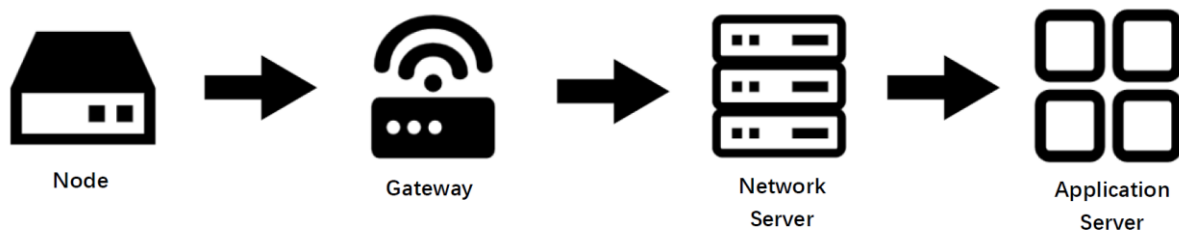


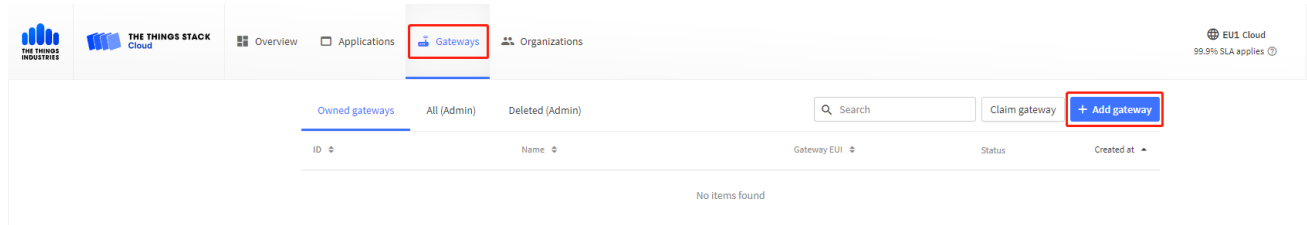
1. Overview

A typical LoRaWAN architecture consists of a node, gateway, network server, and application server. The architecture is as follows. This document mainly introduces how to add a Lansitec LoRaWAN end device to a LoRaWAN network. The network server used is [The Things Network](#) (TTN).



2. Connect the LoRa gateway to the TTN platform

2.1 Configure TTN platform



- Fill in **the gateway EUI you can get in Section 2.2.2** to Gateway EUI and click **Confirm***;
- Fill in **your gateway name** to Gateway name*;
- Select **your frequency band (e.g., US902-928, FSB 1)** from the Frequency plan*.

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 ⓘ
60 C5 A8 FF FE 74 D4 6A Reset

Gateway ID ⓘ *
eui-60c5a8fffe74d46a

Gateway name ⓘ
RAK_test_gateway

Frequency plan ⓘ *
United States 902-928 MHz, FSB 1 | v

+ Add frequency plan

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

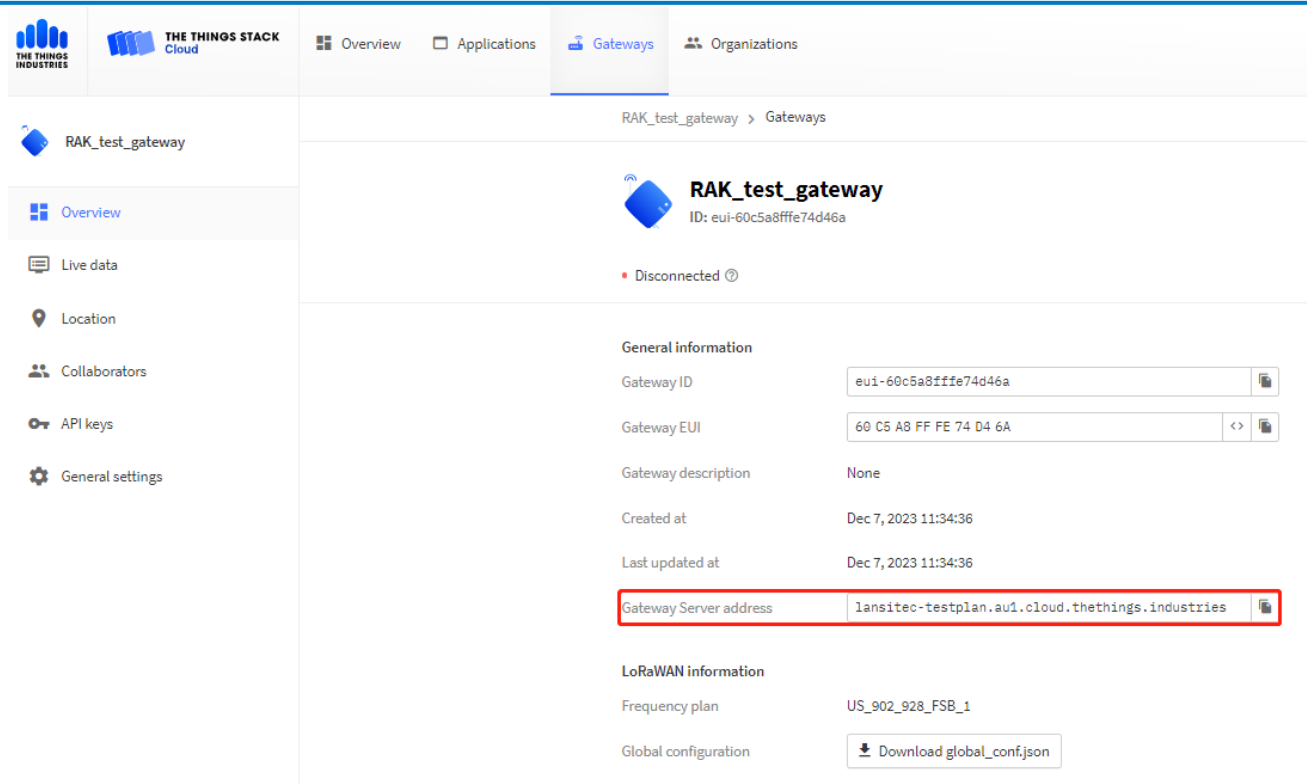
☐ Require authenticated connection ⓘ
Choose this option eg. if your gateway is powered by [LoRa Basic Station](#) ⓘ

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

- Copy the Gateway Server address* to be used in **Section 2.2.2**;



THE THINGS STACK Cloud

Overview Applications Gateways Organizations

RAK_test_gateway > Gateways

RAK_test_gateway
ID: eui-60c5a8fffe74d46a

Disconnected ⓘ

General information

Gateway ID	eui-60c5a8fffe74d46a
Gateway EUI	60 C5 A8 FF FE 74 D4 6A
Gateway description	None
Created at	Dec 7, 2023 11:34:36
Last updated at	Dec 7, 2023 11:34:36
Gateway Server address	lansitec-testplan.au1.cloud.thethings.industries

LoRaWAN information

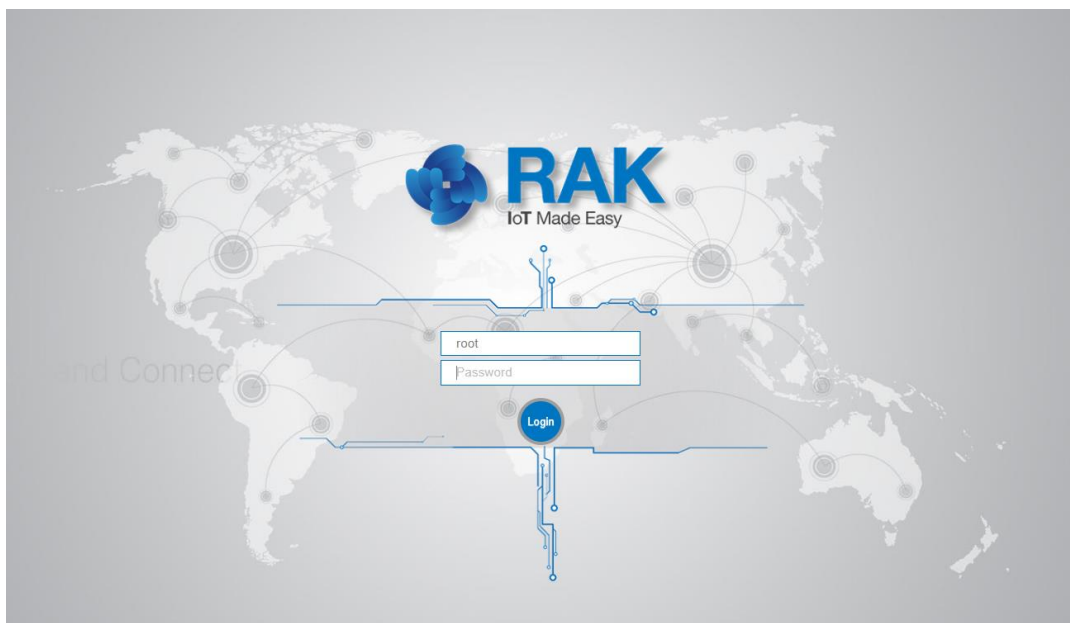
Frequency plan	US_902_928_FSB_1
Global configuration	Download global_conf.json

2.2 Configure the Gateway

2.2.1 Log in RAK Gateway configuration UI

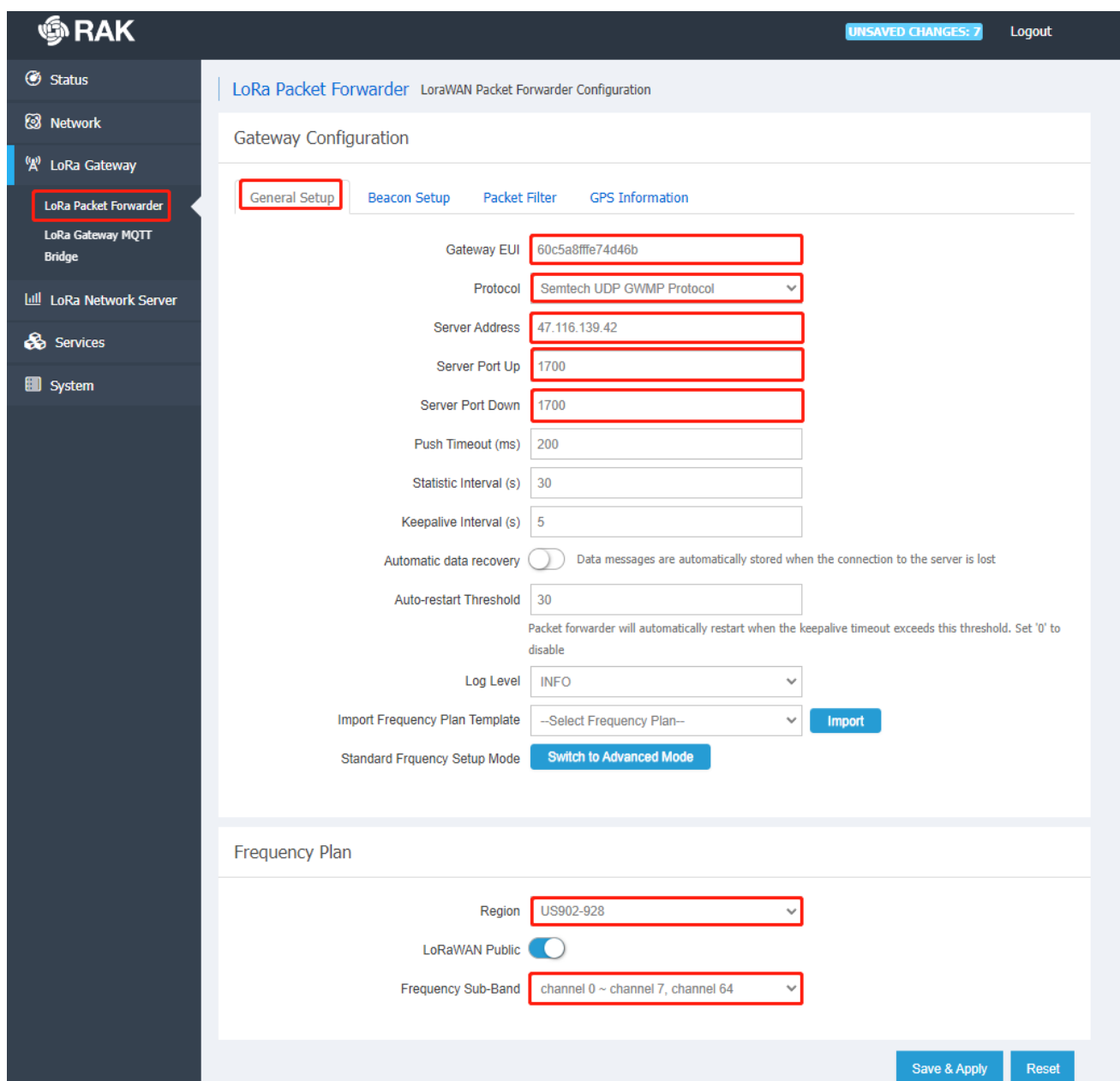
Username: root

Password: root



2.2.2 Configure LoRa Packet Forwarder*

- Copy Gateway EUI to be used in **Section 2.1**;
- Select **Semtech UDP GWMP Protocol** from Protocol*;
- Fill in **your TTN server IP address got in Section 2.1** to Server Address*;
- Fill in **the uplink port and the default downlink port 1700** to Server Port*;
- Select **the frequency band you used (e.g., US905-928)** from Region*;
- Select **the frequency channels you used (e.g., channel 0~channel 7, channel 64)** from Sub-Band*;
- Save&Apply* and reboot* gateway;



RAK UNSAVED CHANGES: 7 [Logout](#)

LoRa Packet Forwarder LoraWAN Packet Forwarder Configuration

Gateway Configuration

General Setup Beacon Setup Packet Filter GPS Information

Gateway EUI: 60c5a8ffe74d46b

Protocol: Semtech UDP GWMP Protocol

Server Address: 47.116.139.42

Server Port Up: 1700

Server Port Down: 1700

Push Timeout (ms): 200

Statistic Interval (s): 30

Keepalive Interval (s): 5

Automatic data recovery: ☐ Data messages are automatically stored when the connection to the server is lost

Auto-restart Threshold: 30

Packet forwarder will automatically restart when the keepalive timeout exceeds this threshold. Set '0' to disable

Log Level: INFO

Import Frequency Plan Template: --Select Frequency Plan-- [Import](#)

Standard Frequency Setup Mode: [Switch to Advanced Mode](#)

Frequency Plan

Region: US902-928

LoRaWAN Public: ☒

Frequency Sub-Band: channel 0 ~ channel 7, channel 64

[Save & Apply](#) [Reset](#)

2.3 Gateway connected

The screenshot shows the 'Gateways' section of the The Things Stack Cloud interface. The top navigation bar includes 'Overview', 'Applications', 'Gateways', and 'Organizations'. The 'Gateways' tab is active. Below the navigation bar, there are tabs for 'Owned gateways', 'All (Admin)', and 'Deleted (Admin)'. A search bar and buttons for 'Claim gateway' and '+ Add gateway' are also present. The main content area displays a table of gateways. A single gateway is listed, highlighted with a red border:

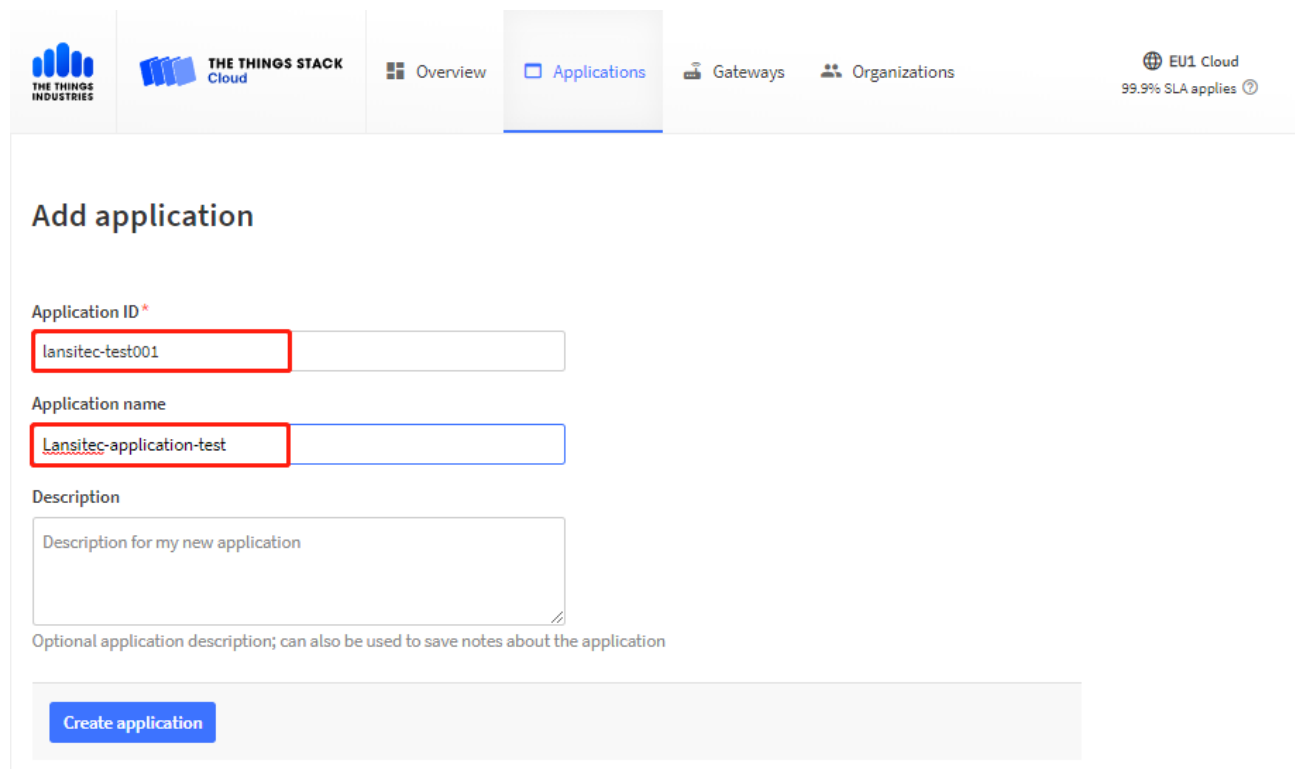
ID	Name	Gateway EUI	Status	Created at
lansitec-office-gateway01	RAK_Test_gateway	68 C5 A8 FF FE 74 D4 68	Connected	9 minutes ago

3. Connect the LoRa device to TTN

3.1 Configure TTN

3.1.1 Add Application*

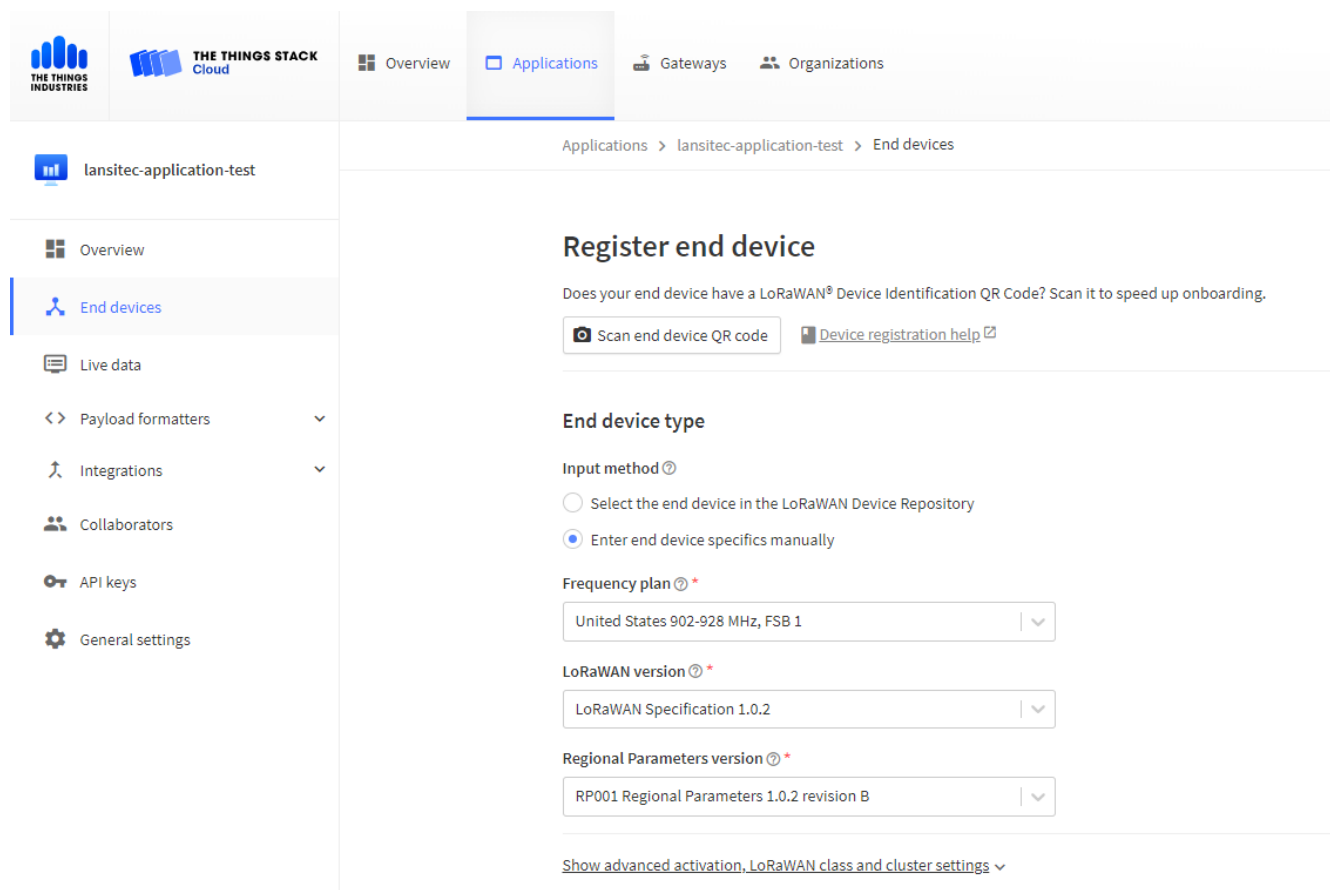
- Fill in **your application ID** to Application ID*;
- Fill in **your application name** to Application name*.



The screenshot shows the 'Add application' interface in the TTN console. The top navigation bar includes logos for 'THE THINGS INDUSTRIES' and 'THE THINGS STACK Cloud', along with menu items for 'Overview', 'Applications' (highlighted), 'Gateways', and 'Organizations'. On the right, it indicates 'EU1 Cloud' with a '99.9% SLA applies' note. The main form area is titled 'Add application' and contains three input fields: 'Application ID*' with the value 'lansitec-test001', 'Application name' with the value 'Lansitec-application-test', and a 'Description' field with the placeholder text 'Description for my new application'. Below the description field is a note: 'Optional application description; can also be used to save notes about the application'. At the bottom of the form is a blue 'Create application' button.

3.1.2 Add Device*

- Select **the Enter end device specifics manually** from the Input method*;
- Select **the frequency plan you used (e.g., US902-928, FSB 1)** from the Frequency plan*;
- Select **the LoRaWAN version 1.0.2** from the Frequency plan*;
- Select **the Regional parameter 1.0.2 revision B** from Regional Parameters version*;



The screenshot shows the TTN (The Things Network) web interface. The top navigation bar includes 'Overview', 'Applications' (selected), 'Gateways', and 'Organizations'. The left sidebar shows the application 'lansitec-application-test' with a menu including 'Overview', 'End devices' (selected), 'Live data', 'Payload formatters', 'Integrations', 'Collaborators', 'API keys', and 'General settings'. The main content area is titled 'Register end device' and contains the following fields:

- Does your end device have a LoRaWAN® Device Identification QR Code? Scan it to speed up onboarding.**
 -
 - [Device registration help](#)
- End device type**
 - Input method**
 - ☐ Select the end device in the LoRaWAN Device Repository
 - ☒ Enter end device specifics manually
 - Frequency plan**
 - United States 902-928 MHz, FSB 1
 - LoRaWAN version**
 - LoRaWAN Specification 1.0.2
 - Regional Parameters version**
 - RP001 Regional Parameters 1.0.2 revision B
- [Show advanced activation, LoRaWAN class and cluster settings](#)

- Fill in **your application EUI** to the Join EUI and click **Confirm***;
- Fill in **your device EUI** to the DevEUI*;
- Fill in **your application key** to the Appkey*.

lansitec-application-test

Overview

End devices

Live data

Payload formatters

Integrations

Collaborators

API keys

General settings

Frequency plan [?]*

United States 902-928 MHz, FSB 1

LoRaWAN version [?]*

LoRaWAN Specification 1.0.2

Regional Parameters version [?]*

RP001 Regional Parameters 1.0.2 revision B

Show advanced activation, LoRaWAN class and cluster settings

Provisioning information

JoinEUI [?]*

70 B3 D5 A4 D0 16 00 01

Reset

This end device can be registered on the network

DevEUI [?]*

70 B3 D5 A4 D2 10 02 E3

Generate

1/50 used

AppKey [?]*

C9 D1 B8 A9 FE A3 E2 76 08 76 A1 A5 81 20 BF 5D

Generate

End device ID [?]*

eui-70b3d5a4d21002e3

This value is automatically prefilled using the DevEUI

After registration

☒ View registered end device
 ☐ Register another end device of this type

Register end device

3.2 Reboot the Device

- Reboot the device;
- The device is connected to the TTN successfully

THE THINGS STACK Cloud

Overview Applications Gateways Organizations

AI1 Cloud

99.9% SLA applies

Martin Ma

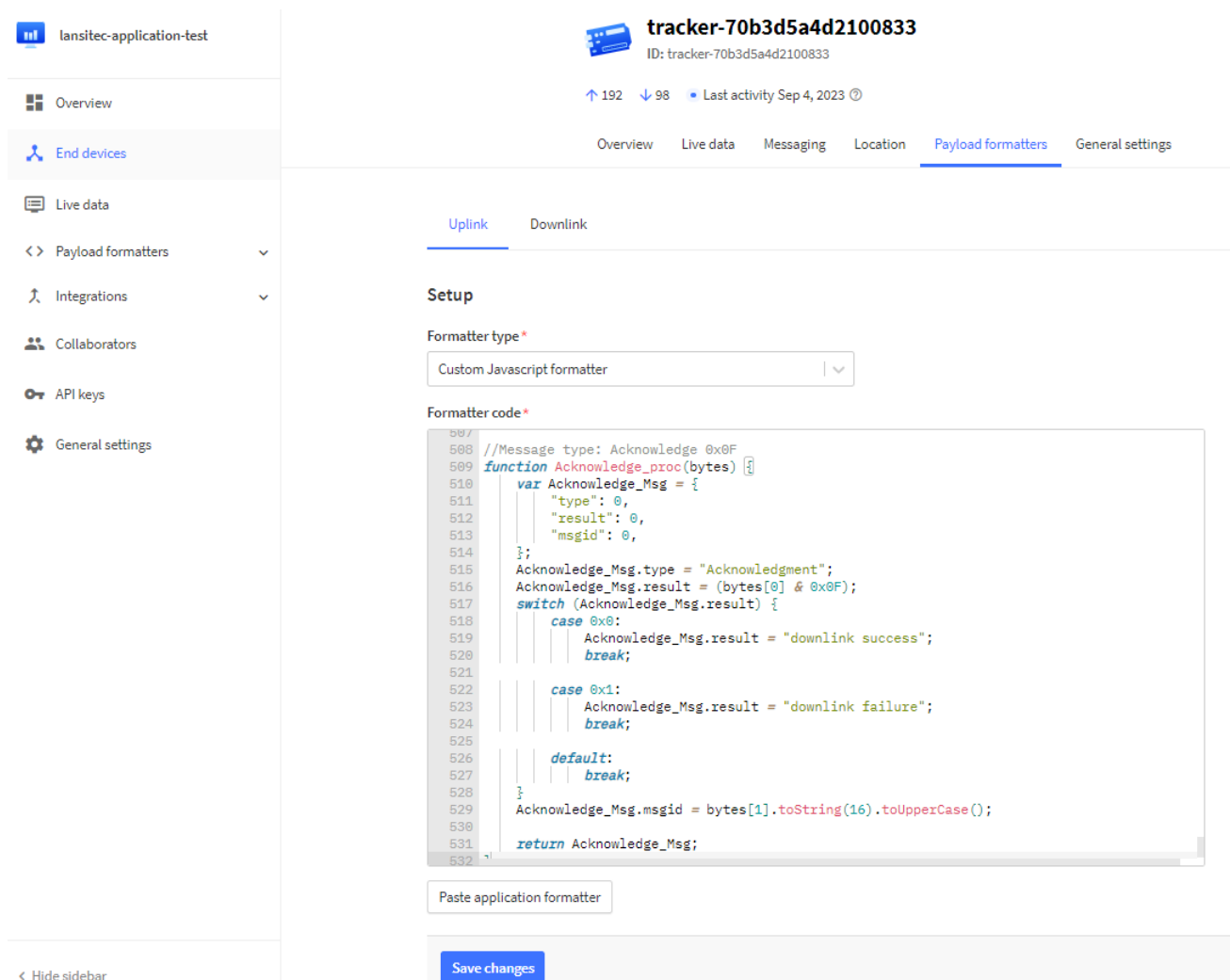
Applications > Lansitec-application-test > Live data

Time	Entity ID	Type	Data preview
17:19:15	eui-70b3d5a4d21002e3	Forward uplink data message	DevAddr: 27 FE 07 16 Payload: [bytes: [115,0,0,0,0,255,246,255,248,172,255,250,255,252,171,0,1,0,2,169]] 73 00 00 00 00 00 FF F6
17:18:15	eui-70b3d5a4d21002e3	Forward uplink data message	DevAddr: 27 FE 07 16 Payload: [bytes: [115,0,0,0,0,16,0,97,174,1,2,3,4,173,255,250,255,252,171]] 73 00 00 00 00 00 10 -
17:17:12	eui-70b3d5a4d21002e3	Forward uplink data message	DevAddr: 27 FE 07 16 Payload: [bytes: [16,0,160,40,0,12,10,5,32,2,5,4]] 10 00 A0 20 00 0C 0A 05 -
17:17:08	eui-70b3d5a4d21002e3	Forward join-accept message	DevAddr: 27 FE 07 16
17:17:06	eui-70b3d5a4d21002e3	Accept join-request	DevAddr: 27 FE 07 16
17:16:57	eui-70b3d5a4d21002e3	Update end device	["activated_at"]
17:16:57	eui-70b3d5a4d21002e3	Forward uplink data message	DevAddr: 27 FE 07 1F Payload: [bytes: [16,0,160,40,0,12,10,5,32,2,5,4]] 10 00 A0 20 00 0C 0A 05 -
17:16:53	eui-70b3d5a4d21002e3	Forward join-accept message	DevAddr: 27 FE 07 1F
17:16:51	eui-70b3d5a4d21002e3	Accept join-request	DevAddr: 27 FE 07 1F
16:09:30	eui-70b3d5a4d21002e3	Create end device	

3.3 Add Decoder File

- Select **the Payload formatters** from the device page*;
- Select **the Custom Javascript formatter** from the Formatter type in the Setup area on the left *;

- Contact Lansitec staff and get the credentials to log in and find the payload decoder on the DownloadCenter: <https://www.lansitec.com/download/DownloadCentre.html>, and paste the decoder file content into the Foematter code column*;



The screenshot shows the Lansitec application interface for a device named "tracker-70b3d5a4d2100833". The left sidebar contains navigation options: Overview, End devices, Live data, Payload formatters, Integrations, Collaborators, API keys, and General settings. The main area displays the "Payload formatters" tab for the selected device. It shows the "Setup" section with a "Formatter type" dropdown set to "Custom Javascript formatter". Below this is a "Formatter code" text area containing JavaScript code for processing acknowledgment messages. The code defines a function "Acknowledge_proc" that takes "bytes" as input and returns an "Acknowledge_Msg" object. The code includes a switch statement to handle different message types (0x0 for success, 0x1 for failure, and default for break). The "msgid" is derived from the first byte of the input. At the bottom, there is a "Paste application formatter" button and a "Save changes" button.

```

507
508 //Message type: Acknowledge 0x0F
509 function Acknowledge_proc(bytes) {
510     var Acknowledge_Msg = {
511         "type": 0,
512         "result": 0,
513         "msgid": 0,
514     };
515     Acknowledge_Msg.type = "Acknowledgment";
516     Acknowledge_Msg.result = (bytes[0] & 0x0F);
517     switch (Acknowledge_Msg.result) {
518         case 0x0:
519             Acknowledge_Msg.result = "downlink success";
520             break;
521         case 0x1:
522             Acknowledge_Msg.result = "downlink failure";
523             break;
524         default:
525             break;
526     }
527     Acknowledge_Msg.msgid = bytes[1].toString(16).toUpperCase();
528     return Acknowledge_Msg;
529 }
530
531
532

```

Note: You can also enter **the device's uplink message** in the Byte payload in the Test area on the right to test the decoder parsing results.

**tracker-70b3d5a4d2100833**
ID: tracker-70b3d5a4d2100833

↑ 192 ↓ 98 • Last activity Sep 4, 2023

OverviewLive dataMessagingLocationPayload formattersGeneral settings

UplinkDownlink

Setup

Formatter type*
Custom Javascript formatter

Formatter code*

```
588 //Message type: Acknowledge 0x0F
589 function Acknowledge_proc(bytes) {
590     var Acknowledge_Msg = {
591         "type": 0,
592         "result": 0,
593         "msgid": 0,
594     };
595     Acknowledge_Msg.type = "Acknowledgment";
596     Acknowledge_Msg.result = (bytes[0] & 0x0F);
597     switch (Acknowledge_Msg.result) {
598         case 0x0:
599             Acknowledge_Msg.result = "downlink success";
600             break;
601         case 0x1:
602             Acknowledge_Msg.result = "downlink failure";
603             break;
604         default:
605             break;
606     }
607     Acknowledge_Msg.msgid = bytes[1].toString(16).toUpperCase();
608     return Acknowledge_Msg;
609 }
```

Paste application formatter

Save changes

Test

Byte payload16 20 A0 30 00 3C 78 08 06FPort1Test decoder

Decoded test payload

```
{
  "bytes": {
    "BLEReceivingDuration": "11sec",
    "ContinuousReceive": "Continuous Bluetooth receiving is disabled",
    "LoRaTxpower": "26dBm",
    "LoRaWANband": "EU868",
    "adr": "OFF",
    "dr": "DR3",
    "hb": "3600sec",
    "mode": "EU868",
    "pos": "300sec",
    "type": "Register"
  }
}
```

Complete uplink data

```
{
  "f_port": 1,
  "frm_payload": "FicgMAA8eAsG",
  "decoded_payload": {
    "bytes": {
      "BLEReceivingDuration": "11sec",
      "ContinuousReceive": "Continuous Bluetooth receiving is disabled",
      "LoRaTxpower": "26dBm",
      "LoRaWANband": "EU868",
      "adr": "OFF",
      "dr": "DR3",
    }
  }
}
```

✔ Payload is valid
[Learn more about payload formatters](#)

End of document