

Kuando Busylight LoRa and ChirpStack: Getting started

Table of Contents

- Registering your Gateway..... 1
- Registering your Busylight 3
- Testing the Busylight 9
- Controlling the Busylight with http Requests 11

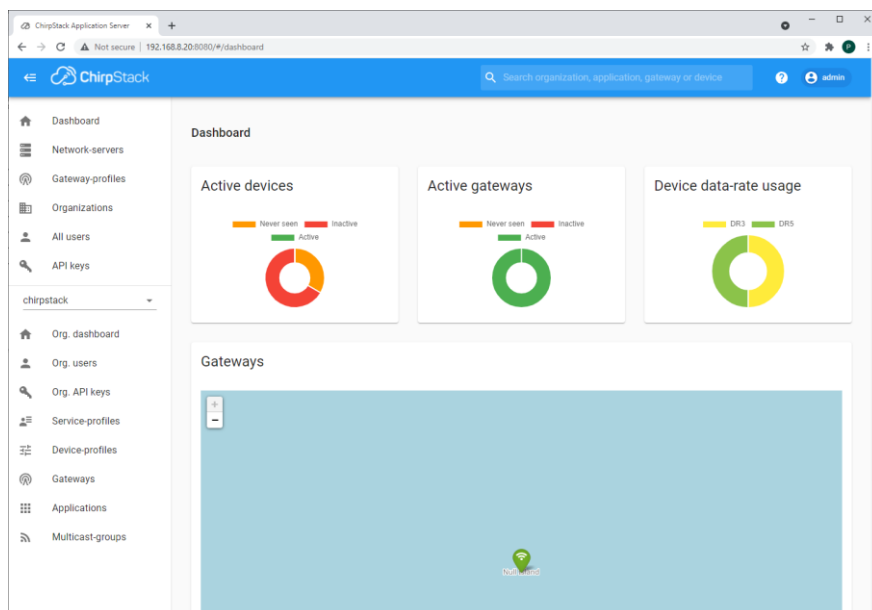
ChirpStack allows the addressing of the Busylight LoRa devices with https – requests.

We will describe how to send commands to your LoRa Busylight in various ways.

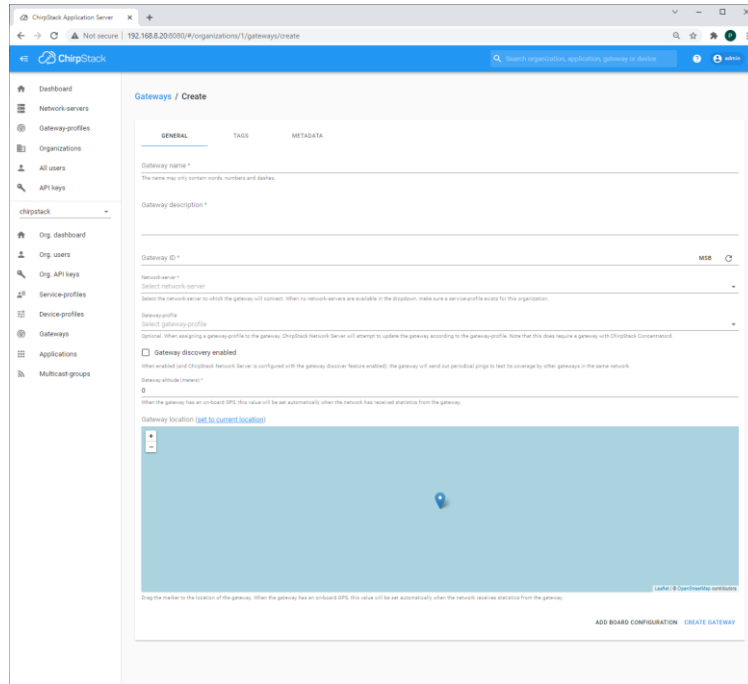
Registering your Gateway

First log in into your Chipstack Admin GUI.

If you have a Gateway with built-in Network Server, the Gateway may already be registered.



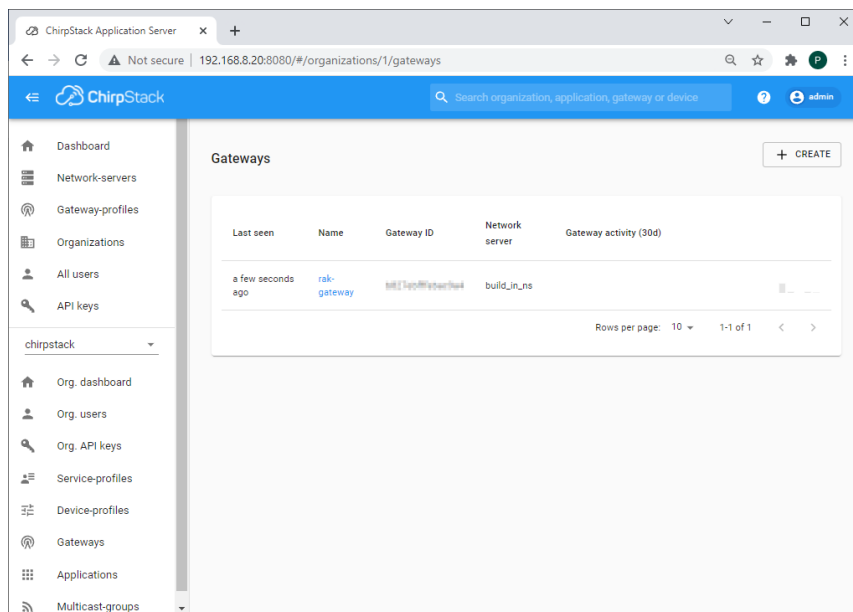
Click on Gateways / “+ Create”:



The Gateway Name and the description can be set as you would like defined.

The Gateway ID must match the ID of your gateway. In most cases, it is the MAC Address of the LoRa radio.

If everything is OK, you will see a recent 'Last seen' entry in the Gateway list.

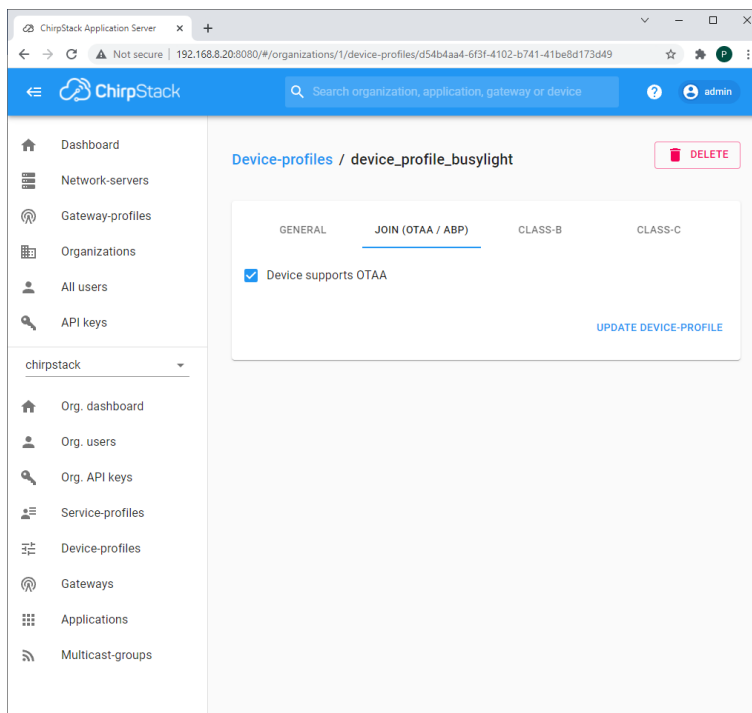
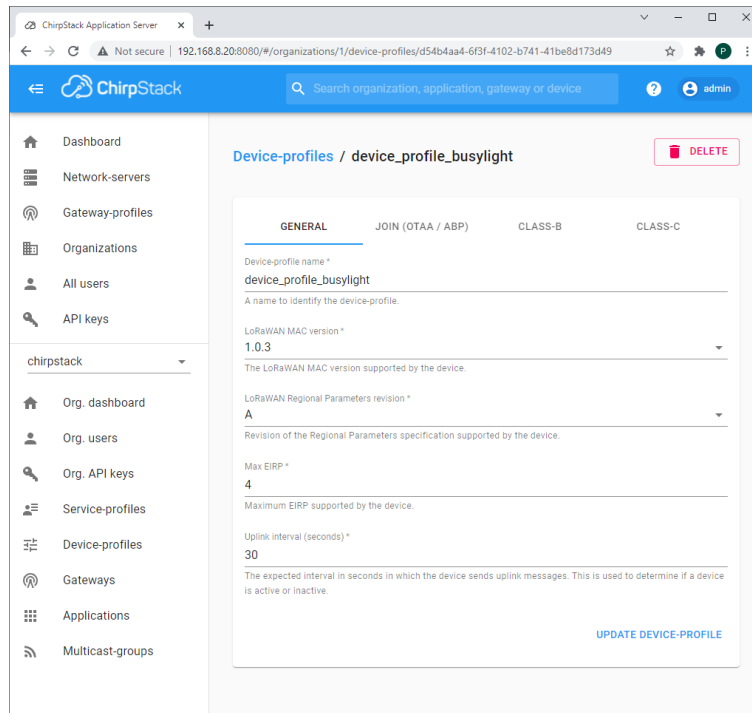


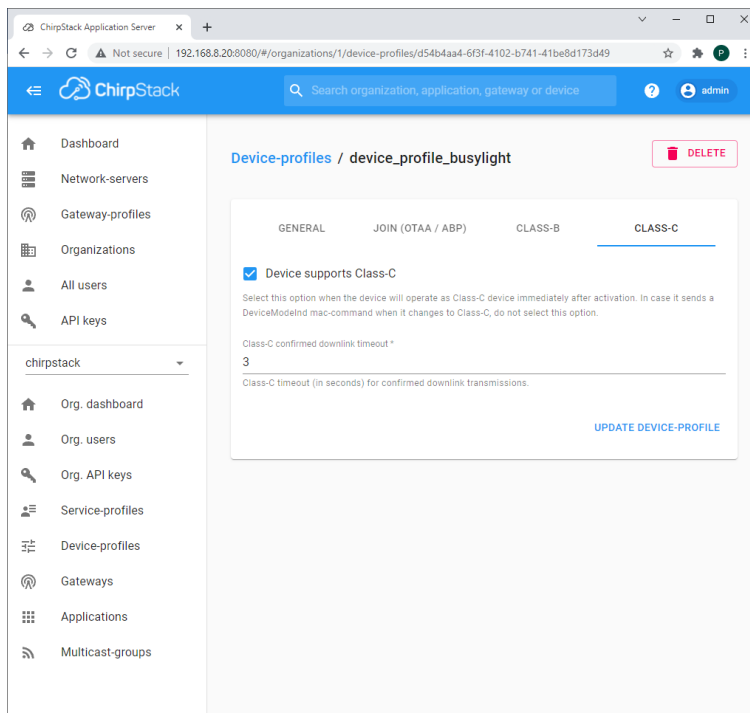
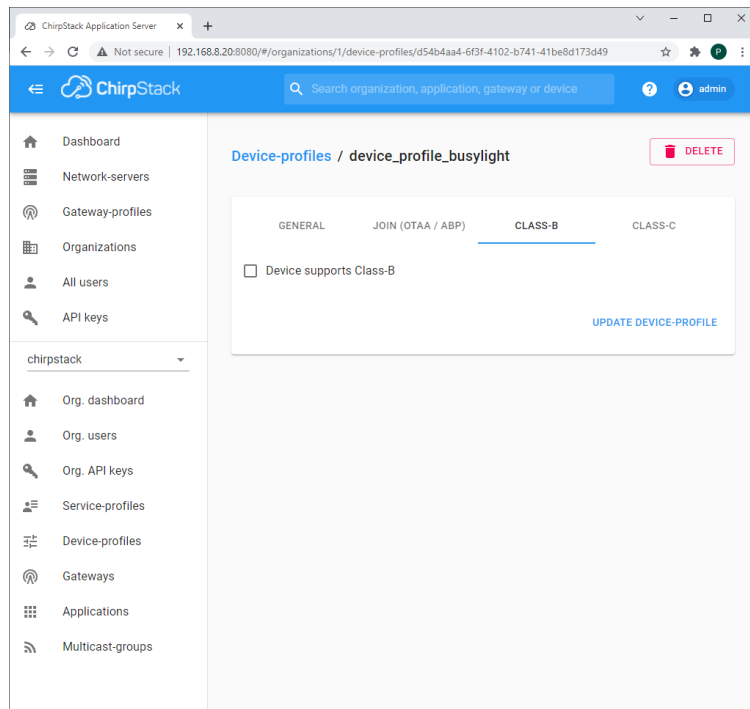
Registering your Busylight

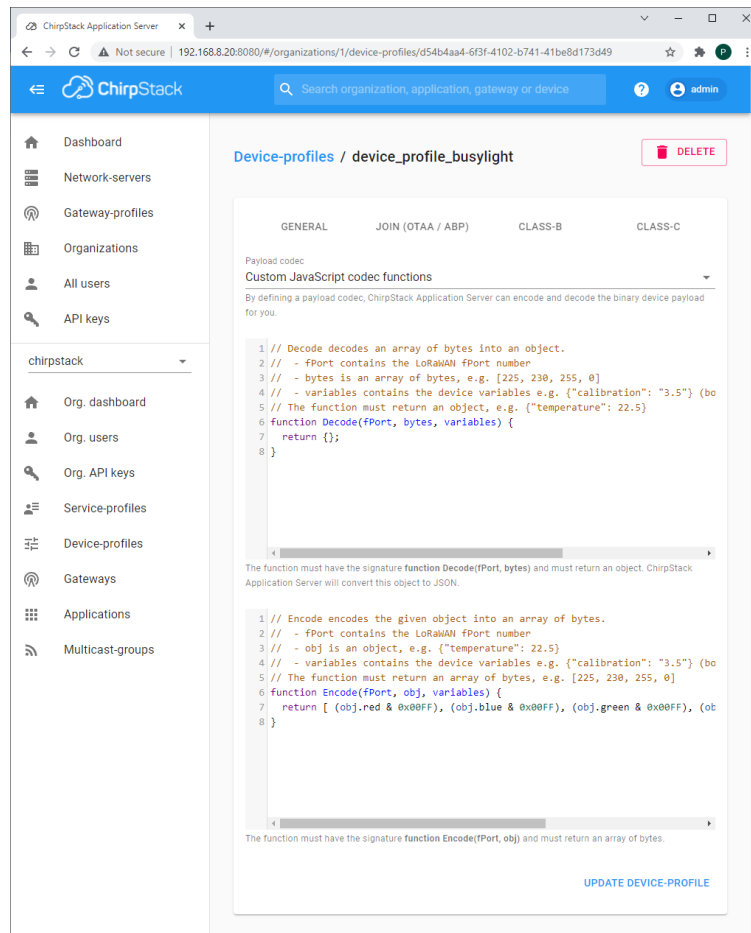
First you need to create a device profile. You need to create a device profile once and can use it for all of your Busylight devices.

In the ChirpStack Console, click on Device Profiles, then “+ Create”.

You can copy the entries from these screen shots:







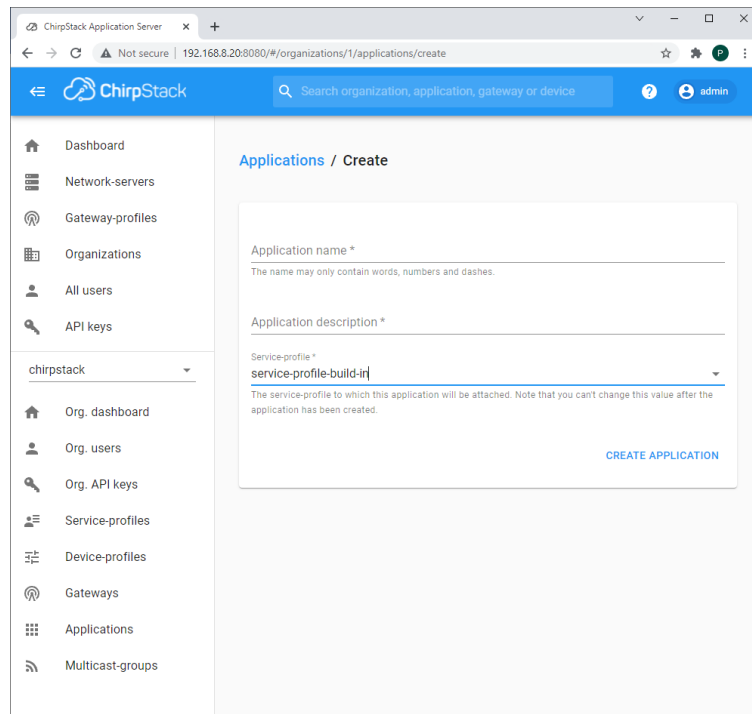
You need to enter a Payload Encoder to let ChirpStack code the hardware payload bytes.

Here is the Encoder to copy into your own ChirpStack instance:

```
// Encode encodes the given object into an array of bytes.
// - fPort contains the LoRaWAN fPort number
// - obj is an object, e.g. {"temperature": 22.5}
// - variables contains the device variables e.g. {"calibration": "3.5"} (both the
key / value are of type string)
// The function must return an array of bytes, e.g. [225, 230, 255, 0]
function Encode(fPort, obj, variables) {
    return [ (obj.red & 0x00FF), (obj.blue & 0x00FF), (obj.green & 0x00FF),
(obj.ontime & 0x00FF), (obj.offtime & 0x00FF) ];
}
```

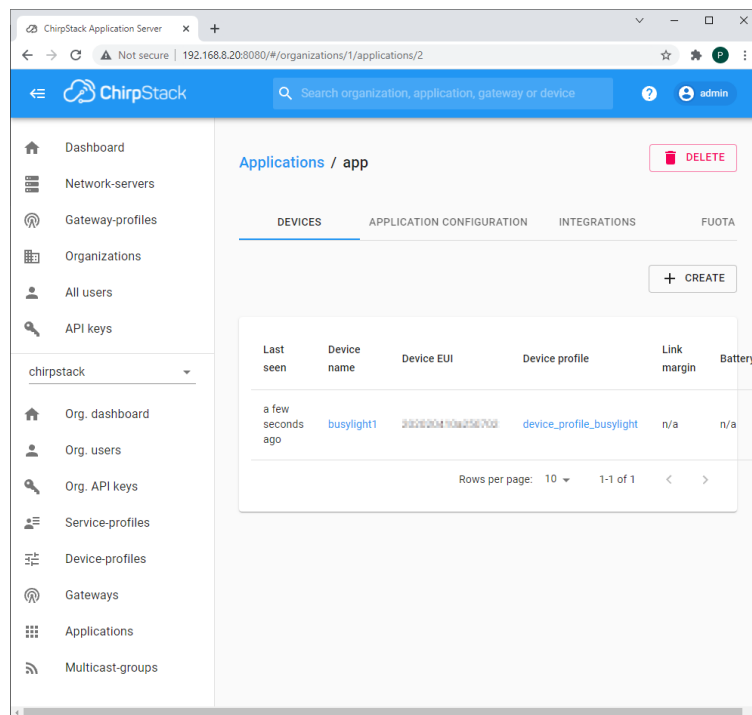
Next, you need to create a ChirpStack Application. Please Click on Applications and then to “+ Create” in the ChirpStack Console.

You can choose Application Name and Application Description as you would like it defined.



After Creation, please enter the new Application by clicking on your Application name.

You will see it in the device list.

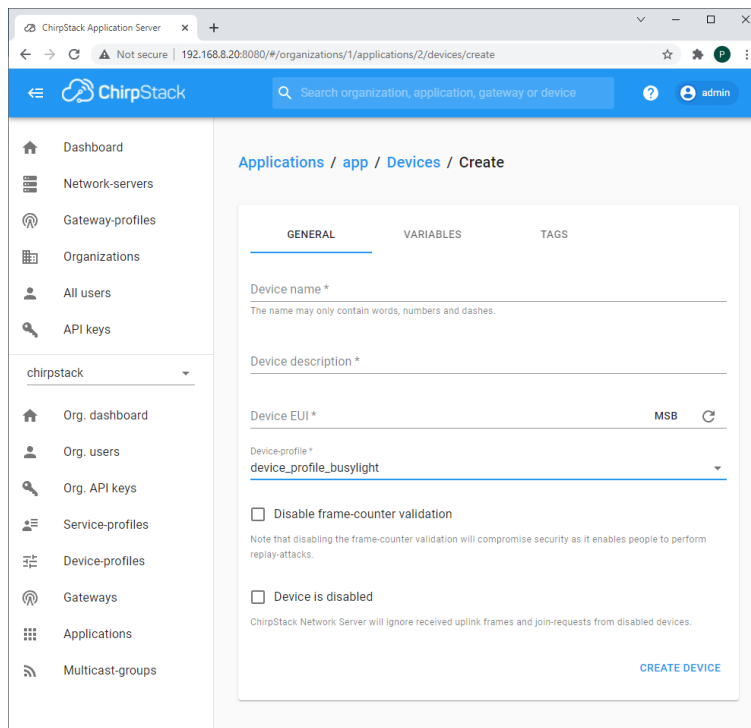


To Add a Busylight Device, click on “+ Create”.

For registering your Busylight device, you need this information:

- DeviceEUI (8 Byte Hex)
- AppEUI (8 Byte Hex)
- AppKey (16 Byte Hex)

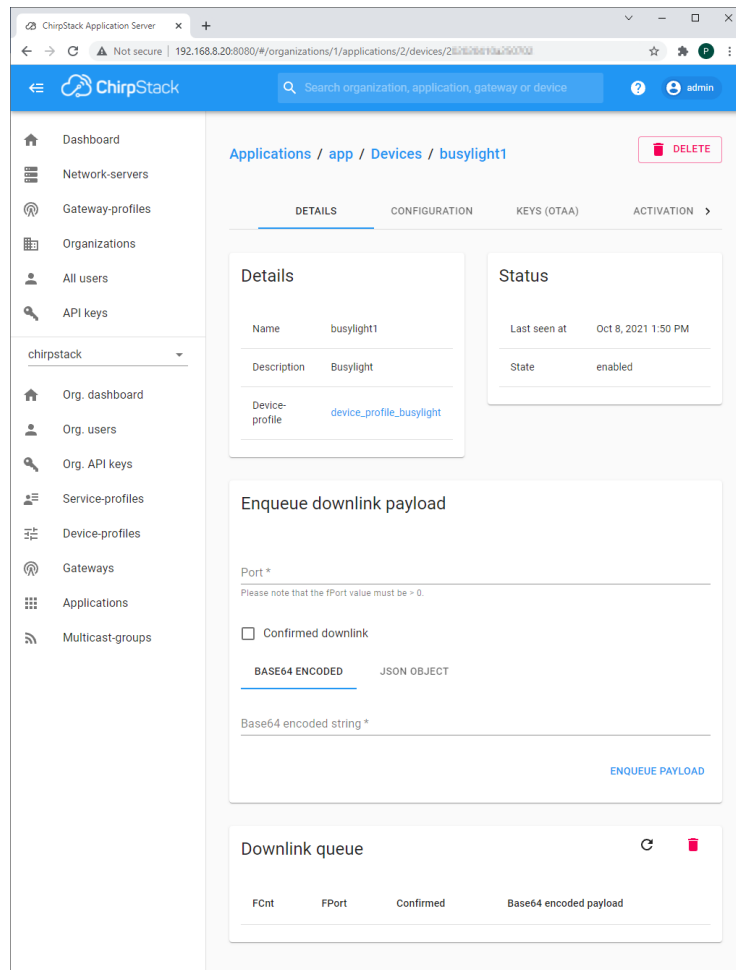
First, you need to create the device itself:



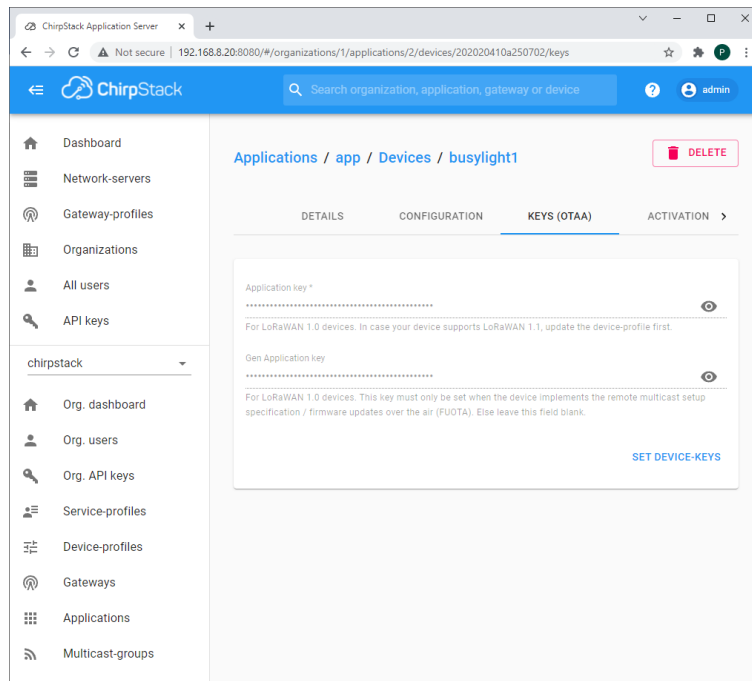
You can use appropriate entries for Device name and Device description here.

Please enter the DeviceEUI of your Busylight and select the device profile you have created earlier.

After creation, enter the device details by clicking on the device name:

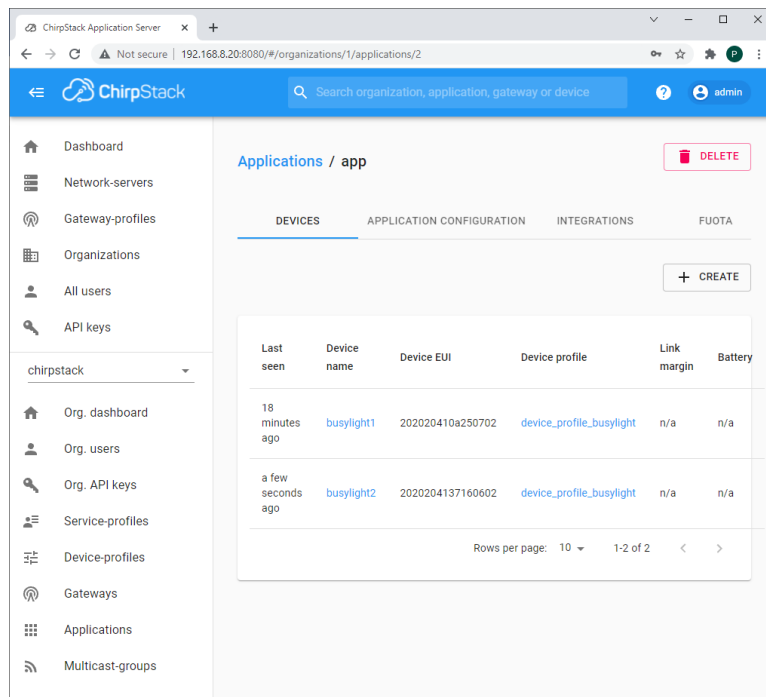


Please click on KEYS (OTAA).



Please click on "SET DEVICE KEYS" and enter your devices' APPKey.

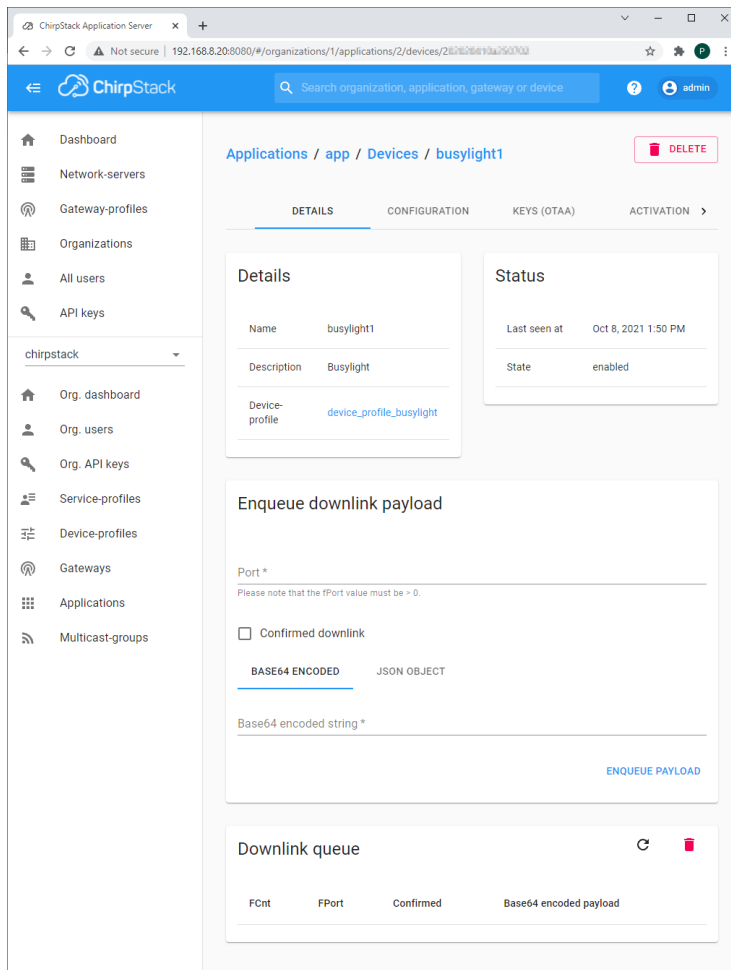
You can now power up your Busylight, it will go to green light and you can see the status in the ChirpStack Console as well:



Testing the Busylight

You can test your connected Busylight by sending the downlink payload from the ChirpStack console.

Please open the details page for your device.



The Port needs to be set to 15. They Payload needs to be specified in Base64 or JSON notation.

For a first test, enter this string into the Base64 payload:

AP8A/wA=

In this case, the Busylight will be solid blue.

Please have a look into the chapter about the hardware payload format.

You can use this JSON object for testing:

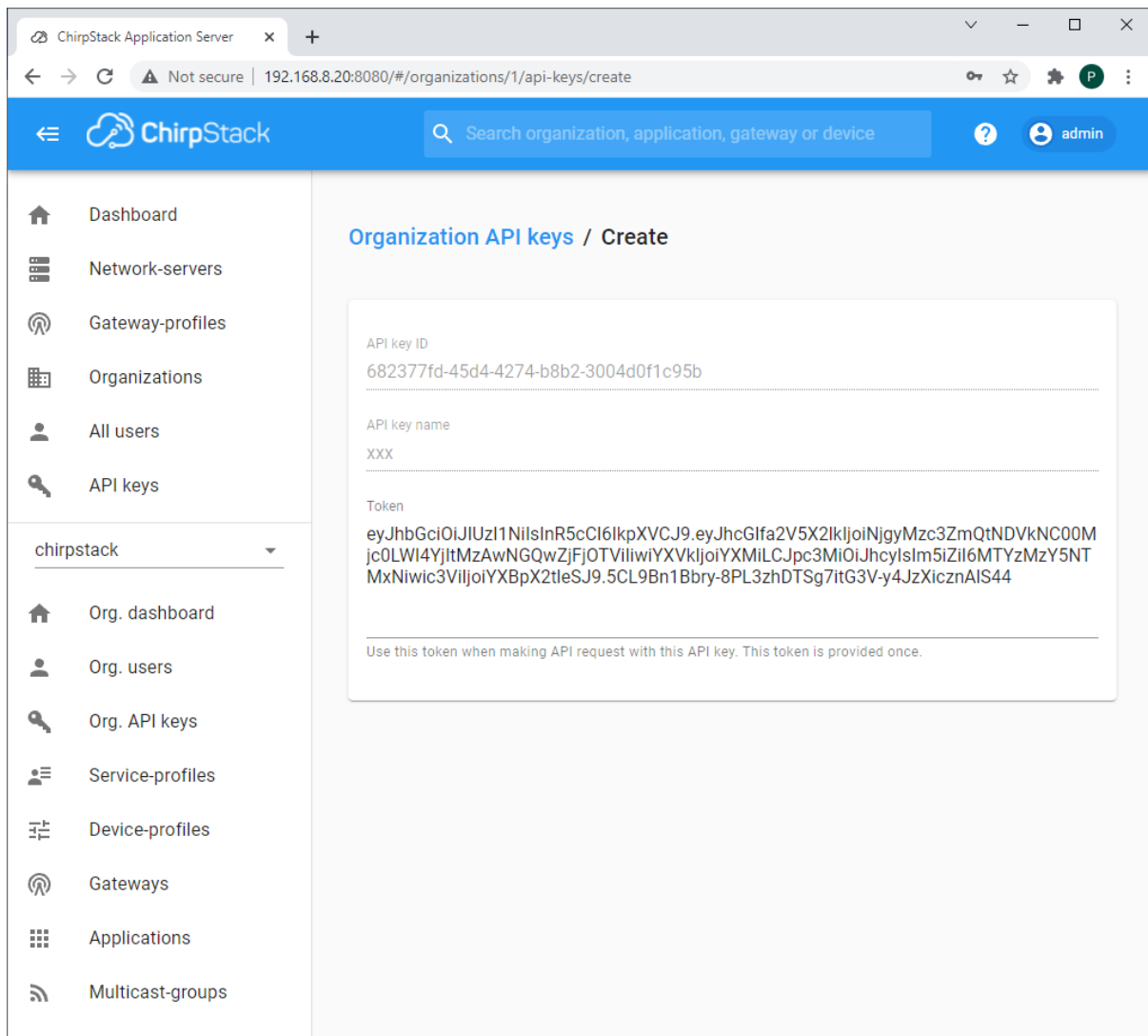
```
{ "red":255, "blue":255,"green":0,"ontime":255,"offtime":0 }
```

In this case, the Busylight will be solid purple.

Controlling the Busylight with http Requests

If you plan to control the Busylight with http/https requests, you need to create an API Key.

To create a key, open the application and click on “Org. API Keys”.



Please copy the token now - you will need it for your http requests.

You are now ready to write control your Busylight with a HTML request. Please change the yellow highlighted text parts to your to match your setup.

You need to send a POST request to this URI:

https://your_chirpstack_server:8080/api/devices/your_device_eu1/queue

You send this Body:

```
{
  "deviceQueueItem": {
    "confirmed": true,
    "jsonObject": "{ \"red\":255,
    \"blue\":255,\"green\":0,\"ontime\":255,\"offtime\":0 }",
    "fPort": 15
  }
}
```

And you need to send these Headers:

