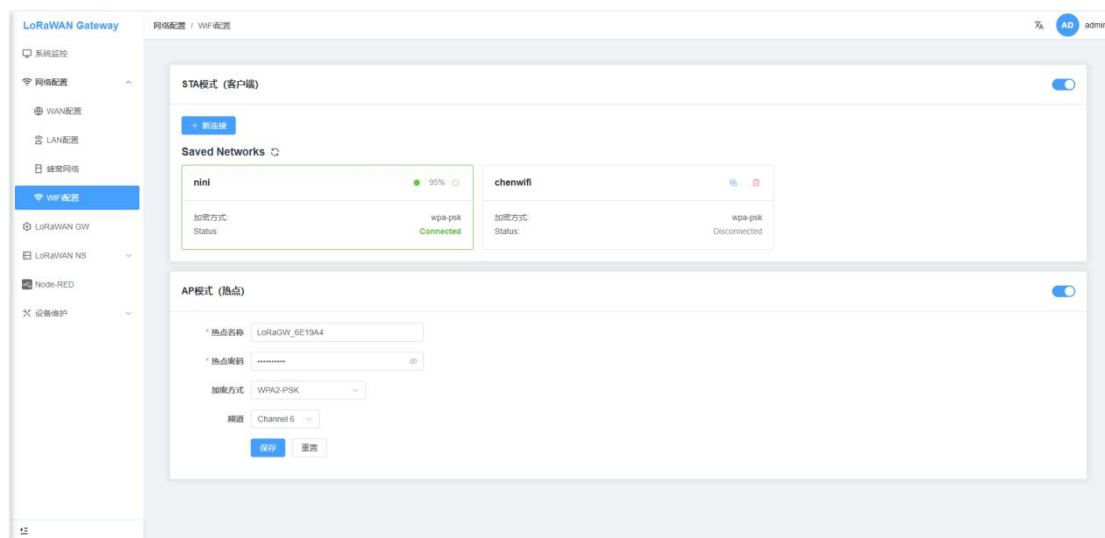
2.2.1.3 蜂窝配置



如果不支持蜂窝会显示



2.2.1.4 WIFI 配置



设备支持 STA+AP 同时开启

2.2.2 Gateway 设置

2.2.2.1 LoRaGW

LoRaWAN Gateway

LoRaWAN GW

Gateway ID: D6C88EEFF6E19A4

转发方式: [+NS](#) UDP Packet Forwarder

射频设置

Auto (Region) Manual

Region Plan: US915_0

Radio Units

Center Radio Frequency

Enable	Name	Center Frequency (Hz)
☒	Radio 0	902700000
☒	Radio 1	902900000

Multi-Channel(0-7)

Enable	Index	Radio Chain	IF (Hz)	Frequency (MHz)
☒	0	Radio 0	-40000	902.3
☒	1	Radio 0	-20000	902.5
☒	2	Radio 0	0	902.7
☒	3	Radio 0	20000	902.9

- 此页面主要设置网关的 ID 和射频相关参数，另外网关的指向可以指向本地 NS 或者第三方 NS
- 其中射频配置部分，已经载入了所有支持的区域的频点模版，用户只需要配置修改 Region Plan 可以自动更改合适的频点。当然用户也可以选择手动模式，手动修改频点。

2.2.2.2 MQTT Forward

LoRaWAN Gateway

LoRaWAN NS / MQTT Forward

ChirpStack MQTT Forward Configuration

MQTT Server: tcp://127.0.0.1:1883

Client ID: 12345

Username: 23456

Password: ****

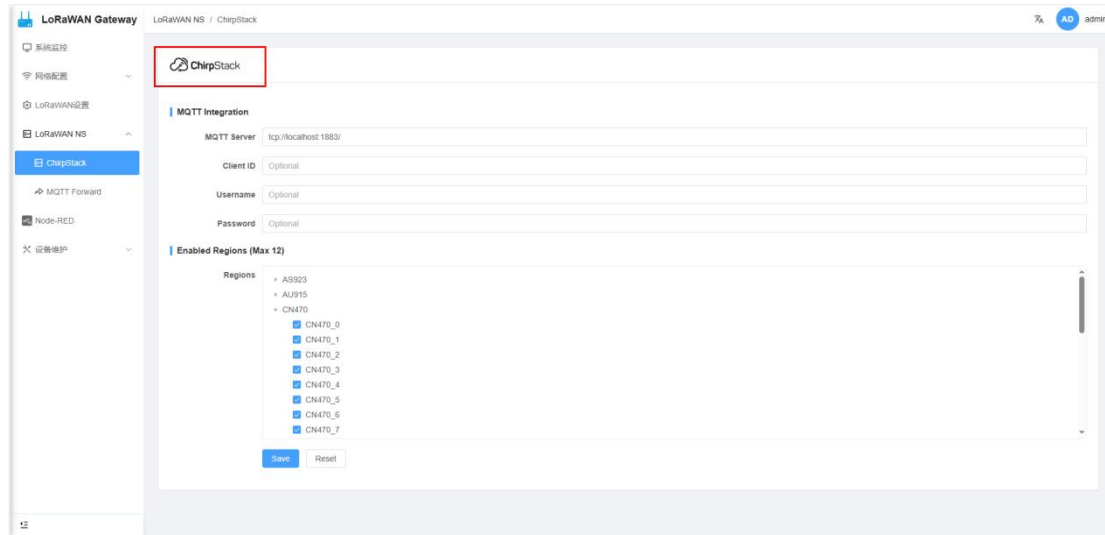
[Save](#) [Reset](#)

支持配置 Chirpstack MQTT Forwarder，注意此功能要搭配 Chirpstack 使用，此功能是将原先 UDP 的 Packet Forward 转换为 MQTT 协议，减少因为 UDP 的不稳定导致丢包的概率，注意需要保留前面的 tcp://前缀。

2.2.3 LoRaWAN NS 配置

设备使用的是 NS 是 Chirpstack，是开源的 NS，使用上与 WEB 端部署的 chirpstack 是相同的。

2.2.3.1 登录 Chirpstack



点击 logo 跳转到 Chirpstack

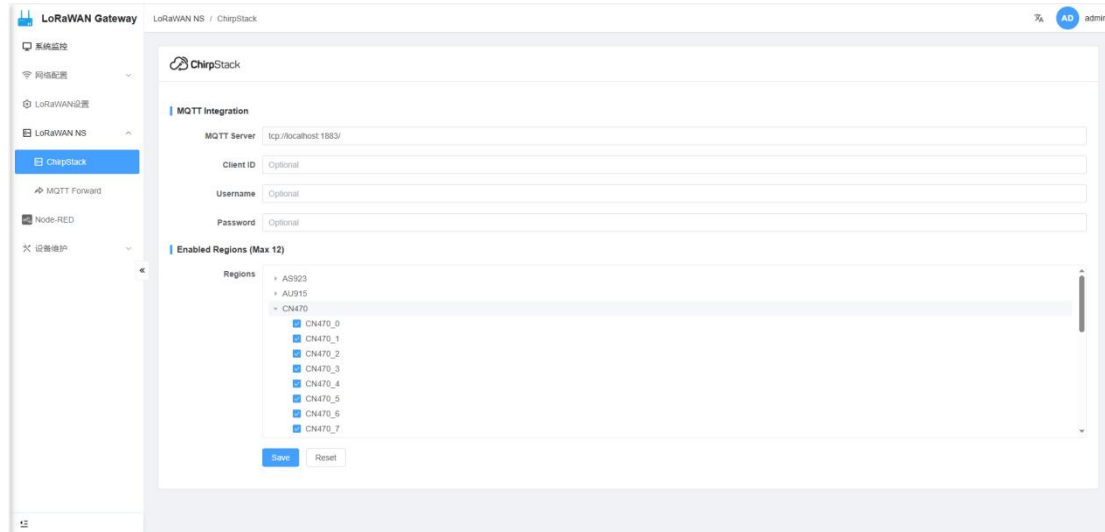
chirpstack 默认的用户名密码:

用户名: admin

密码: admin

建议用户在登陆之后修改用户名密码以增强安全性。

2.2.3.2 Chirpstack 配置



- 支持 MQTT Intergration, 此功能可以将 LoRaWAN 数据解析之后转发到指定 MQTT, 注意需要保留前面的 tcp://前缀。
- 支持配置多个 Region, 即除了设备本信道, 其他信道的网关也可以接入
- 支持其他网关也使用本设备内部的 NS

2.2.3.3 快速添加设备

此功能需软件版本 V1.1.19 以上支持

此功能需要先在 ChirpStack 中创建 Application 和 Device Profile。

支持录入单个设备或者通过表格批量录入设备

批量添加设备时先下载模版，再将设备参数添加到表格中，最后再将表格导入之后开始添加。

	A	B	C	D	E	F	G	H	I
1	DevEui	JoinType	AppKey	AppKey	NwkKey	DevAddr	Name	Description	
2	0011223344-OTAA		00112233445566778899AABBCCDDEEFF				otaa-device		
3	8899AABBCC-ABP			00112233445566778899AABBCCDDEEFF	00112233445566778899AABBCC	01020304	abp-device		
4									
5									
6									
7									

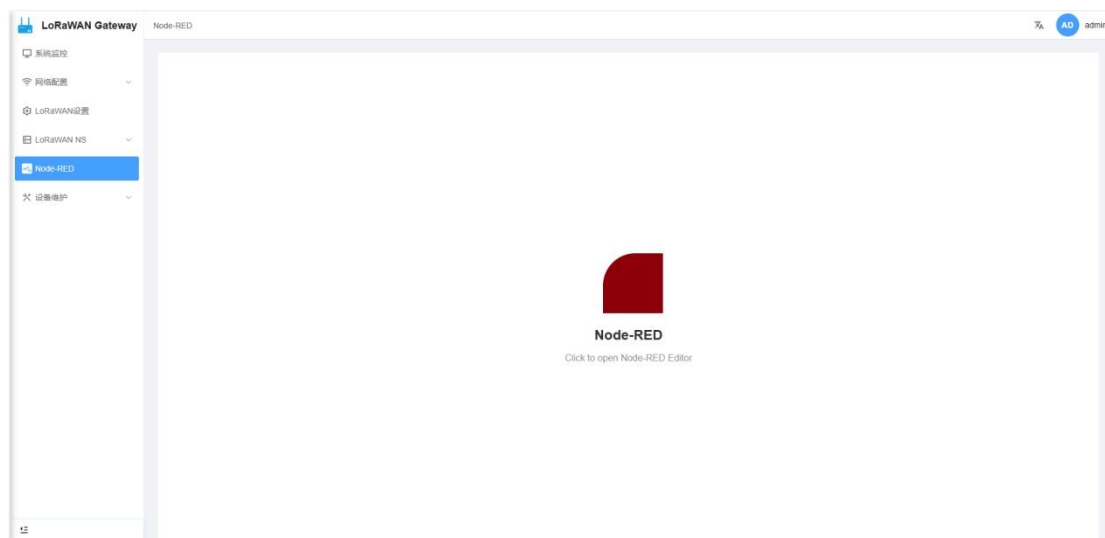
批量添加的表格如上，不选的参数可以空着

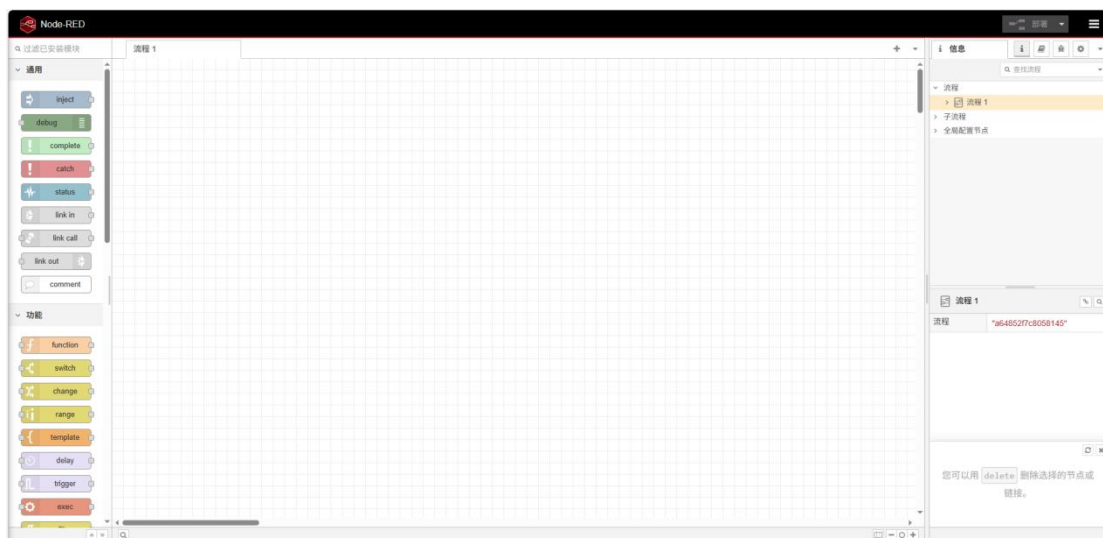
例如：

- OTAA 入网时不需要填写 DevAddr, NwkSKey, AppSKey
- ABP 入网时不需要填写 AppKey

表格的第一行为表头，不能删除，第二三行分别为 otaa 和 abp 参数的示例，录入时要删掉，并添加相关的设备参数。

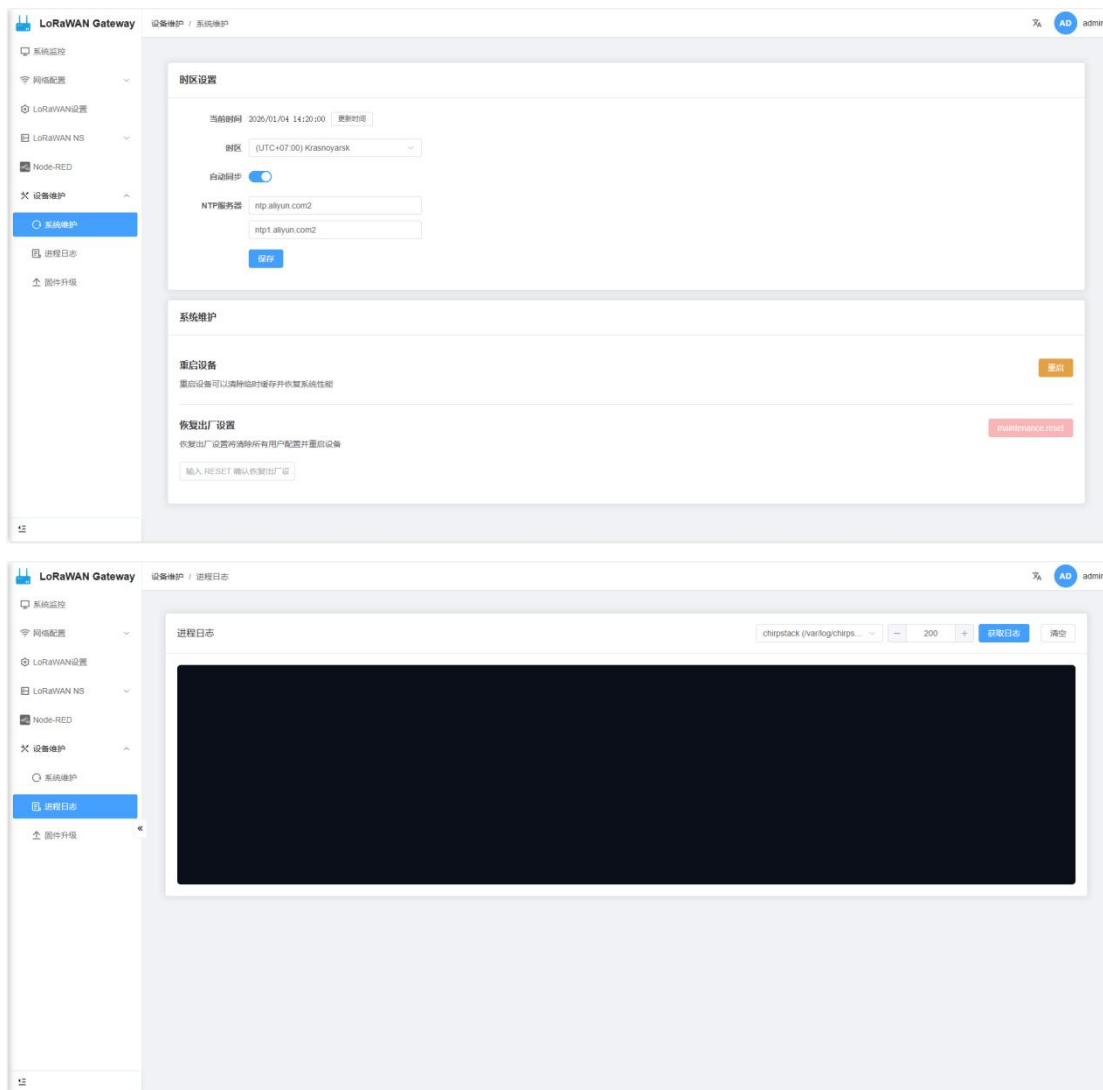
2.2.4 Node-RED

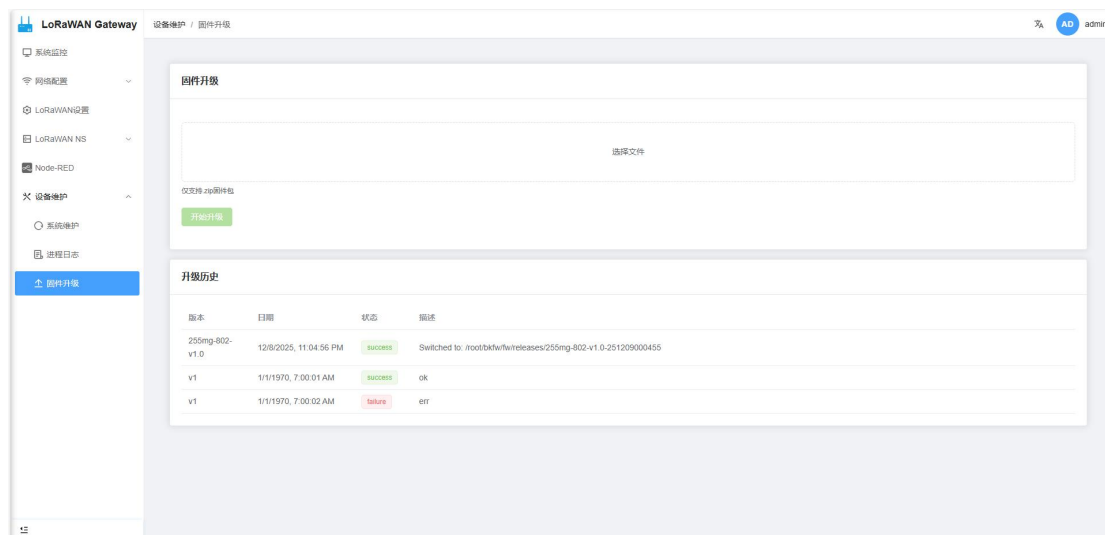




Node-RED 是个支持图形化开发的工业级二次开发工具。支持各种插件安装来扩展各种功能。

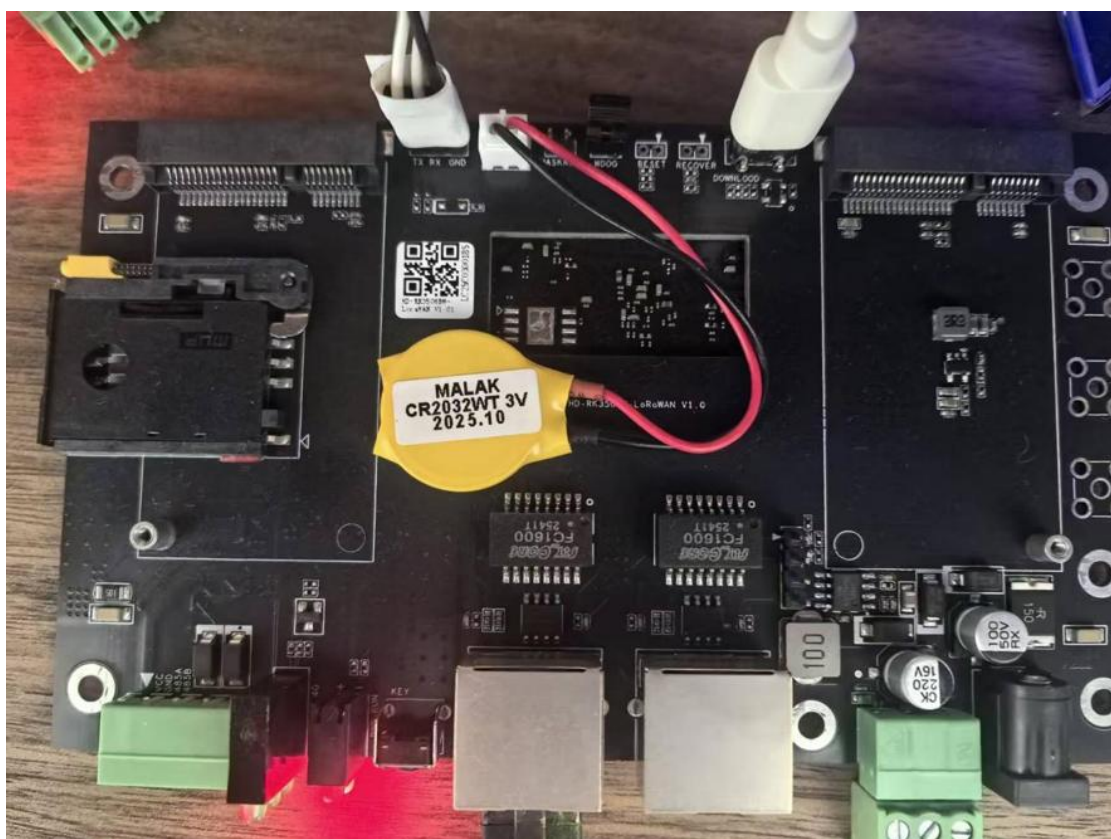
2.2.5 设备维护页





2.2.6 串口连接

如下图所示，连接 TTL 接口，板上丝印和排针顺序是一一对应的，注意 TX 和 RX 交叉连接，即 TX->RX, RX->TX，若板上未焊接排针，需手动焊接排针。



3. 功能介绍

3.1 多网卡管理

设备至多支持 3 种上网方式：

- WIFI STA
- Ethernet
- 4G

这三种上网方式的选择是遵循网卡的就绪状态和优先级进行的。

- 就绪状态：当此种方式能获取到 IP，即认为此网卡就绪；
- 优先级：Ethernet>WIFI STA>4G；
- 网卡选择逻辑：设备总是会选择就绪网卡中，优先级最高的那个进行通讯。

3.2 LED 指示灯状态

指示灯详细描述如下：

- RF-LED：LoRaWAN 模块处于接收模式，接收时常亮，发送时常灭。
- RUN-LED：
 - ◆ 系统已启动，LoRaWAN 线程未启动，亮 0.5 秒，灭 0.5 秒。
 - ◆ 系统已启动，LoRaWAN 线程已启动，常亮。
- WIFI-LED：
 - ◆ WIFI STA 未启用，常灭
 - ◆ WIFI STA 搜网中，亮 0.5 秒，灭 0.5 秒。
 - ◆ WIFI STA 已连接，常亮。
- 4G-LED：亮 0.2 秒，灭 1.8 秒；处于搜网状态。
 - ◆ 亮 1.8 秒，灭 0.2 秒，处于待机。
 - ◆ 亮 0.125 秒，灭 0.125 秒，处于数据传输状态

3.3 按键重启和恢复出厂设置

1. 按键超过 1s，设备重启
2. 按照超过 5s，恢复出厂设置

4.测试用 NS

我也提供了测试用 NS，域名：lora.ansitw.com，IP：106.52.124.231

WEB 端地址：106.52.124.231:8080

包转发地址：

- CN470_0:106.52.124.231:1700
- CN470_10:106.52.124.231:1704
- EU868:106.52.124.231:1701
- US915_0:106.52.124.231:1702
- US915_1:106.52.124.231:1703
- AU915:106.52.124.231:1705
- AS923:106.52.124.231:1706
- IN865:106.52.124.231:1707

为了方便测试，我司提供了一个测试账号，可用于临时测试使用：

WEB 端地址：106.52.124.231:8080

测试账号：test@loragw.com

测试账号密码：test123456

Note:

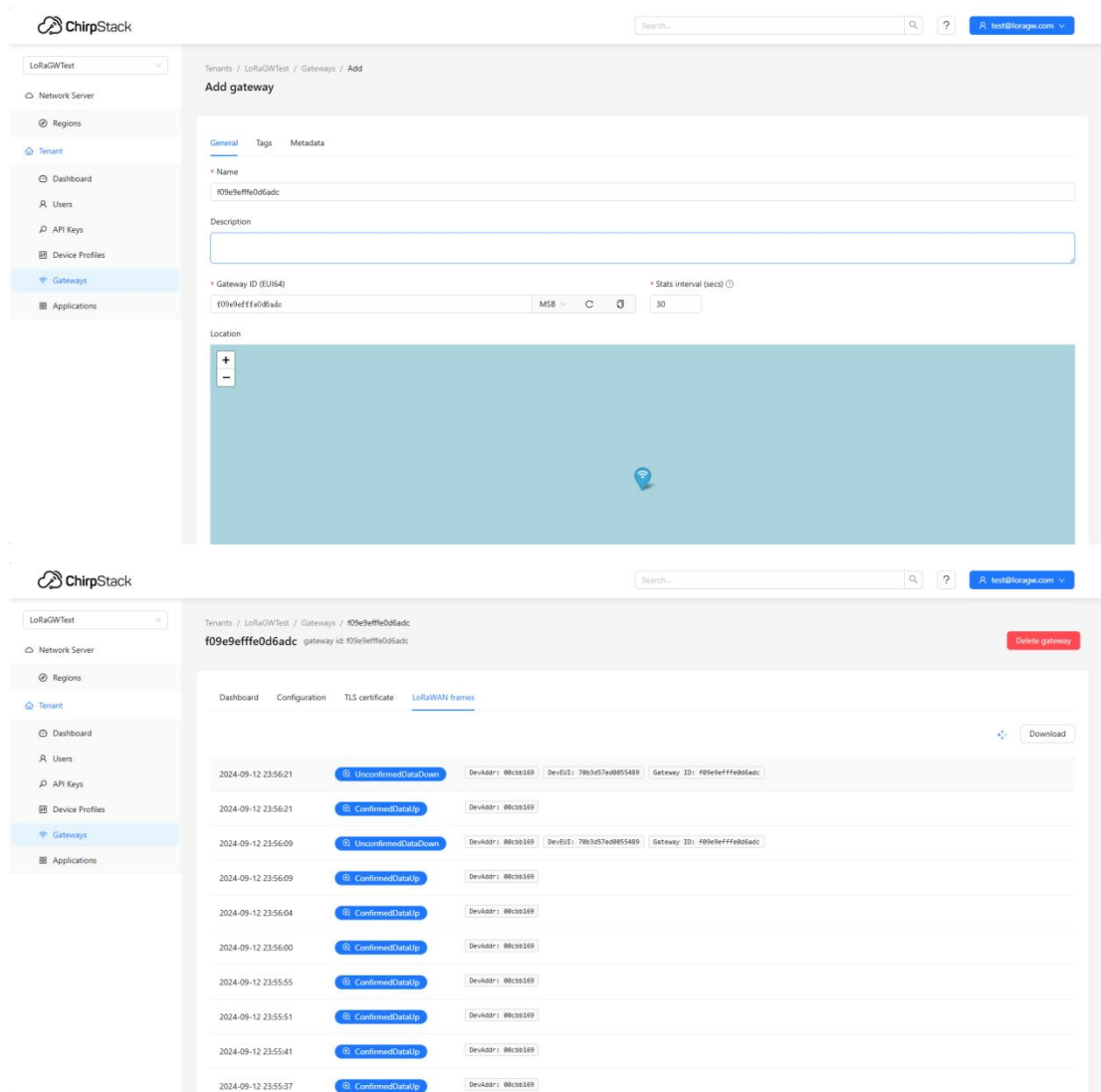
- 1.测试账号仅供测试使用
- 2.因为测试账号是公开的，可能会有多人在使用，所以使用时请不要去删除自己设备以外的数据
- 3.同上，请不要在测试 NS 上跑重要的业务数据，因为这是不安全

5. NS 对接

网关使用 UDP Packet-forward 协议对接 NS。这个协议由 semtech 设计, 广泛应用与 LoRaWAN Gateway 与 NS 之间。

5.1 对接 ChirpStack

如下图录入即可



5.2 对接 TTN

网关信息录入

Register gateway

Register your gateway to enable data traffic between nearby end devices and the network.
Learn more in our guide on [Adding Gateways](#).

Gateway EUI

[Reset](#)

Gateway ID

Gateway name

Frequency plan

[+ Add frequency plan](#)

Note: most gateways use a single frequency plan. Some 16 and 64 channel gateways however allow setting multiple within the same band.

☐ **Require authenticated connection**

Choose this option eg. if your gateway is powered by [LoRa Basic Station](#)

Share gateway information

Select which information can be seen by other network participants, including [Packet Broker](#)

☒ **Share status within network**

☒ **Share location within network**

[Register gateway](#)

网关已上线，可看到数据已有上行

f09e9efffe0d0ade
EUI: f09e9efffe0d0ade

Last activity 30 seconds ago • 14:40 up / 0 down

General information

Gateway ID:

Gateway EUI:

Frequency plan: China 470-510 MHz, FSB 1

Created at: Sep 13, 2024 00:30:02

Network settings

Automatic updates : Disabled

Require authenticated connection : Disabled

Public status : Enabled

Public location : Enabled

Packet Broker forwarding : Enabled

Status location updates : Disabled

Enforce duty cycle : Enabled

Gateway status

30 day uptime

Unlock uptime graph

[Upgrade now](#)

Roundtrip times (ms)

No data available

This gateway doesn't have recent downlinks and cannot display the roundtrip time.

Connection status

↑ 40 34 sec ago

↓ No downlinks yet

⬆ Received 1 min. ago

🕒 Connection started 7 minutes ago

📡 UDP

Duty cycle utilization

No data available

This gateway does not have recent downlinks and cannot display the duty cycle utilization.

Network activity

Packets per data rate

Packets per channel (24 days)

Location

[Map](#)

6. 二次开发

6.1 二次开发概述

本设备内置 ubuntu 系统，理论上可以支持的二次开发方式有很多。

本设备已经内置 node-red、nodejs、python、gcc 环境，用户基于这四种开发方式，并且搭配 AI，可以很方便地进行二次开发。

GCC 交叉编译器地址:

【金山文档 | WPS 云文档】

arm-gnu-toolchain-11.3.rel1-x86_64-arm-none-linux-gnueabi.tar

<https://www.kdocs.cn/l/ca3lacsmdCT>

二次开发时可能需要登录设备控制台，可以通过串口或者 SSH 登录设备控制台

默认登录控制台:

用户名:root

密码: loragw

Chirpstack 登录:

用户名: admin

密码: admin

关于 LoRaWAN 部分的 chirpstack 部分的协议，请参考 chirpstack。

关于上下行，请参考

[MQTT - ChirpStack open-source LoRaWAN® Network Server documentation](#)

二次开发主要就是通过 MQTT 进行上下行 json 数据的订阅和发布，从而完成对 lorawan 设备的上下行。

二次开发的 Demo 仓库:

[ansitw/701SecDev](#)

二次开发相关的内容会持续迭代，增加更多 Demo

6.2 对接 LoRaWAN 数据

在 LoRaWAN 的架构中，要获取节点上行数据，或者对节点下发下行数据，都要通过 NS 进行，所以，对接 LoRaWAN 数据其实就是对接 NS 的数据接口。

6.2.1 上行

[MQTT - ChirpStack open-source LoRaWAN® Network Server documentation](#)

The MQTT integration publishes all the data it receives from the devices as JSON over MQTT. To receive data from your device, you therefore need to subscribe to its MQTT topic. For debugging, you could use a (command-line) tool like `mosquitto_sub` which is part of the Mosquitto MQTT broker.

MQTT
MQTT quickstart
Events
Scheduling a downlink

MQTT quickstart

Use `+` for a single-level wildcard, `#` for a multi-level wildcard. Examples:

```
mosquitto_sub -t "application/APPLICATION_ID/#" -v # display everything
mosquitto_sub -t "application/APPLICATION_ID/device/+event/up" -v # display only the u
```

Note:

- MQTT topics are case-sensitive
- The `APPLICATION_ID` can be retrieved using the API or using the web-interface, this is not the same as the `AppEUI` / `JoinEUI`!

Events

The MQTT integration exposes all events as documented by Event types. The default event topic is: `application/APPLICATION_ID/device/DEV_EUI/event/EVENT`.

Scheduling a downlink

简单的说

- 订阅 `application/+device/+event/+`
- data 部分解析出来，做 base64 解码即可获得上报的原始数据

例如

eg:上行数据 可订阅: application/# topic: application/7aa8a286-c592-4bc1-9479-b8c6b30f55c8/device/8dafa60100113001/event/up

```
{
  "deduplicationId": "e76fe6dd-c06a-4c15-a4b0-176220b46e92",
  "time": "2026-01-13T08:18:03.931824220+00:00",
  "deviceInfo": {
    "tenantId": "0c5a7ec6-b6f9-49f3-8745-0abaeb1ce77c",
    "tenantName": "CustomerTest",
    "applicationId": "7aa8a286-c592-4bc1-9479-b8c6b30f55c8",
    "applicationName": "eu868-test-zw",
    "deviceProfileId": "fec0e378-734f-439d-a1c6-118fe85cb9ca",
    "deviceProfileName": "EU868_OTAA_CLASSA",
    "deviceName": "EC sensor",
    "devEui": "8dafa60100113001",
    "deviceClassEnabled": "CLASS_C",
    "tags": {}
  },
  "devAddr": "000dc18f",
  "adr": false,
  "dr": 3,
  "fCnt": 23,
  "fPort": 10,
  "confirmed": true,
  "data": "AQAKAAAAGAAAAAPVW==",
  "rxInfo": [
    {
      "gatewayId": "f09e9efffe0d6fb4",
      "uplinkId": 46060,
      "nsTime": "2026-01-13T08:18:03.725344212+00:00",
      "rssi": -60,
      "snr": 12.0,
      "channel": 4,
      "location": {},
      "context": "sw4n1w==",
      "crcStatus": "CRC_OK"
    }
  ],
  "txInfo": {
    "frequency": 867300000,
    "modulation": {
      "lora": {
        "bandwidth": 125000,
        "spreadingFactor": 9,
        "codeRate": "CR_4_5"
      }
    }
  },
  "regionConfigId": "eu868"
}
```

若想获得上行的数据负载，将data部分解析出来，做base64解码即可获得上报的原始数据

6.2.2 下行

Scheduling a downlink

The default topic for scheduling downlink payloads is:

application/APPLICATION_ID/device/DEV_EUI/command/down.

The Application ID and DevEUI of the device will be taken from the topic.

Example payload:

```
{
  "devEui": "0102030405060708", // this must match the DEV_EUI of the MQTT
  "confirmed": true,           // whether the payload must be sent as con
  "fPort": 10,                 // FPort to use (must be > 0)
  "data": "...",               // base64 encoded data (plaintext, will be
  "object": {                  // decoded object (when application coded
    "temperatureSensor": {"1": 25}, // when providing the 'object', you can om
    "humiditySensor": {"1": 32}
  }
}
```

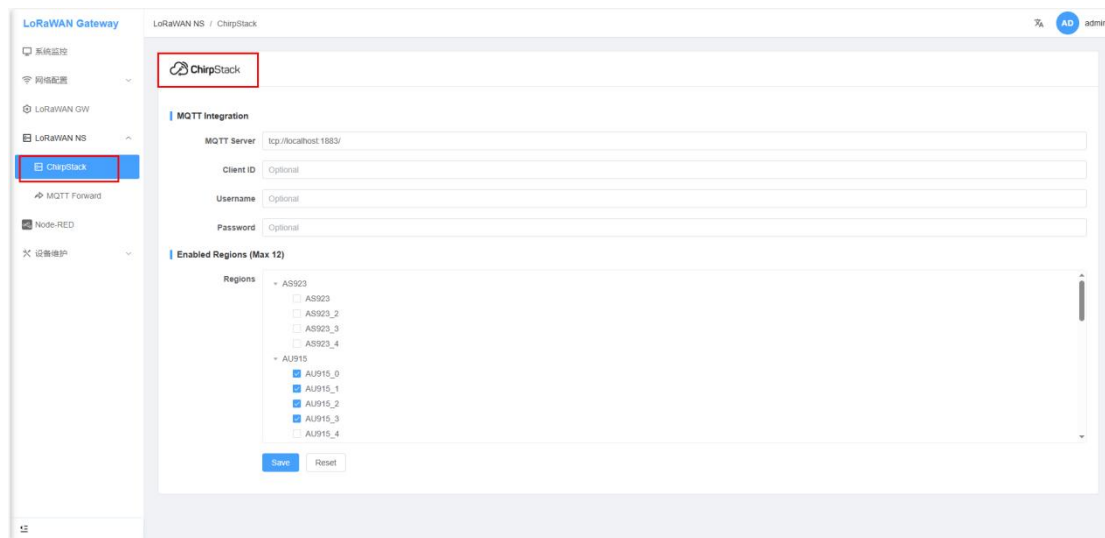
同样，data 部分也是 base64 编码，object 在未设置 codec 时，可忽略。

7. ChirpStack 的使用

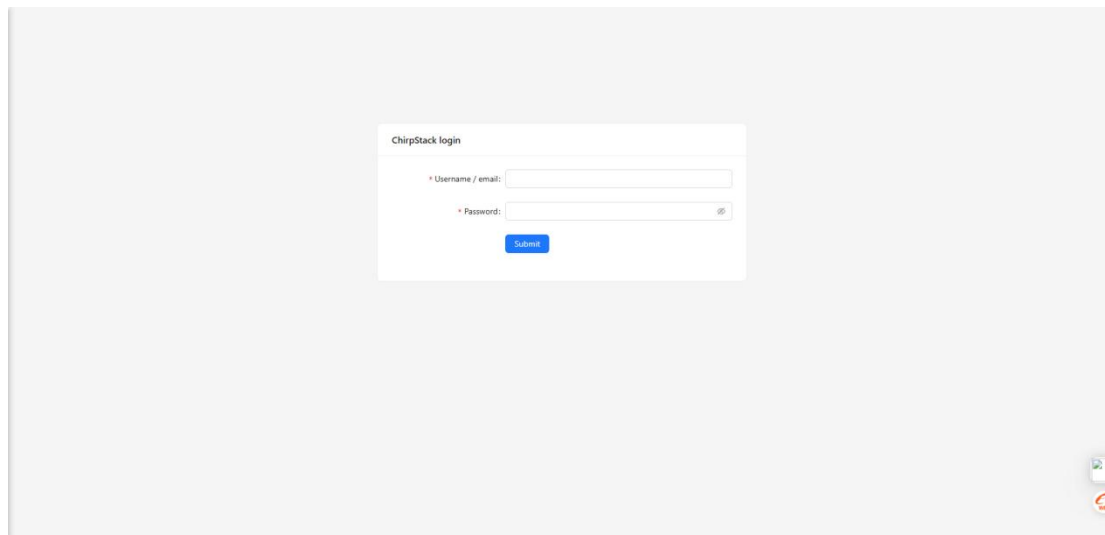
7.1 Chirpstack 简介

Chirpstack 是一个开源的 LoRaWAN NS，是一种被广泛应用的私有化部署的 LoRaWAN NS。其对 LoRaWAN 各种机制的实现都比较全面。

7.2 进入 Chirpstack



点击 chirpstack logo 跳转到网关内置 Chirpstack



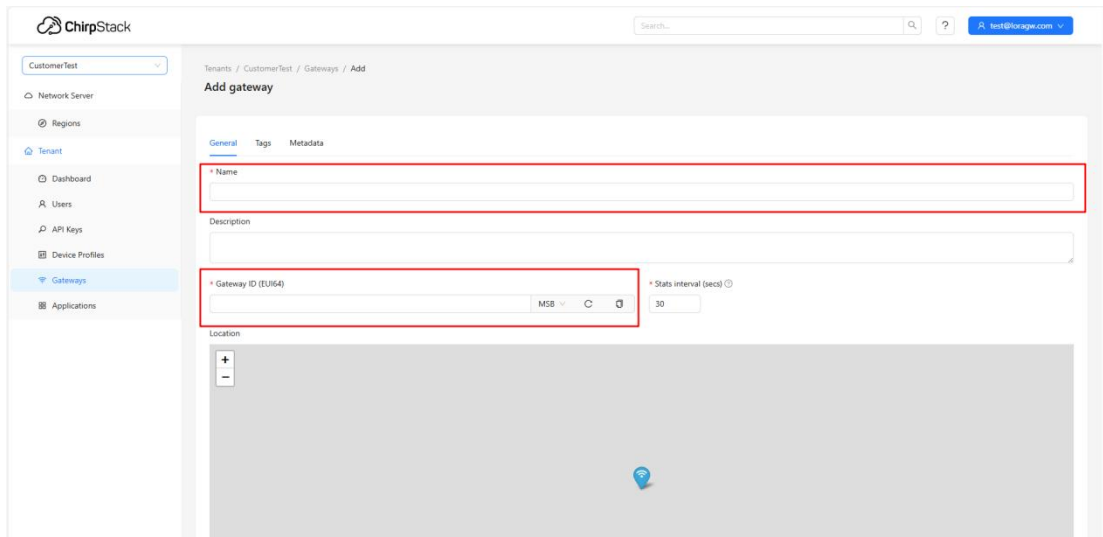
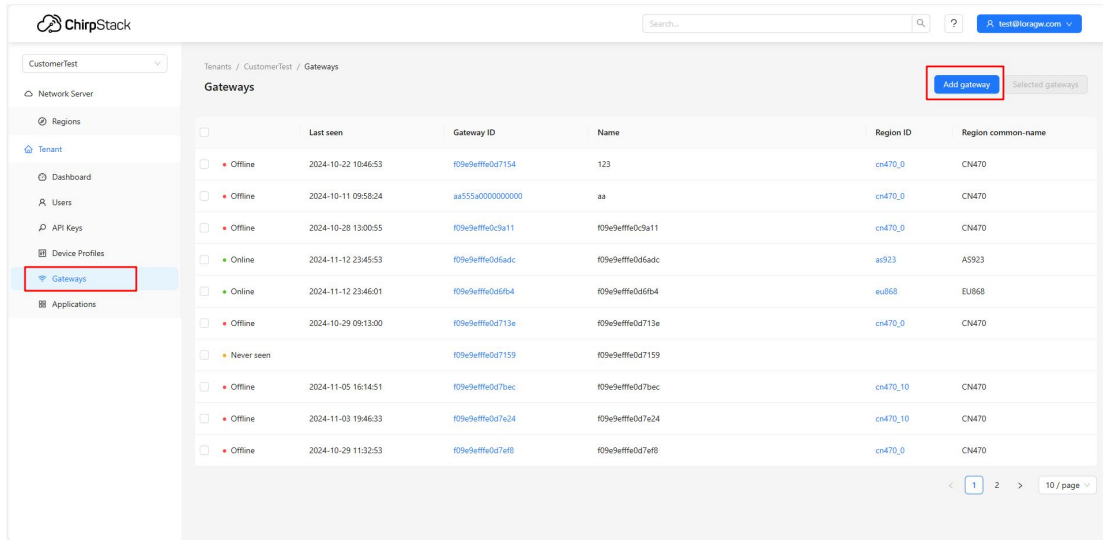
输入 chirpstack 的用户名密码进行登录。

默认用户名密码为：

用户名：admin

密码：admin

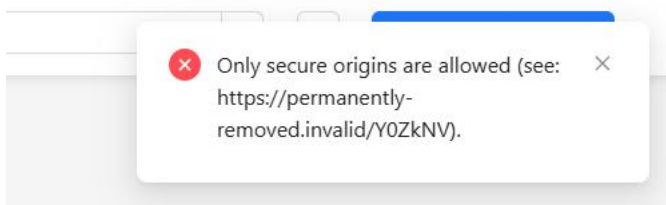
7.3 网关录入



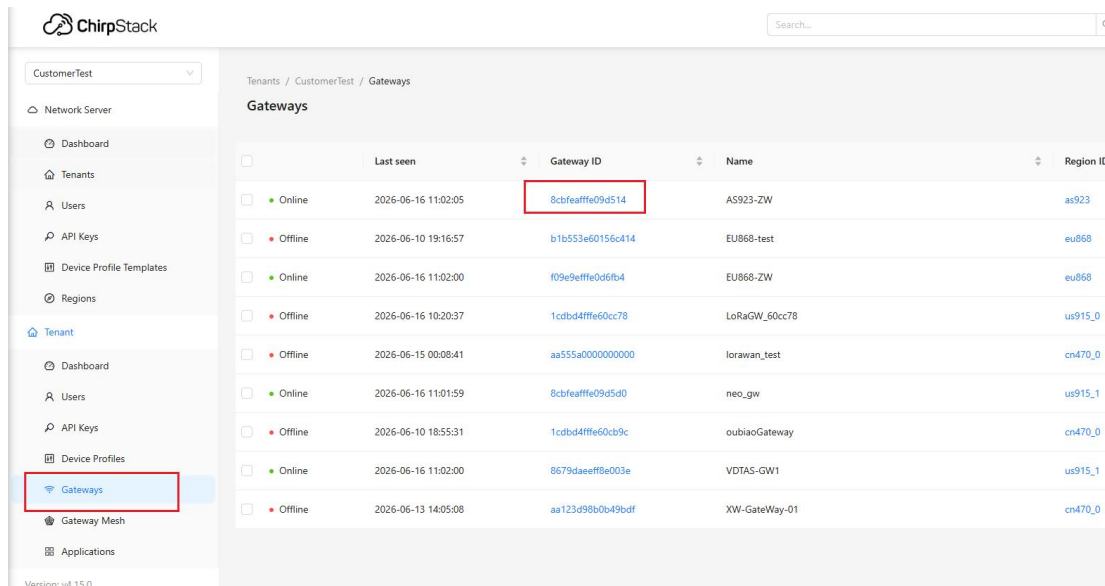
1. Name 可填写任意字符串，也可与 Gateway ID 相同；
2. Gateway ID 复制网关的 Gateway ID 填进去即可；
3. 填完后滚动页面点击下面的 Submit 提交即可。

Note:

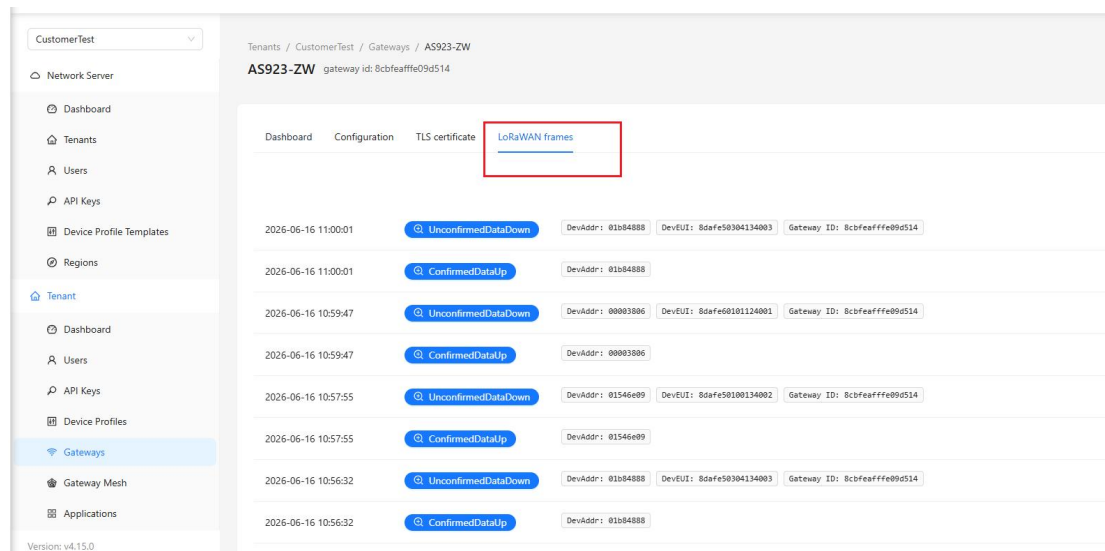
1. 点击 Add gateway 可能如下提示次错误，可忽略，此错误是因为国外的地图在国内无法使用。



7.3.1 查看网关数据



点击进入制定 gateway 后，切换到 LoRaWAN Frames 查看网关上下行的数据



7.4 设备录入

添加设备主要分为三步

1. 创建设备的 Device Profiles，如果已有可跳过
2. 创建应用，如果已有可跳过
3. 添加设备到应用，录入设备信息，选择合适的 Device Profiles

7.4.1 创建 Device Profiles

ChirpStack

CustomerTest

Network Server

Regions

Tenant

Dashboard

Users

API Keys

Device Profiles

Gateways

Applications

Add device profile

General Join (OTAA / ABP) Class-B Class-C Codec Relay Tags Measurements

Select device-profile template

* Name

Description

* Region EU868 Region configuration

* MAC version LoRaWAN 1.0.3 * Regional parameters revision A

* ADR algorithm Default ADR algorithm (LoRa only)

Flush queue on activate ☒

* Expected uplink interval (secs) 3600

Allow roaming ☐

Device-status request frequency (req/day) 1

根据设备的情况，选择合适的频段、入网方式、Class 即可

7.4.2 创建 Application

ChirpStack

CustomerTest

Network Server

Regions

Tenant

Dashboard

Users

API Keys

Device Profiles

Gateways

Applications

Tenants / CustomerTest / Applications

Applications

Add application

Name	Description
AS923	
AU915	
CN470-TEST	
cttest	
eu868-test	
TX_Test	
y9	

< 1 > 10 / page

没什么特别的，在 chirpstack 中，设备是按照 Application 为单元组织在一起的

7.4.3 录入设备

ChirpStack

Tenants / CustomerTest / Applications / CN470-TEST

CN470-TEST application id: 1e55f062-0555-457c-aa4f-5b098502d7c1

Delete application

Devices Multicast groups Relays Application configuration Integrations

Add device Selected devices

☐	Last seen	DevEUI	Name	Device profile	Battery
☐	2024-11-03 17:47:09	20c8a8ffe010101	20C8A8FFE010101	OTAA-CN470-CLASSA	
☐	2024-09-13 00:29:31	70b3d57ed0055489	70B3D57ED0055489	OTAA-CN470-CLASSA	
☐	2024-11-05 23:59:38	70b3d57ed005548a	70B3D57ED005548A	OTAA-CN470-CLASSA	
☐	Never	693b21d30e1d72e4	mygg	OTAA-CN470-CLASSA	
☐	2024-09-24 10:58:26	2233000000000000	otaa	#	
☐	Never	ffff100004b7e5	xyz_test	OTAA-CN470-CLASSA	

< 1 > 10 / page

Add device

Device Tags Variables

* Name

Description

* Device EUI (EUI64)

MSB C U

Join EUI (EUI64)

MSB C U

* Device profile

Device is disabled

Disable frame-counter validation

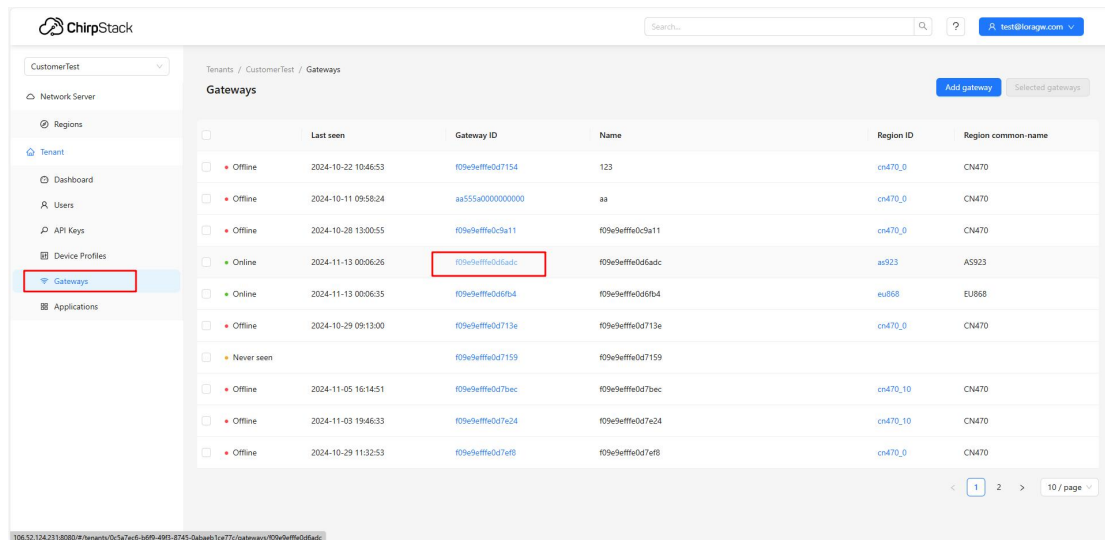
Submit

Note:

1. 录入设备是需要录入 key 的，OTAA 和 ABP 需要录入不同的 key；
2. OTAA 的 APPEUI(JoinEUI)因为是明文，所以可以不用录入。

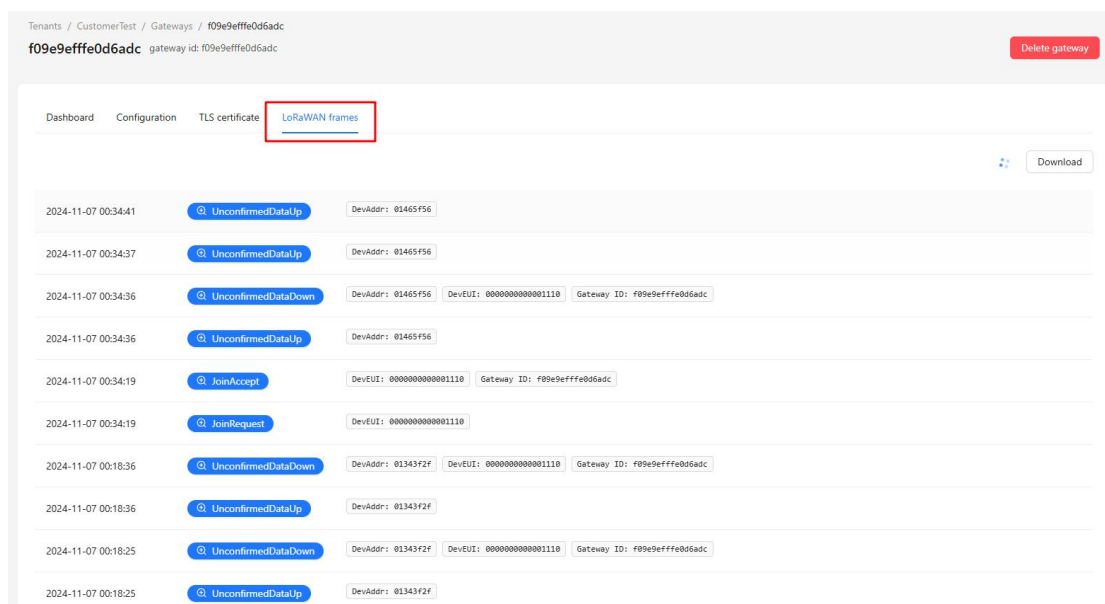
7.5 查看数据

7.5.1 查看网关数据



The screenshot shows the ChirpStack web interface. On the left is a sidebar with navigation links: Network Server, Regions, Tenant (selected), Dashboard, Users, API Keys, Device Profiles, Gateways (highlighted with a red box), and Applications. The main area displays a table of gateways under the path 'Tenants / CustomerTest / Gateways'. The table has columns for 'Last seen', 'Gateway ID', 'Name', 'Region ID', and 'Region common-name'. One gateway, 'f09e9effe0d6adc', is highlighted with a red box. Below the table is a pagination bar showing '1' of 10 pages.

	Last seen	Gateway ID	Name	Region ID	Region common-name
☐	Offline 2024-10-22 10:46:53	f09e9effe0d7154	123	cn470_0	CN470
☐	Offline 2024-10-11 09:58:24	aa555a0000000000	aa	cn470_0	CN470
☐	Offline 2024-10-28 13:00:55	f09e9effe0c3a11	f09e9effe0c3a11	cn470_0	CN470
☐	Online 2024-11-13 00:06:26	f09e9effe0d6adc	f09e9effe0d6adc	as923	AS923
☐	Online 2024-11-13 00:06:35	f09e9effe0d6fb4	f09e9effe0d6fb4	eu868	EU868
☐	Offline 2024-10-29 09:13:00	f09e9effe0d713e	f09e9effe0d713e	cn470_0	CN470
☐	Never seen	f09e9effe0d7159	f09e9effe0d7159		
☐	Offline 2024-11-05 16:14:51	f09e9effe0d7bec	f09e9effe0d7bec	cn470_10	CN470
☐	Offline 2024-11-03 19:46:33	f09e9effe0d7e24	f09e9effe0d7e24	cn470_10	CN470
☐	Offline 2024-10-29 11:32:53	f09e9effe0d7ef8	f09e9effe0d7ef8	cn470_0	CN470

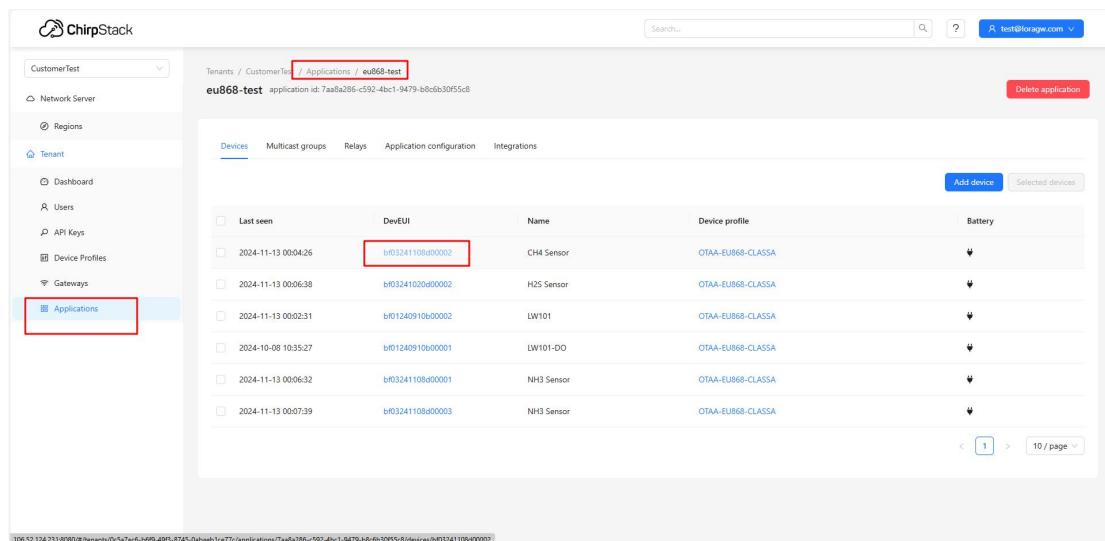


The screenshot shows the 'LoRaWAN frames' page in ChirpStack. The breadcrumb path is 'Tenants / CustomerTest / Gateways / f09e9effe0d6adc'. The page title is 'f09e9effe0d6adc gateway id: f09e9effe0d6adc'. There is a 'Delete gateway' button. Below the title are tabs for 'Dashboard', 'Configuration', 'TLS certificate', and 'LoRaWAN frames' (selected and highlighted with a red box). A 'Download' button is on the right. The main area displays a list of frames with columns for 'Last seen', 'Action', 'DevAddr', 'DevEUI', and 'Gateway ID'. The actions include 'UnconfirmedDataUp', 'UnconfirmedDataDown', 'JoinAccept', and 'JoinRequest'. The gateway ID for all frames is 'f09e9effe0d6adc'.

Last seen	Action	DevAddr	DevEUI	Gateway ID
2024-11-07 00:34:41	UnconfirmedDataUp	01465F56		
2024-11-07 00:34:37	UnconfirmedDataUp	01465F56		
2024-11-07 00:34:36	UnconfirmedDataDown	01465F56	0000000000001118	f09e9effe0d6adc
2024-11-07 00:34:36	UnconfirmedDataUp	01465F56		
2024-11-07 00:34:19	JoinAccept		0000000000001118	f09e9effe0d6adc
2024-11-07 00:34:19	JoinRequest		0000000000001118	
2024-11-07 00:18:36	UnconfirmedDataDown	01343F2F	0000000000001118	f09e9effe0d6adc
2024-11-07 00:18:36	UnconfirmedDataUp	01343F2F		
2024-11-07 00:18:25	UnconfirmedDataDown	01343F2F	0000000000001118	f09e9effe0d6adc
2024-11-07 00:18:25	UnconfirmedDataUp	01343F2F		

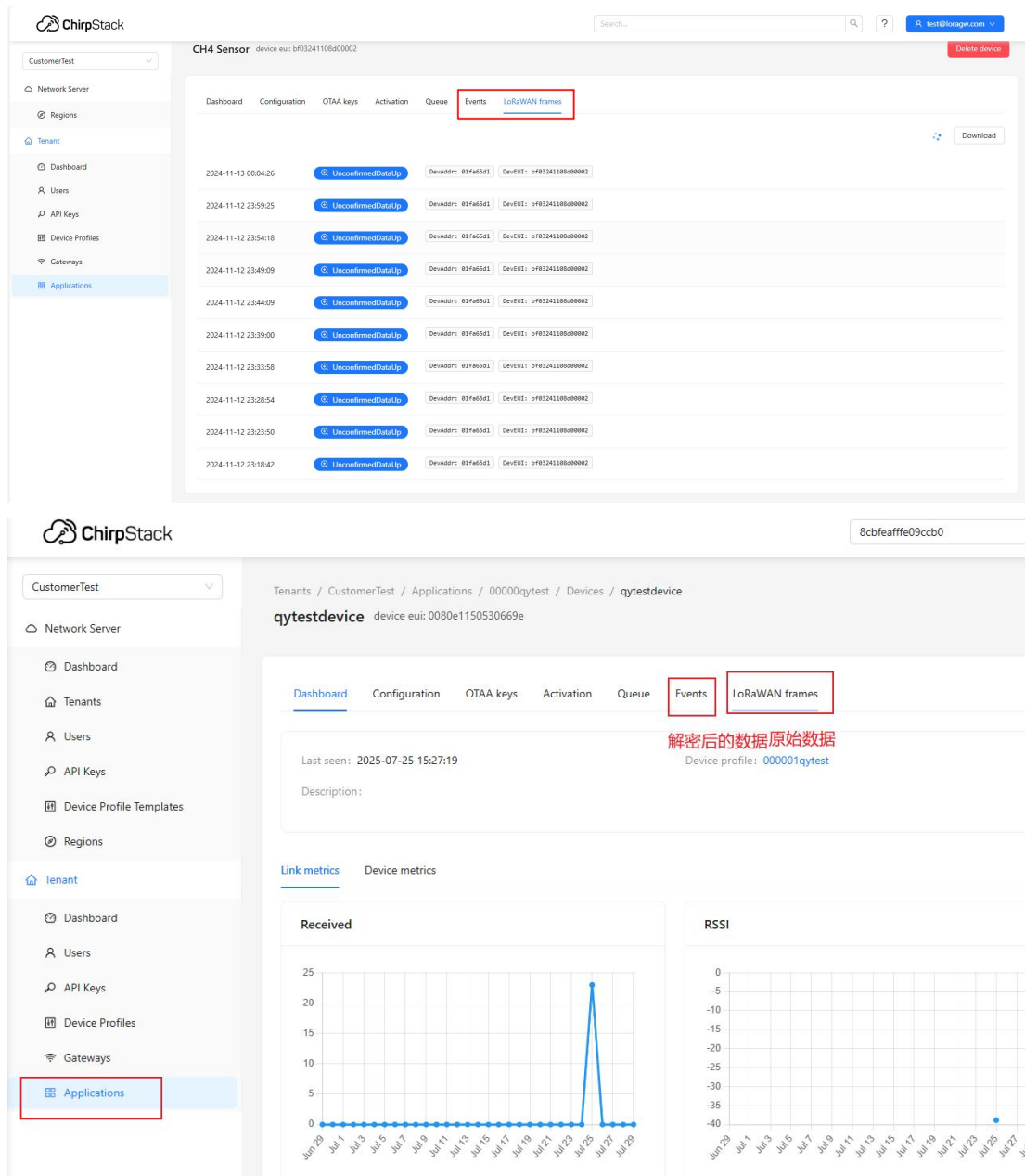
此处查看的数据为网关收发的原始数据，所以是加密的

7.5.2 查看节点数据



The screenshot shows the ChirpStack web interface. On the left is a sidebar with navigation links: Network Server, Regions, Tenant (selected), Dashboard, Users, API Keys, Device Profiles, Gateways, and Applications (highlighted with a red box). The main area displays the 'Applications' page under the path 'Tenants / CustomerTest / Applications / eu868-test'. The page title is 'eu868-test application id: 7aa8a286~592~4bc1~9479~b8c6b30f55c8'. There is a 'Delete application' button. Below the title are tabs for 'Devices', 'Multicast groups', 'Relays', 'Application configuration', and 'Integrations'. The 'Devices' tab is selected. A table lists devices with columns for 'Last seen', 'DevEUI', 'Name', 'Device profile', and 'Battery'. The DevEUI 'b0324110b00002' is highlighted with a red box. The battery status is indicated by a downward arrow.

Last seen	DevEUI	Name	Device profile	Battery
2024-11-13 00:04:26	b0324110b00002	CH4 Sensor	OTAA-EU868-CLASSA	↓
2024-11-13 00:06:38	b0324110b000002	H25 Sensor	OTAA-EU868-CLASSA	↓
2024-11-13 00:02:31	b01240910b00002	LW101	OTAA-EU868-CLASSA	↓
2024-10-08 10:55:27	b01240910b00001	LW101-DO	OTAA-EU868-CLASSA	↓
2024-11-13 00:06:32	b0324110b000001	NH3 Sensor	OTAA-EU868-CLASSA	↓
2024-11-13 00:07:39	b0324110b000003	NH3 Sensor	OTAA-EU868-CLASSA	↓

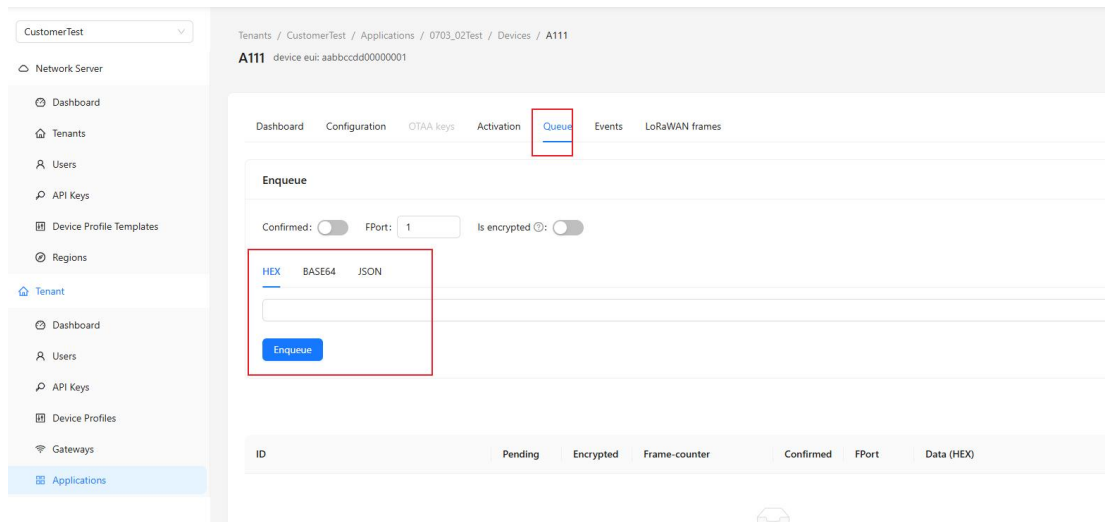


在 Events 和 LoRaWAN frames 中都可以看到设备的上下行数据，其中 LoRaWAN frames 中看到的会更原始，每次数据收发都可以看到，包括 ADR 等 Mac Commnad 都可以看到。

Note:

1. 因为 Chirpstack 会在节点入网后根据所连接的网关的频点下发 ADR 去调节节点的 ChannelMask，所以如果网关的频点和实际端口所代表的频点不一致，会导致节点被修改成不能通讯的频点，从而导致通讯出错。

7.6 数据下行



在如图所示位置填写数据，点击 enqueue，NS 会将下行数据入队，然后根据下发时候进行下发。
根据节点的 CLASS 类型不同

- Class A 设备，NS 会在设备有上行数据的时候将数据下发下去
- Class C 设备，NS 会直接在 RX2 下发数据，会立即下发

- FPort: 类似 TCP/IP 的 Port，NS 和终端可以通过 Port 来区分不同的数据类型
- Confirmed: 数据包是否为确认模式

7.7 MQTT 对接设备上下行

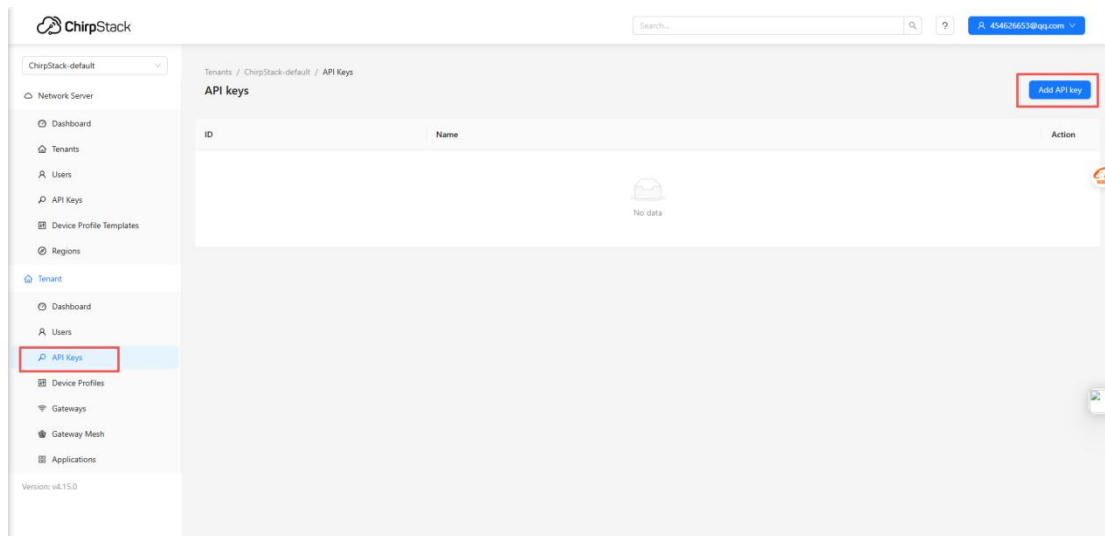
参考 [6.2 对接 LoRaWAN 数据](#)

7.8 chirpstack-rest-api 对接

用户访问 chirpstack 所在服务器的 8090 端口，例如 <http://192.168.168.1:8090>，可以看到 api 页面，该页面有 Swagger UI，罗列了基本所有的 API

用户可在次页面上进行调试

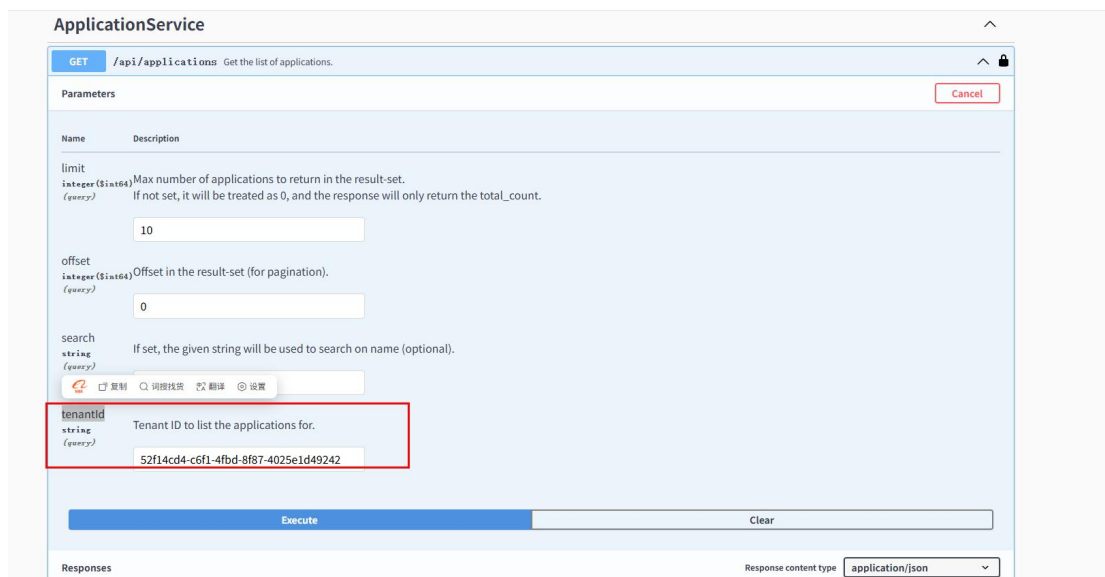
注意 API 需要授权，授权需要申请 APIKEY，方式如下



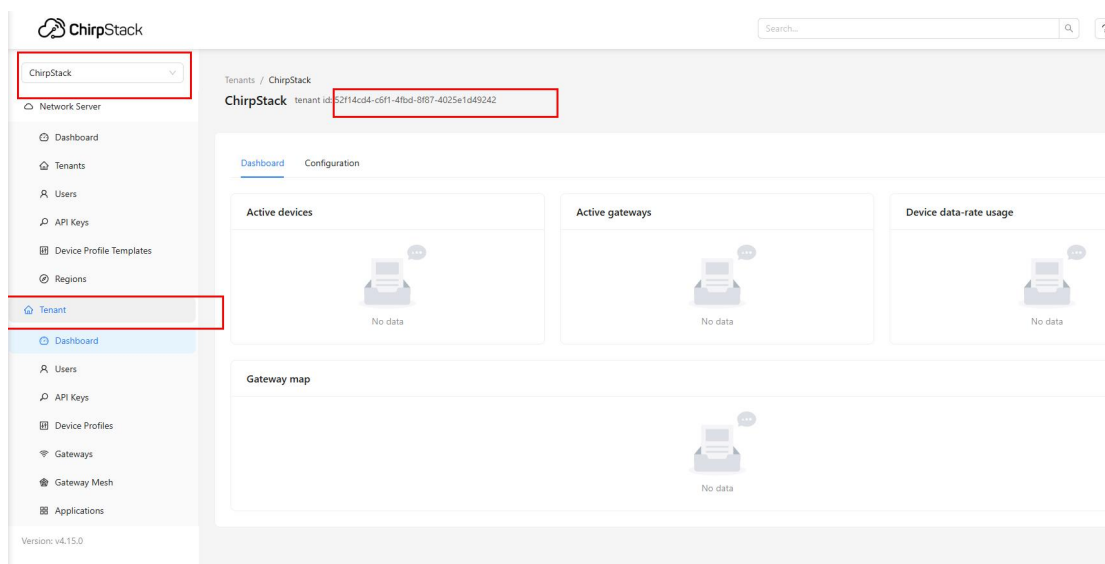
实际的 key 为 Bearer <API TOKEN>

注意需要在申请到的 Token 前面加上 Bearer ，例如 API Token 为 12345，那么 key 就是 Bearer 12345

API 的 tenantId,

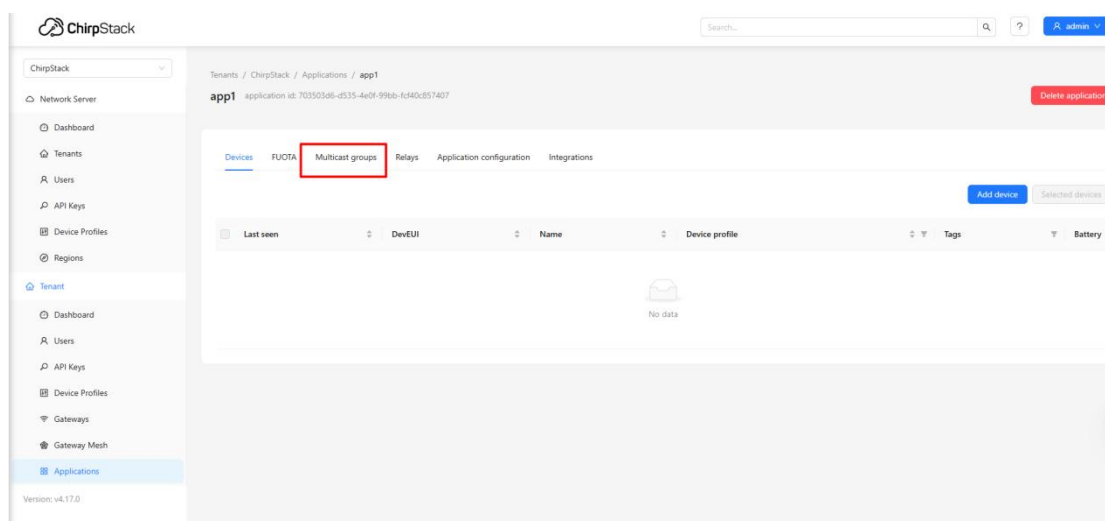


在 tenant 的首页可以获取到，chirpstack 类似多租户架构，很多时候要输入租户的 ID 才能正常获取到参数



7.9 组播

7.9.1 组播的使用方法



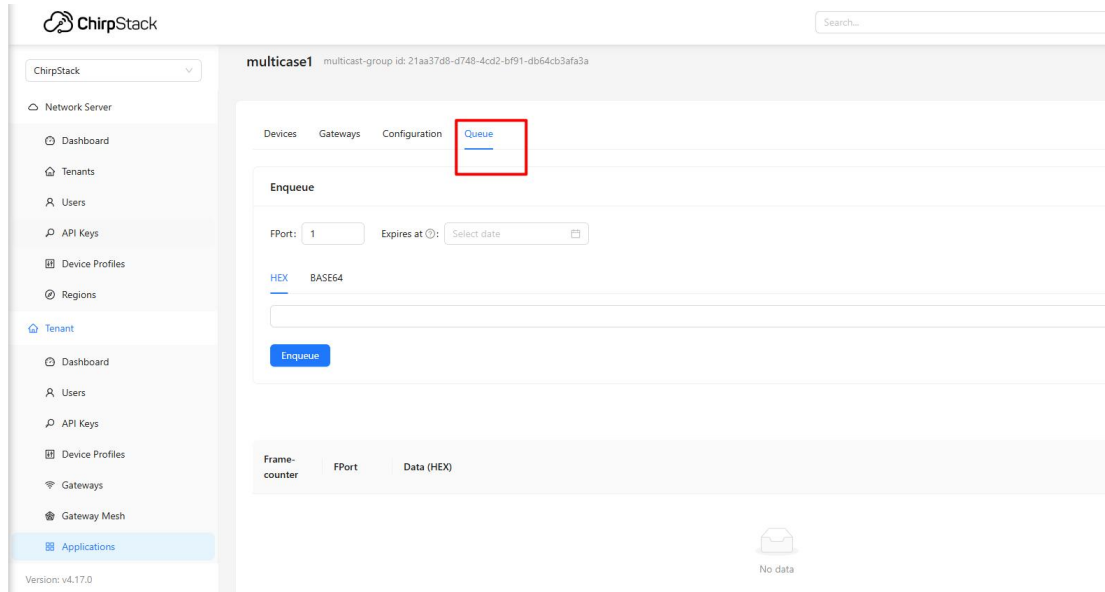
The screenshot shows the ChirpStack web interface. The left sidebar contains a navigation menu with options like Dashboard, Tenants, Users, API Keys, Device Profiles, Regions, Tenant, Gateways, and Applications. The main content area is titled 'multicast1' and shows the 'Configuration' tab. The configuration fields include: Multicast-group name (multicast1), Multicast address (11223344), Multicast network session key (2b7e15162bae2afab47158809cf4f3e), Multicast application session key (2b7e15162bae2afab47158809cf4f3e), Region (CN470), Data-rate (0), Frame-counter (0), Frequency (Hz) (505300000), Group type (Class-C), Class-B ping-slot periodicity (Every second), and Class-C scheduling type (Delay). A 'Submit' button is at the bottom.

注意:

1. CLASSC 设备组播的参数为 RX2，即 CLASSC 的默认下行通道，故此时需要按照 RX2 的参数配置

The screenshot shows the ChirpStack web interface with the 'Devices' tab selected. The tab is highlighted with a red box. The main content area shows a table with columns 'Name' and 'DevEUI'. The table is currently empty, and a 'No data' message is displayed below it. A 'Remove from multicast-group' button is visible in the top right corner.

将 Device 和 Gateway 添加到组里面



可在此处下发数据

7.9.2 MQTT 下发组播数据

(此功能需 V1.1.15 版本以上才会支持)

Topic

固定 topic:

```
application/{application_id}/multicast/{multicast_group_id}
```

示例:

```
application/a25210c1-2342-408a-b84b-8f9846251a17/multicast/26e341ba-d8e5-4d98-871d-5166e57b45a9
```

说明:

- application_id 必须是组播组所属 Application 的 UUID。
- multicast_group_id 必须是目标组播组的 UUID。

该入口只支持 enqueue, 不支持 flush、list 或其他 multicast 管理操作。

Payload 必须是 JSON, 格式与 REST API 的 EnqueueMulticastGroupQueueItemRequest 保持一致:

```
{
  "queueItem": {
    "multicastGroupId": "26e341ba-d8e5-4d98-871d-5166e57b45a9",
    "fPort": 10,
    "data": "AQIDBA=="
  }
}
```

字段说明:

- `queueItem`: 必填。
- `queueItem.multicastGroupId`: 必填, 且必须与 `topic` 中的 `{multicast_group_id}` 完全一致。
- `queueItem.fPort`: 必填。
- `queueItem.data`: 必填, 必须是 base64 编码后的二进制数据。
- `queueItem.expiresAt`: 可选, 格式为 RFC3339 时间字符串。

带 `expiresAt` 的示例:

```
{
  "queueItem": {
    "multicastGroupId": "26e341ba-d8e5-4d98-871d-5166e57b45a9",
    "fPort": 10,
    "data": "AQIDBA==",
    "expiresAt": "2030-01-01T00:00:00Z"
  }
}
```

行为说明

收到 MQTT 消息后, ChirpStack 会执行以下校验:

- `topic` 中的 `application_id` 必须是合法 UUID。
- `topic` 中的 `multicast_group_id` 必须是合法 UUID。
- `payload` 必须是合法 JSON。
- `queueItem` 必须存在。
- `queueItem.multicastGroupId` 必须与 `topic` 中的 `multicast_group_id` 一致。
- 组播组的 `application_id` 必须与 `topic` 中的 `application_id` 一致。
- `data` 必须是合法 base64。
- `payload size` 必须满足当前 `multicast MAC / RP` 配置下的最大下行长度限制。

通过校验后, 消息会复用现有组播入队逻辑, 进入 `multicast queue`。

7.10 快速录入单个/批量设备

见 [2.2.3.3 快速添加设备](#)

(此需 1.1.19 版本以上软件支持)