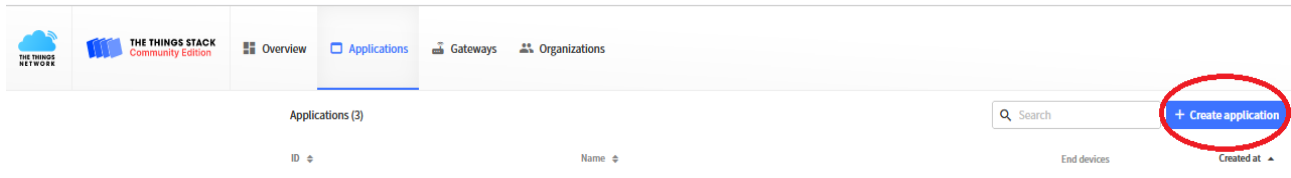


Onboarding a Light Tracker with TTN

Create TTN account

Create a Application



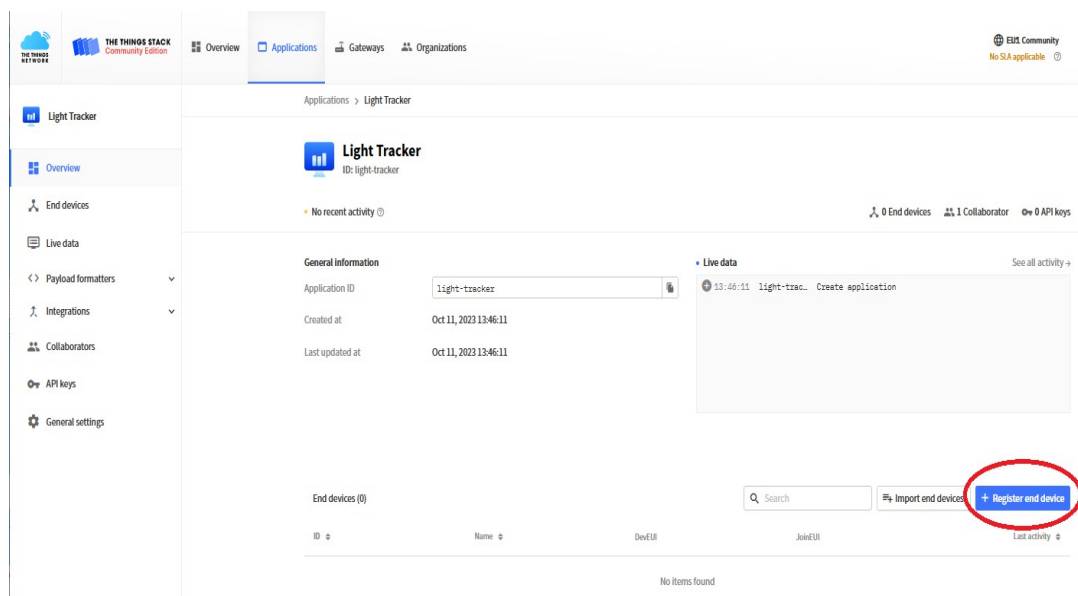
and give it a name

The screenshot shows the 'Create application' form. The form has the following fields:


- Application ID ***: light-tracker
- Application name**: Light Tracker
- Description**: ORPLabs light tracker


Below the description field, there is a note: 'Optional application description; can also be used to save notes about the application'. At the bottom of the form, there is a '+ Create application' button, which is circled in red.

And register a new device:



the Light tracker is a LoraWAN 1.0.4 device and you will need to set up a initial Frequency Plan.

 THE THINGS NETWORK

 THE THINGS STACK
Community Edition

Overview

Applications

Gateways

Organizations

Light Tracker

Overview

End devices

Live data

Payload formatters

Integrations

Collaborators

API keys

General settings

Applications > Light Tracker > End devices

Register end device

Does your end device have a LoRaWAN® Device Identification QR Code? Scan it to speed up onboarding.

Scan end device QR code

Device registration help

End device type

Input method

Select the end device in the LoRaWAN Device Repository

Enter end device specifics manually

Frequency plan

Europe 863-870 MHz (SF9 for RX2 - recommended)

LoRaWAN version

LoRaWAN Specification 1.0.4

Regional Parameters version

RP002 Regional Parameters 1.0.4

[Show advanced activation, LoRaWAN class and cluster settings](#)

Provisioning information

JoinEUI

.. .. .

Confirm

To continue, please enter the JoinEUI of the end device so we can determine onboarding options

Click on Show advanced activation, LoRaWAN class and cluster settings and select Over the Air Activation and Class A LoraWAN

[Show advanced activation, LoRaWAN class and cluster settings](#)

Activation mode

- Over the air activation (OTAA)

Activation by personalization (ABP)

Define multicast group (ABP & Multicast)

Additional LoRaWAN class capabilities

None (class A only)

Network defaults

- Use network's default MAC settings

Cluster settings

- Skip registration on Join Server

JoinEUI for the light tracker is 6081F9B609A7673F and generate the Device EUI and Appkey:

Provisioning information

JoinEUI ⓘ *

50 81 F9 B6 09 A7 67 3F

Reset

This end device can be registered on the network

DevEUI ⓘ *

70 B3 D5 7E D0 06 1B 1F

Generate

1/50 used

AppKey ⓘ *

1A 3D B6 01 17 FF 57 14 A3 3C C2 65 52 B6 4A 88

Generate

End device ID ⓘ *

eui-70b3d57ed0061b1b

This value is automatically prefilled using the DevEUI

After registration

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

Register end device

and register the device.

Use the generated keys in the Light Tracker software, compile and program:

```
//***** UPDATE HERE WITH YOUR DEVICE KEYS *****/
//You should copy device keys from Helium or TTN Console and update following keys. Please check out:
https://github.com/lightaprs/LightTracker-1.0/wiki/Adding-Device-on-Helium-Console

// This EUI must be in little-endian format, so least-significant-byte (lsb)
// first. When copying an EUI from Helium Console or ttnctl output, this means to reverse the bytes.
static const ul_t PROGMEM DEVEUI[8]={0x1B, 0x1B, 0x06, 0xD0, 0x7E, 0xD5, 0xB3, 0x70}; //ttn

// This EUI must be in little-endian format, so least-significant-byte (lsb)
// first. When copying an EUI from Helium Console or ttnctl output, this means to reverse the bytes.
static const ul_t PROGMEM APPEUI[8]={0x3F, 0x67, 0xA7, 0x09, 0xB6, 0xF9, 0x81, 0x60}; //ttn

// This key should be in big endian format (or, since it is not really a
// number but a block of memory, endianness does not really apply). In practice, a key taken from Helium Console or ttnctl can be
// copied as-is.

static const ul_t PROGMEM APPKEY[16] = {0x1A, 0x3D, 0xB6, 0x01, 0x17, 0xFF, 0x57, 0x14, 0xA3, 0x3C, 0xC2, 0x65, 0x52, 0xB6, 0xA4,
0x88}; //ttn
```