

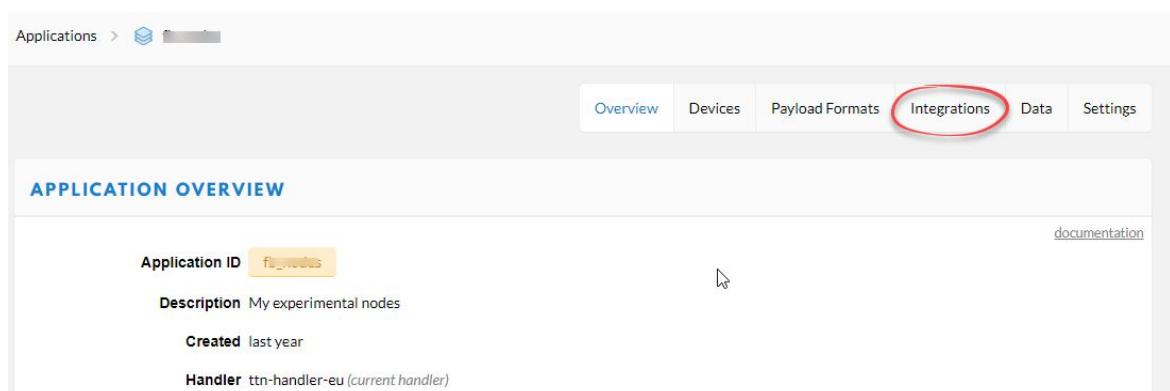
Using the data of your Node

Building a TTN node was part of our previous workshop. We will use the `ttn_bmp280_abp` example, posting temperature and humidity to TTN. There are multiple ways to use your values in an application. This will depend on your usage. First we cover the standard integrations offered by TTN, afterwards we come to the IoT building blocks of Node Red, hosted on IBM Cloud platform

This work is licensed under the Creative Commons Attribution-NonCommercial 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

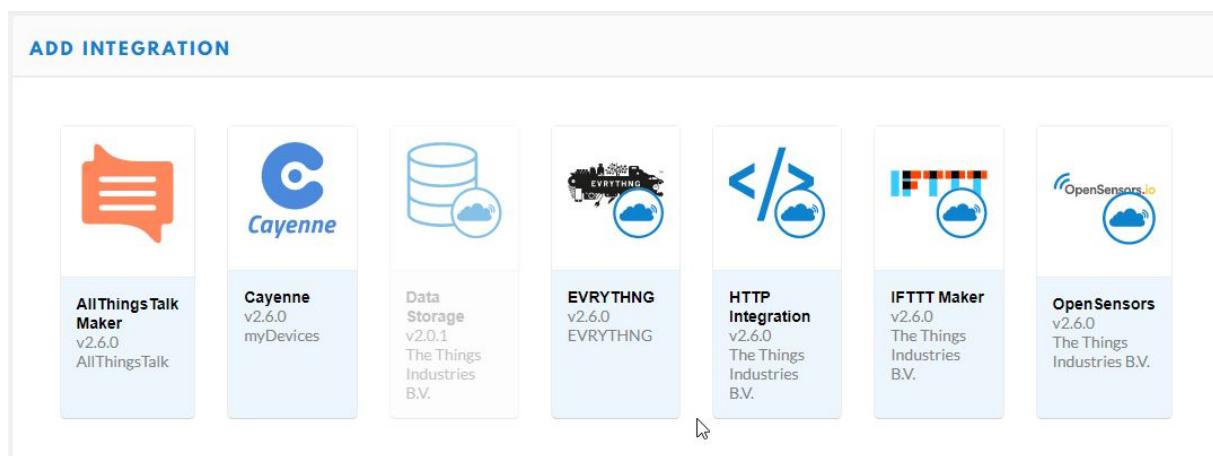
Integrations

The Things Network offers some integrations 'out-of-the-box'. They are connected on 'application' level, so every application can have its own individual integration.



A screenshot of the TTN Application Overview page. At the top, there is a navigation bar with tabs: Overview, Devices, Payload Formats, **Integrations**, Data, and Settings. The 'Integrations' tab is highlighted with a red oval. Below the navigation bar, the page title is 'APPLICATION OVERVIEW'. Underneath, there is a section for the application with fields: Application ID (ttn-test), Description (My experimental nodes), Created (last year), and Handler (ttn-handler-eu). A 'documentation' link is located in the top right corner.

At this moment these are offered:



A screenshot of the 'ADD INTEGRATION' page. It displays seven integration options in a grid:

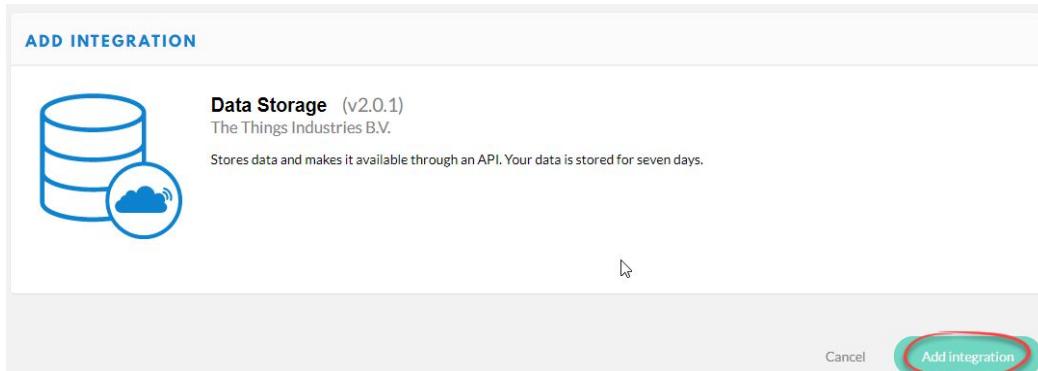
- AllThingsTalk Maker v2.6.0 AllThingsTalk
- Cayenne v2.6.0 myDevices
- Data Storage v2.0.1 The Things Industries B.V.
- EVRYTHNG v2.6.0 EVRYTHNG
- HTTP Integration v2.6.0 The Things Industries B.V.
- IFTTT Maker v2.6.0 The Things Industries B.V.
- OpenSensors v2.6.0 The Things Industries B.V.

AllThingsTalk -	AllThingsTalk	http://docs.allthingstalk.com/network
-----------------	---------------	---

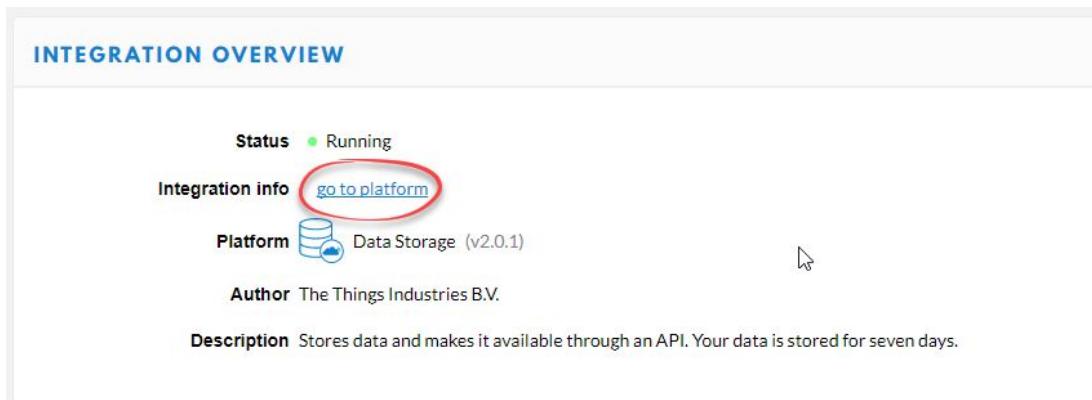
Maker		https://use-the-things-network/
Cayenne	Cayenne myDevices platform	https://www.thethingsnetwork.org/abs/story/tutorial-microchip-lora-module-and-the-things-network-cayenne-integration
Data Storage	Store 7 days of values	
Evrythng	Using THNGHUB as a IoT platform	
HTTP Integration	Post HTTP to another application	
IFTTT Maker	If this, than that, action oriented service	https://www.thethingsnetwork.org/forum/t/ifttt-integration/5332
OpenSensors	Workspace allocation and usage	

Data Storage

Normally the data of your sensors is not stored. So you have to put it in a database if you want to save your history. TTN offers a free storage option for seven days. You add it by choosing the Data Storage integration:



Choose Add Integration to install.



Choose 'go to platform' to go to the platform:

The Things Network Data Storage

Stores data and makes it available using a REST API

Created by The Things Industries B.V.
See more at <https://www.thethingsindustries.com>
[Contact the developer](#)

devices

GET /api/v2/devices Query the devices for which data has been stored

query

GET /api/v2/query Query data

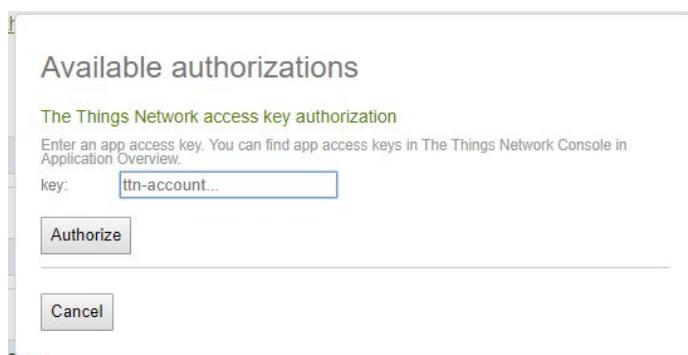
query/{device-id}

GET /api/v2/query/{device-id} Query data for a specific device

[BASE URL: , API VERSION: 2.0.0]

VALID { }

To access your data, you have to authorize the platform. Choose 'Authorize' and fill in your application access key.



Now you can query your data, select the period from which you want to see the data (1d = 1 day, 1h = 1 hour):

Parameter	Value	Description	Parameter Type	Data Type
last	1d	Duration on which we want to get the data (default 1h). Pass 30s for the last 30 seconds, 1h for the last hour, 2d for the last 48 hours, etc	query	string

Try it out! [Hide Response](#)

Curl

```
curl -X GET --header 'Accept: application/json' --header 'Authorization: key ttn-account-v2.0AhmhCIIF'
```

Request URL

<https://s.data.thethingsnetwork.org/api/v2/query?last=1d>

You can use the URLs and code in your own application.

IFTTT

IFTTT covers a lot of ‘If this than that’ actions. It can be integrated in The Things Network so changes in any value could trigger some other actions (sending an email, sent a tweet).

ADD INTEGRATION

IFTTT Maker (v2.6.0)
The Things Industries B.V.



Sends uplink messages to your IFTTT Maker Channel. If you want to trigger this integration based on sensor values, use the 'trigger' payload field (true or false) as a result of your decoder or converter payload function. Use the special fields 'dev_id' for device ID, 'app_id' for application ID and 'hardware_serial' for the DevEUI.

[documentation](#)

Process ID
The unique identifier of the new integration process

Event Name
The event name of your IFTTT recipe

Key
Your key

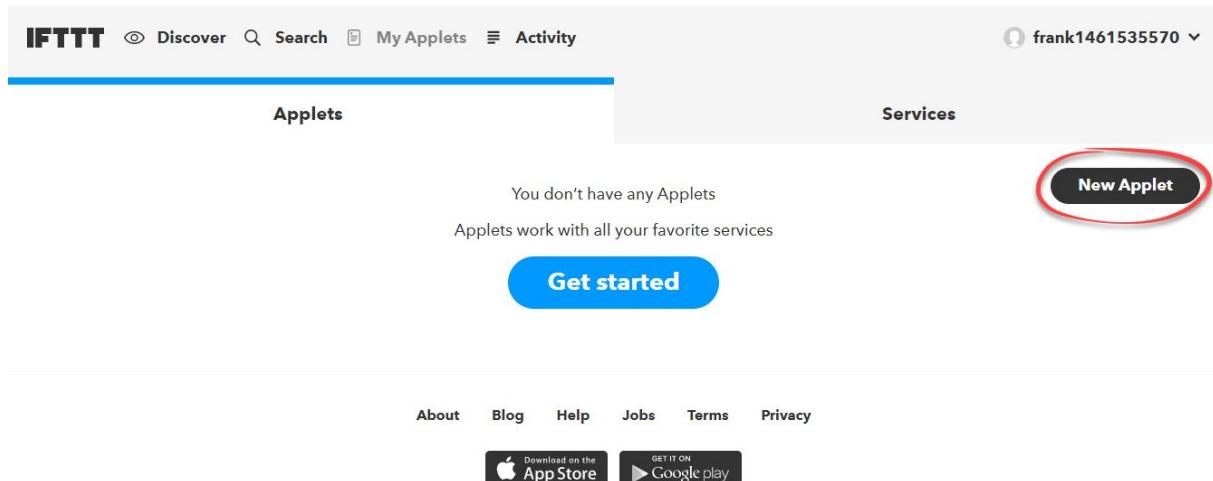
Value 1
Payload field name to send as value 1

Value 2
Payload field name to send as value 2

Value 3
Payload field name to send as value 3

Process ID can be the name of your integration ('ifttt-temperature'), Event name the name of your event ('temperature').

In IFTTT create an account.



The screenshot shows the IFTTT website interface. At the top, there's a navigation bar with 'Discover', 'Search', 'My Applets', and 'Activity'. On the right, it shows a user profile 'frank1461535570'. Below the navigation, there are two tabs: 'Applets' (selected) and 'Services'. A message says 'You don't have any Applets' and 'Applets work with all your favorite services'. A prominent blue button labeled 'Get started' is centered. To the right, a black button labeled 'New Applet' is circled in red. At the bottom, there are links for 'About', 'Blog', 'Help', 'Jobs', 'Terms', and 'Privacy', along with download links for the 'App Store' and 'Google play'.

Create a new Applet by choosing ‘New Applet’

New Applet

if **+this** then that

It will be the ‘+this’ part, so click on it.

Choose a service

Step 1 of 6

Q webhook



Look for the ‘webhook’ service and select it.

Choose a trigger gives only one possibility: ‘Receive a web request’ , so choose that one.



Complete trigger fields

Step 2 of 6

Receive a web request

This trigger fires every time the Maker service receives a web request to notify it of an event. For information on triggering events, go to your Maker service settings and then the listed URL (web) or tap your username (mobile)

Event Name

temperature

The name of the event, like "button_pressed" or "front_door_opened"

Create trigger

Fill in the name of your Event as defined previous in the TTN console ('temperature') and 'create Trigger'.



Next the '+that' part, what to do when the webhook is triggered.

We will add the values to a Google Docs Spreadsheet:

Choose action service

Step 3 of 6

Q google sheets



Choose 'Google Sheets'

« Back



Choose action

Step 4 of 6

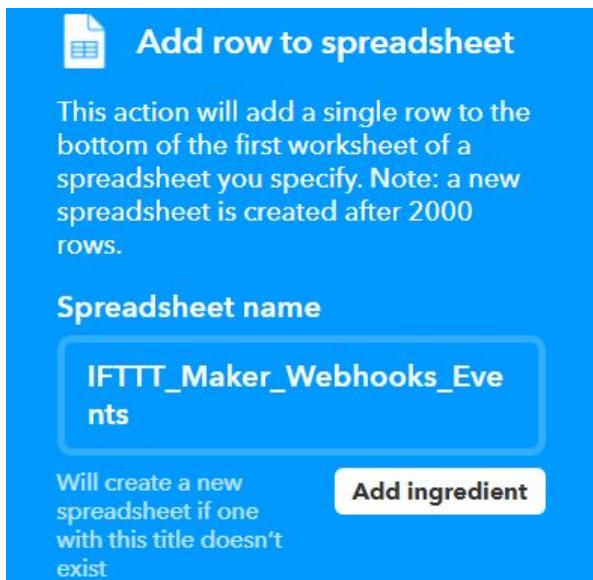
Add row to spreadsheet

This action will add a single row to the bottom of the first worksheet of a spreadsheet you specify. Note: a new spreadsheet is created after 2000 rows.

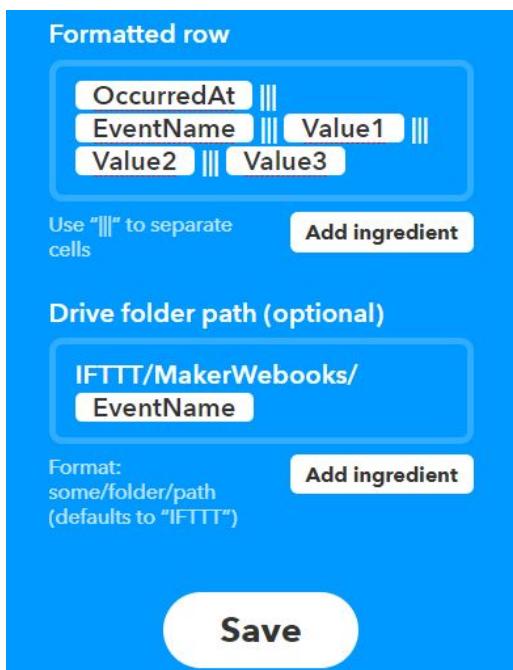
Update cell in spreadsheet

This action will update a single cell in the first worksheet of a spreadsheet you specify. Note: a new spreadsheet is created if the file doesn't exist.

Choose 'Add row to spreadsheet'



You can give your spreadsheet a name.

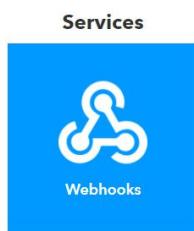


The Formatted row describes the data fields added to your Sheet. The values are defined in the TTN console. You can dive a Drive Folder Path where the file is stored. It will be created if it does not exist.

Save your settings. You have to fill in your Google account credentials to connect and give IFTTT access to your Google Drive.

Now look for your webhook service:

 webhook



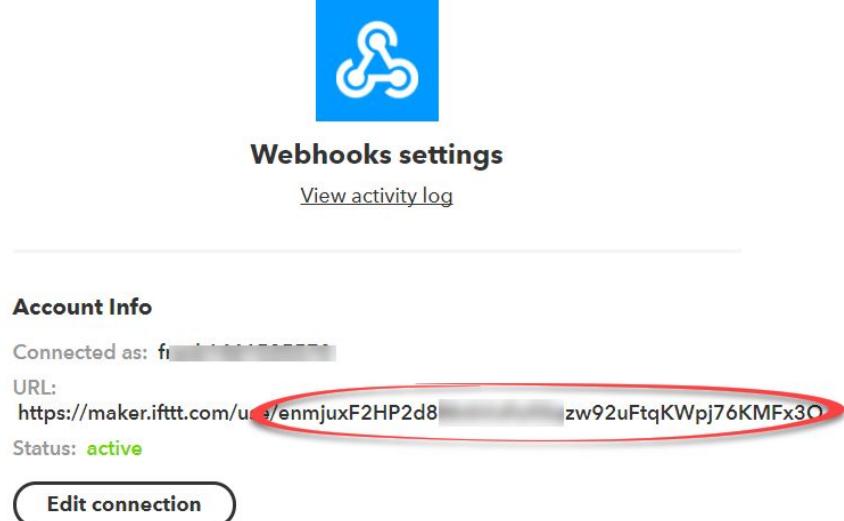
Select 'Webhooks':



The screenshot shows the IFTTT Webhooks settings page. At the top, there's a navigation bar with 'Documentation' and a 'Settings' button, which is highlighted with a red circle. Below the navigation is the IFTTT logo and the word 'Webhooks'. A descriptive text block says: 'Integrate other services on IFTTT with your DIY projects. You can create Applets that work with any device or app that can make or receive a web request. If you'd like to build your own service and Applets, check out the [IFTTT platform](#).'. There are also 'View activity log' and 'Edit connection' buttons.

Select settings to get to the Webhook settings.

My Applets > Webhooks



This screenshot shows the 'Webhooks settings' page. It features the IFTTT logo and the title 'Webhooks settings'. Below it is a 'View activity log' link. The 'Account Info' section includes: 'Connected as: [REDACTED]', 'URL: https://maker.ifttt.com/use/enmjuxF2HP2d8[REDACTED]zw92uFtqKWPj76KMFx3O' (with the URL part circled in red), and 'Status: active'. At the bottom is an 'Edit connection' button.

Copy the key after maker.ifttt.com/use/

And put this key in the TTN console:

SETTINGS

Event Name
The event name of your IFTTT recipe

Key
Your key

Value 1
Payload field name to send as value 1

Value 2
Payload field name to send as value 2

Value 3
Payload field name to send as value 3

Add ‘temperature_3’ as value 1 and ‘barometric_pressure_4’ as value 2.

If your node is running, it will post its values to the given worksheet on your Google Drive. To display the values correctly, you have to use UK settings (File, Spreadsheet settings in Google Sheets) for your values (dot as a separator):

57	December 12, 2017 at 01:07AM	temperature	20.1	988
58	December 12, 2017 at 01:08AM	temperature	20.2	988
59	December 12, 2017 at 01:08AM	temperature	20.9	988
60	December 12, 2017 at 01:10AM	temperature	20.4	989
61	December 12, 2017 at 01:11AM	temperature	20.3	989

You can add more than one event on a webhook. To get a message to your phone use the ‘notifications’ applet:



you may change the notification to get the values of your sensors.

Install the IFTTT app to your Android or iOS Phone, add the ‘notifications’ applet and see the data coming in.

← Applet ran
12 december 00:52

The event named "temperature" occurred on the Maker Webhooks service

If maker Event "temperature", th...

Webhooks
Receive a web request ▾
Trigger ran, 12 december 00:52

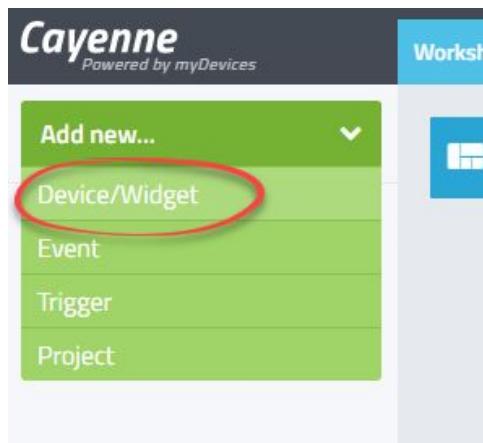
Ingredients

- Value2
988
- Value1
20.1875
- EventName
temperature
- Value3

With this integration you can integrate your connected devices and services to TTN.

Cayenne

Cayenne is a cloud-based IoT platform initiated by MyDevices.io. Go to <https://mydevices.com/> to Sign up for a free account. Login with your newly setup account and Add new 'Device/Widget':



Choose 'LoRa (Beta)'

Devices & Widgets

🔍

DEVICES

- Single Board Computers >
- MicroControllers >
- Sensors ▾
- Actuators ▾
- Extensions ▾
- LoRa (Beta) ▾

Choose 'The Things Network' and then 'Ideetron Nexus Board'.



Fill in the parameters for the Ideetron: DevEUI can be copied from the TTN console:

DEVICE OVERVIEW

Application ID	<input type="text" value="des"/>
Device ID	abp_node
Activation Method	ABP
Device EUI	<input type="text" value="56 29 [REDACTED]"/> [REDACTED]
Application EUI	<input type="text" value="70 B3 I [REDACTED]"/> [REDACTED]

Enter Settings



Ideetron Nexus Board
Development board

This device uses Cayenne LPP

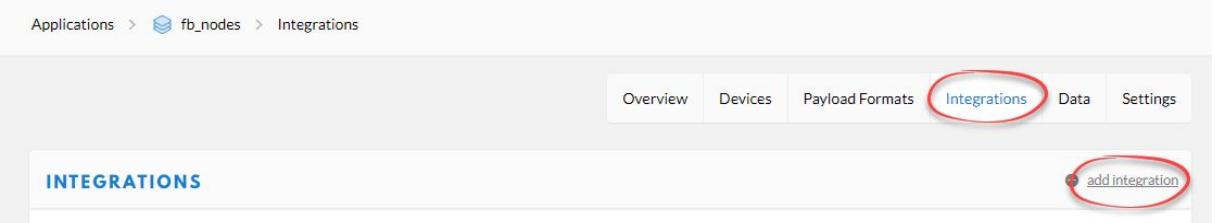
Name	Ideetron Nexus Board
DevEUI	5629
Activation Mode	Already Registered
Location	This device moves

The 'DevEUI' field (5629) is circled in red.

As activation mode choose 'Already Registered' (if using an ABP device) and 'This device moves'.

And finally choose 'Add device'

Go to your application in The Things Network Console, and choose 'Integrations', 'add integration':



Applications > fb_nodes > Integrations

Overview Devices Payload Formats Integrations Data Settings

INTEGRATIONS

add integration

Choose 'Cayenne' in the next screen:



Fill in 'workshop_01' as a unique identifier.

Cayenne

Process ID

The unique identifier of the new integration process

workshop_01

Access Key

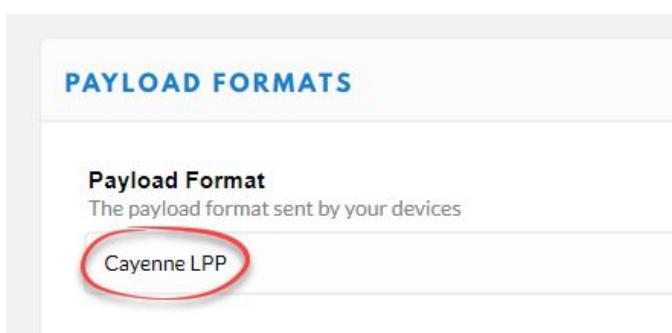
The access key used for downlink

default key

Select 'Default key'.

And finally 'Add integration'.

Now select Application, your application, Payload Format and choose 'Cayenne LPP':



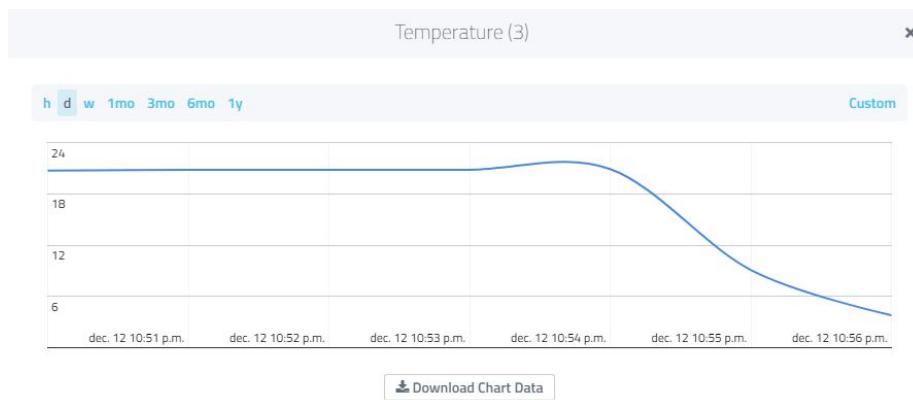
If you look at your data, it will be translated to human readable format:

payload: 03 67 00 D1 05 73 27 8E	barometric_pressure_5: 1012.6	temperature_3: 20.9
payload: 03 67 00 D1 05 73 27 8E	barometric_pressure_5: 1012.6	temperature_3: 20.9
payload: 03 67 00 D2 05 73 27 8E	barometric_pressure_5: 1012.6	temperature_3: 21

As soon as the first string of data is received, the data will be displayed on your Cayenne Dashboard:



By choosing the graph in the upper right corner, you can display the data in time:



IBM Cloud (formerly: IBM BlueMix)

IBM Cloud is IBM's Cloud platform offering almost every service or application you need. Why use IBM?

- It gives a complete overview of a modern 'Cloud Service' (like Azure, Amazons AWS or BlueMix)
- It is free for 30 days, after that you need to register your Credit Card but it stays free for hobby use (<20 nodes, low traffic and memory).

- You can use Node Red running ‘always’ in the Cloud without bothering powering off your laptop or Raspberry PI. You get a 512MB server instance for ‘free’.
- You can use it in a workshop 😊
- There are well documented examples of connecting ‘things’.

Registration

Go to www.bluemix.net



Welcome to Bluemix, the home of 130+ unique services. Start building immediately.

[Create a free account](#) [Log in](#)

Learn more:

[Pricing](#) [Catalog](#) [Docs](#) [Support](#)

- ‘Create a free account’
- Fill in the form, using a valid email address.
- You will receive a confirmation email shortly
- Confirm the BlueMix account email
- Now Log In to the service
- It will last some time for first-time setup
- You have to agree some stuff

Terms and conditions 1 — 2 — 3 — 4

To continue, read and agree to the [Terms and Conditions](#) for your unified account.

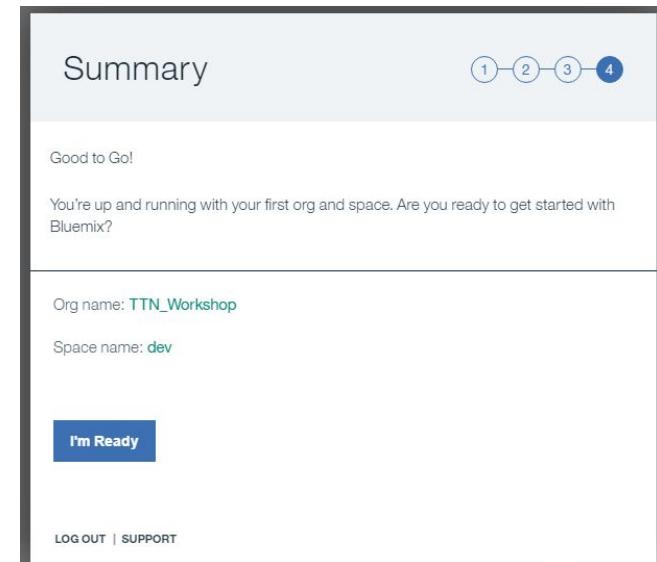
I understand and agree to the terms and conditions

[Cancel](#)
[Continue](#)

- Fill in an Organization name (‘TTN_Workshop’ is a nice name, but already in use) and select United Kingdom

[United Kingdom](#) ▾ [TTN_Workshop](#) [Create](#)

- Name your space like ‘DEV’



- Ready to start!

Create an App in BlueMix

An App is a functionality installed on the BlueMix platform. In the main dashboard choose 'Create Resource'.

The screenshot shows the IBM Bluemix dashboard. At the top, there's a navigation bar with 'IBM Cloud' and links for Catalog, Docs, Support, and Manage. Below the navigation, there's a search bar and a 'Create resource' button, which is circled in red. The main area shows two sections: 'Cloud Foundry Apps' and 'Cloud Foundry Services'. The 'Cloud Foundry Apps' section lists two apps: 'java-cloudant-form-geojson-example-20171113213230447' (Stopped) and 'TTNWorkshop' (Running). The 'Cloud Foundry Services' section lists one service: 'availability-monitoring-auto'.

In the next screen we choose 'Node-Red Starter'

The screenshot shows the IBM Cloud Catalog interface. On the left, there's a sidebar with categories like All Categories, Infrastructure, Compute, Storage, Network, Security, Containers, VMware, Platform, Boilerplates, APIs, Application Services, Blockchain, Cloud Foundry Apps, and Data & Analytics. In the main area, there are several app starters listed:

- MobileFirst Services Starter**: Start building your next mobile app with mobile services on Bluemix. (IBM)
- Node.js Cloudant DB Web Starter**: Use the Cloudant NoSQL DB service with the SDK for Node.js™ runtime. (Lite, IBM)
- Personality Insights Java Web Starter**: A simple Java app that uses the Personality Insights service to analyze... (IBM)
- Personality Insights Node.js Web Starter**: A simple Node.js app that uses Personality Insights to analyze text. (IBM)
- StrongLoop Arc**: This application is the StrongLoop Arc graphical UI, which includes tools for...
- Mendix Rapid Apps**: Model driven rapid app platform that allow users to build, integrate and...
- Node-RED Starter**: This application demonstrates how to run the Node-RED open-source project within IBM Bluemix. (Lite, Community) **(This app is circled in red)**
- Python Flask**: A simple Python Flask application that will get you up and running quickly. (Lite, Community)
- Ruby Sinatra**: Develop a Ruby web application using the Sinatra framework. (Lite, Community)
- Vaadin Rich Web Starter**: This application demonstrates how to...

Select 'Node-RED Starter':

Now you need a hostname for your Node-Red server instance, so this should be globally unique! The App name is just unique for your own environment. Be creative!

The screenshot shows the 'Create a Cloud Foundry App' page for the 'Node-RED Starter'. The form fields are as follows:

- App name:** My-Unique-Name
- Host name:** My-Unique-Name
- Domain:** eu-gb.mybluemix.net
- Choose a region/location to deploy in:** United Kingdom
- Choose an organization:** TTN_Environment
- Choose a space:** dev
- Selected Plan:** SDK for Node.js™
- Cloudant NoSQL DB** (link)

Choose 'Create', this will create a Node-RED instance.

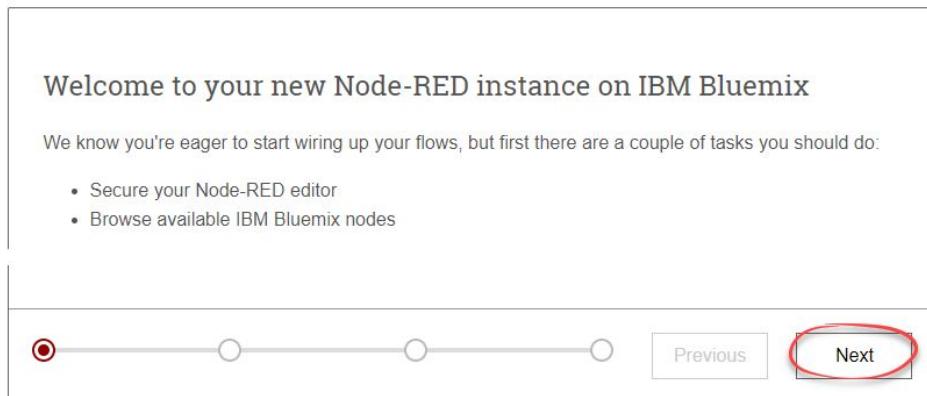
Your Node-RED server will be created and started. It can last up to 5 minutes!

After the creation you can select the 'app url':

The screenshot shows the 'Cloud Foundry apps' page with the app 'My-Unique-Name' listed. The app details are:

- Org:** TTN_Environment
- Location:** United Kingdom
- Space:** dev
- Status:** Starting
- Visit App URL** (button, highlighted with a red oval)

A screen appears, asking you to secure your Node-Red editor



Enter (and remember) a username and password

Secure your Node-RED editor

Secure your editor so only authorised users can access it

Username: frank

Password: strong

Allow anyone to view the editor, but not make any changes

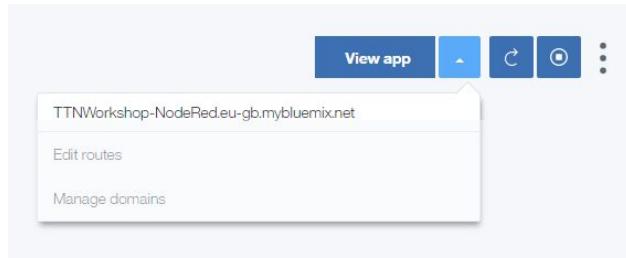
Not recommended: Allow anyone to access the editor and make changes

If you do not enter any password, your Node-Red instance will be worldwide open and available!

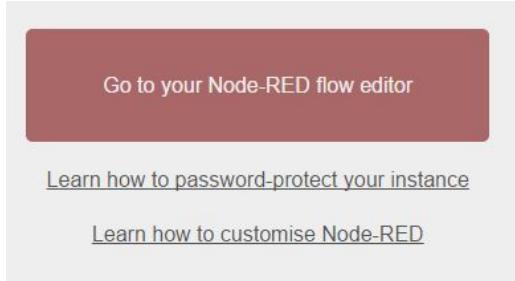
Skip the next info screen ‘Browse available IBM Bluemix nodes’.

Now press ‘Finish’ to finish the installation.

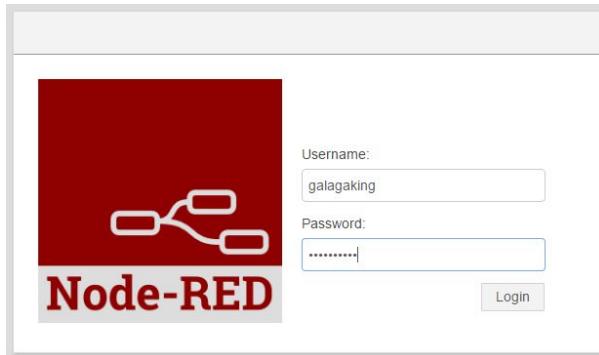
Access your Node-Red server



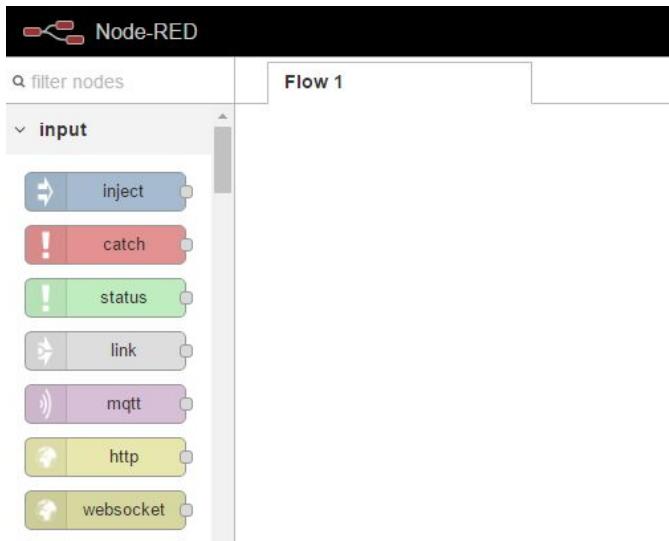
Select the name of your server, you can also use this URL to access your Node-Red server.



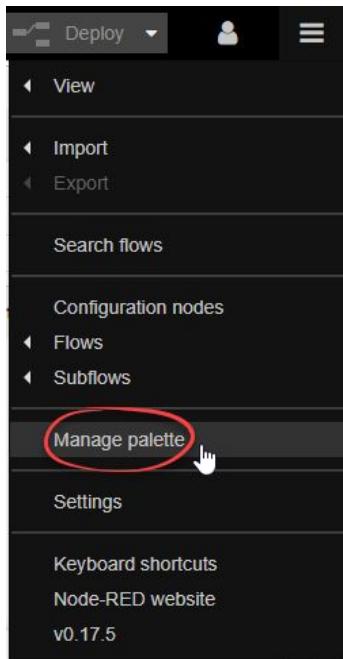
Now you have to enter your previous password:



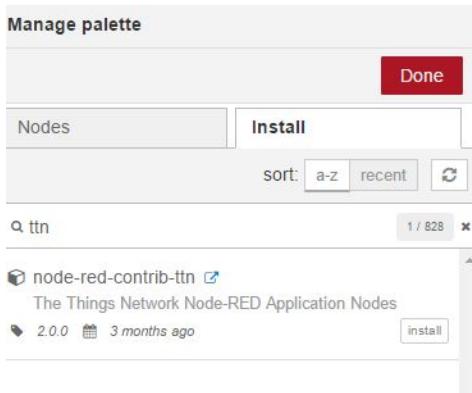
Your Node-Red is nearly ready to roll!



Now select the top right menu:



Choose manage Palette



- Search for TTN in the Install tab and click install to install the TTN nodes.
- Click again install on the warning screen.
- ‘Done’ to exit the palette management.

TTN nodes can now be added in your Node-Red flow.

Node Red introduction

We start with a clean ‘flow’. Remove any flows with the Menu->Flows->Delete function.

Add a TTN message node by drag and drop it on your flow screen.

By double clicking on the TTN message node you can change the values:

Edit ttn message node

<input type="button" value="Delete"/>	<input type="button" value="Cancel"/>	<input type="button" value="Done"/>
Name	sensor	
App	fb_nodes	<input type="button" value=""/>
Device ID	sensor_04	
Field	celcius	

The name can be any name,

App refers to your TTN console. With the Pencil you can add your application here:

ttn message > **Edit ttn app node**

<input type="button" value="Delete"/>	<input type="button" value="Cancel"/>	<input type="button" value="Update"/>
App ID	fb_nodes	
Region or Broker	eu	
Access Key	

- App ID is the written name ‘Application ID’ from the TTN console
- Region is eu (you have to fill this in!!)
- Access Key is a copy of the access key of your application (access key of application, default key).
- Check Update and select your App in the previous screen.
- Device ID is the Device ID in TTN console

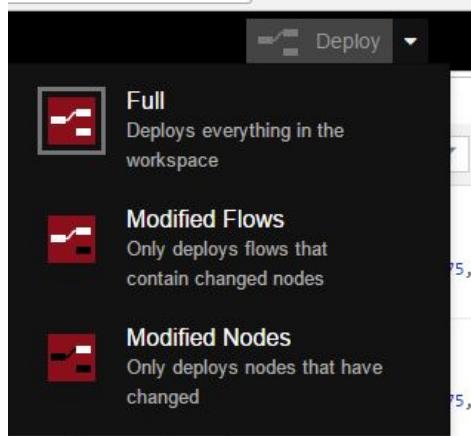
You can select celcius or humidity in ‘field’ if you use the ttn_bmp280 sketch. If you leave ‘field’ empty all the fields will be shown. Beware that you use the same payload function in TTN as used in the previous nodes workshop!! If you do not have a payload function loaded, you can use an empty ‘field’, all the output will be shown.

Now add a ‘Debug’ output node to your flow and connect the both with a wire:



Select msg.payload by double clicking on the debug node and activate it by clicking on the button on the right.

Now activate your flow with Deploy->Full:



You will see the results of your node appear in the right debug column.

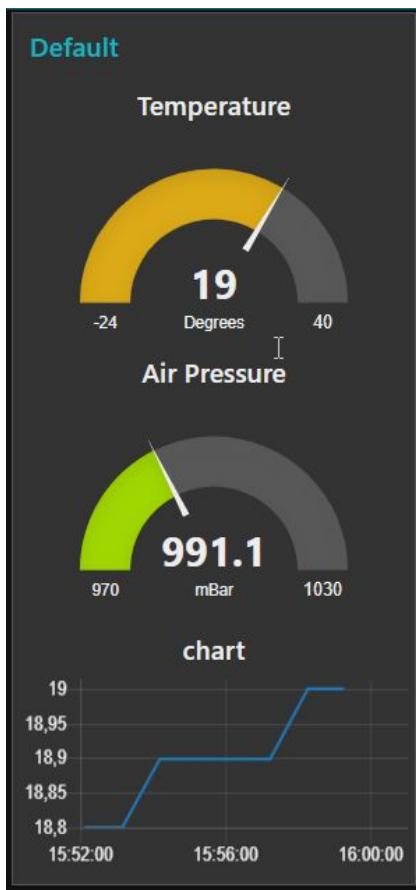
```
14-12-2017 16:02:27 node: e4e344f0.d7afa8
msg.payload : Object
▶ { barometric_pressure_4: 991.3,
temperature_3: 19 }

14-12-2017 16:02:27 node: 9b4856e4.604988
19 : msg.payload : Object
▶ { barometric_pressure_4: 991.3,
temperature_3: 19 }

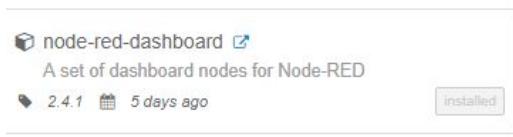
14-12-2017 16:02:27 node: 1941c1d7.aff15e
msg.payload : number
19
```

Using a web interface

Having a web-interface to get info about your nodes can be useful. We will add this to Node-RED.



Choose 'manage palette', select 'install' and find 'node-red-dashboard'



Install this module.

There are more functions added to your palette at the left side of your screen. You can add 'chart' to display a chart of your values:

node properties

Group	Default [Home]	<input type="button" value="edit"/>		
Size	auto			
Type	Gauge			
Label	Temperature			
Value format	<code>{{value}}</code>			
Units	Degrees			
Range	min -24	max 40		
Colour gradient				
Sectors	-24	... optional	... optional	40
Name	Temp Gauge			

You can select the colours, sizing and Legend. You have to create a ‘ui group’ to Group together even more charts, buttons etc.

You have to filter the JSON object for just having the value you want. You can do this by changing your TTN message node, or by adding a function:

Add a ‘function’ object:

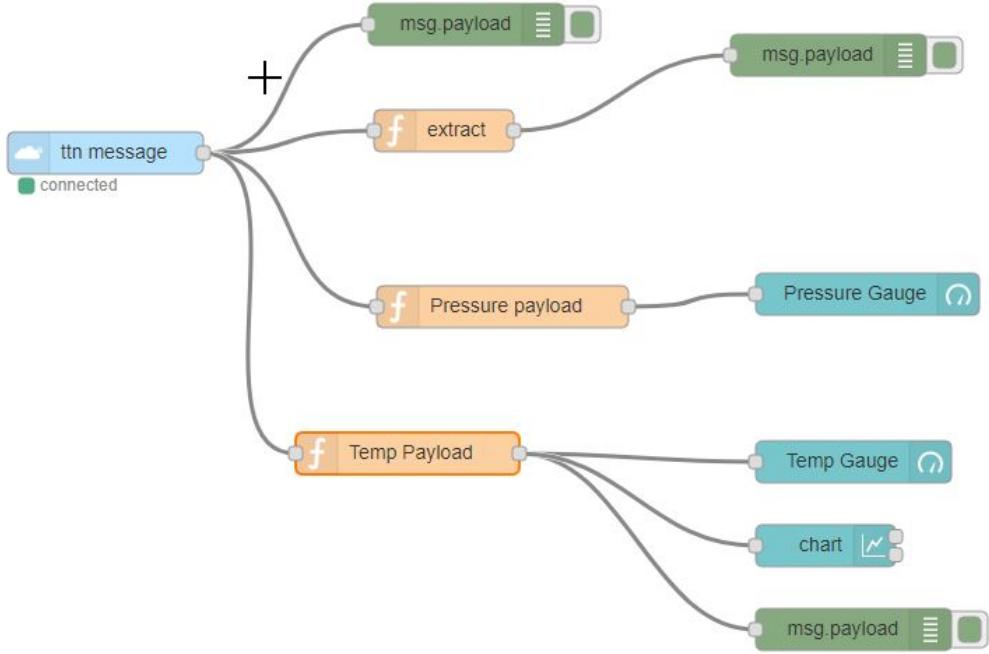


And edit the function in the way you want:

```
return {payload:msg.payload.temperature_3};
```

This refers to the ‘Cayenne’ format, but you may use payload.mbar or payload.celcius if you use these variables in the payload function.

The flow will look like this:



After creation, you can visit: <http://<yourapplicationname>.mybluemix.net/ui/#/0> to show your values

Usefull links

- Node-Red: <https://node-red.org> , <http://noderedguide.com/>
- Node-Red TTN docs: <https://www.npmjs.com/package/node-red-contrib-ttn>
- Node-Red and TTN:
<https://hansboksem.wordpress.com/2017/03/02/thethingsnetwork-weather-station-with-node-red-and-domoticz/>
- http://developers.sensetechnic.com/article/a-node-red-dashboard-using-node-red-dashboard_L
- Cayenne format:
<https://mydevices.com/cayenne/docs/lora/#lora-cayenne-low-power-payload>