# Manage your own TTN application that communicates with Beep.nl

In this document we explain how to save data from Lora powered devices into the Beep App using The Things Network.

1. Login you TTN console
2. Go to applications
3. Click Add application
   1. Name your application
   2. Select the handler of you application according to you location

If you don’t have already registered a device with TTN go to <https://www.thethingsnetwork.org/docs/devices/registration.html> for an explanation how to add an device to their Lora network.

## Integrations:

In TTN you can export your data to different applications using something called *integrations.* The Beep App uses the HTTP integration of TTN*.* In this section we explain how to setup this integration and how to configure it.

1. Go to the *integrations* tab in the application section of the TTN console.
2. Click *add integration*
3. Select the *HTTP integration*
4. Enter the following details:

|  |  |
| --- | --- |
| Process ID | your preferred name |
| Acces key | devices messages |
| URL | https://api.beep.nl/api/lora\_sensors |
| Method | POST |
| Authorization | Leave empty |
| Custom Header Name | leave empty |
| Custom Header Value | leave empty |

1. Click add integration

## Payload formats:

Bandwidth is scarce in Lora, Hence all devices must send their data in binary format. In the tab *Payload formats* we convert the binary format into something Beep can understand. For this we need to provide TTN with a JavaScript function. For example the following script turns the binary data into a ASCII string:

function Decoder(bytes, port) {

// Decode plain text; for testing only

return {

myTestValue: String.fromCharCode.apply(null, bytes)

};

}

For Beep we require that the decoder returns a JSON object for which the parameters corresponds to the sensor values. For example, the object

{

t\_i: 25,

t: 12,

bv: 5,

w\_v: 30,

key: “123456”

}

Corresponds to 25 degrees inside, 12 degrees outside, 5V battery voltage and a weight of 30 Kg. The *key* parameter corresponds to the *Sensor code* in the Beep App.If you don’t define a key, the Device EUI will be used as key instead.

For more information which sensor values you can use, see the Beep HTTP Post API documentation.  
If you build a Beep base you can copy our payload decoder and converter code from our Github repository:

1. Go to the payload format tab in the application
2. Go to <https://github.com/beepnl/BEEP> and download or copy [ttn-payload-converter.js](https://github.com/beepnl/BEEP/blob/master/ttn-payload-converter.js" \o "ttn-payload-converter.js) and [ttn-payload-decoder.js](https://github.com/beepnl/BEEP/blob/master/ttn-payload-decoder.js" \o "ttn-payload-decoder.js)
3. Copy and paste text out of [ttn-payload-decoder.js](https://github.com/beepnl/BEEP/blob/master/ttn-payload-decoder.js" \o "ttn-payload-decoder.js) in the decoder field
4. Copy and paste text out of [ttn-payload-converter.js](https://github.com/beepnl/BEEP/blob/master/ttn-payload-converter.js" \o "ttn-payload-converter.js) in the converter field
5. The validator and encoder fields can be left empty.

Click: Save payload functions