



ExpertLoRaWAN

Application Note 2

Create a LoRa network and send LoRa sensor data to
Internal LoRa Server

Create a LoRa network and send LoRa sensor data to Internal LoRa Server

1. Introduction

Webdyn ExpertLoRaWAN is featured as LoRa-4G gateway using external LoRa Server. Check our app note 40.

Webdyn ExpertLoRaWAN is also featured with integrated LoRa Server inside. This means that Webdyn ExpertLoRaWAN does not depend on external LoRa Servers, like TTN and others, the management of other gateways and end LoRa devices is done internally. This is perfect to be independent of external parties.

Webdyn ExpertLoRaWAN is integrated with ChirpStack <https://www.chirpstack.io/>.

Features:

- End-devices Class A, B and C
- Adaptative data-rate
- Live frame-logging
- Channel (re)configuration
- Multi-tenant
- APIs and integration
- LoRaWAN 1.0 and 1.1 compatible

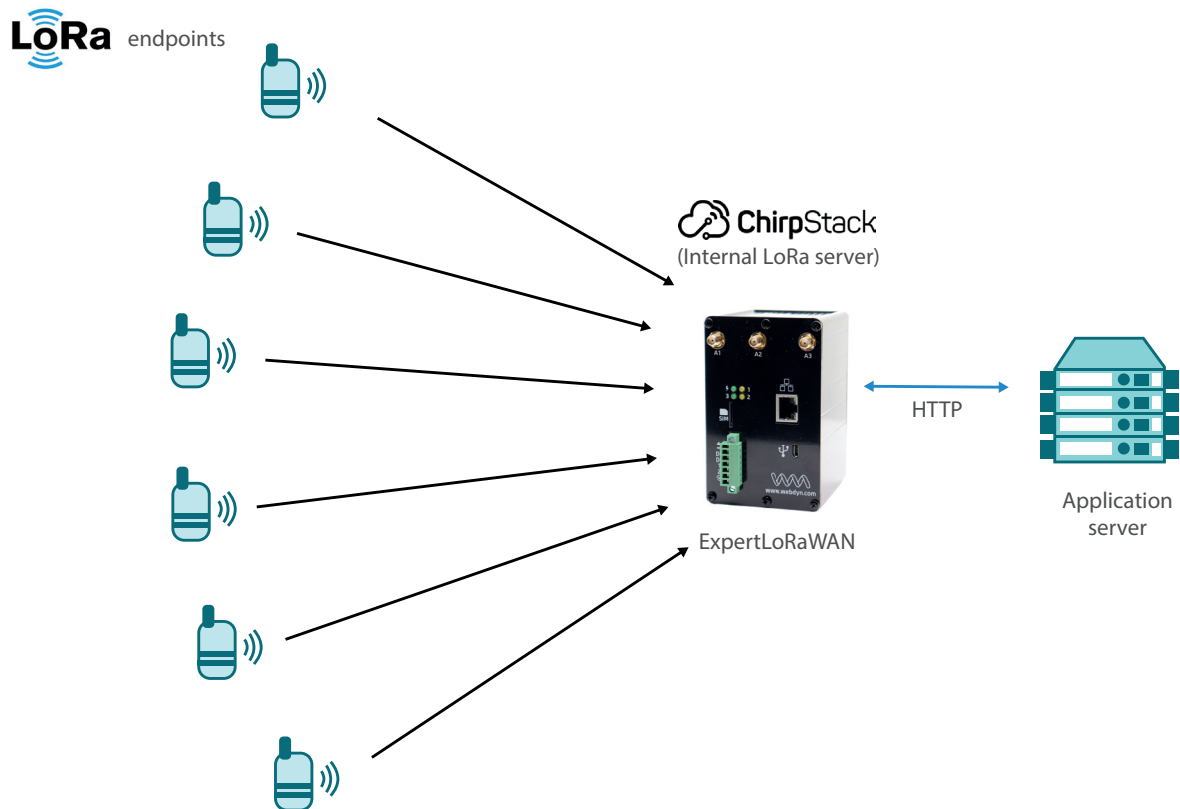
2. Scenario Details

This application note shows step by step how to create a LoRa network, with remote LoRa sensors connected and managed by the internal LoRa server.

Chirpstack LoRa internal server could send LoRa sensor payload data using HTTPS integration to a cloud device manager platform.

We can recommend the following LoRa sensors brands: Milesight (former Uralink), Adeunis, RAK

Webdyn ExpertLoRaWAN: our new Webdyn ExpertLoRaWAN with LoRa capabilities has also all Titan features, so you can still use serial RS232/RS485/USB – Eth-4G gateways, Modbus, Datalogger, VPN, etc making this product one of the most complete industrial M2M-IoT router in the market.



Please check application note "AN3 - Create a LoRa network with a LoRa slave gateway" if you want to extend the LoRa network with a slave gateway and to go deeper into technical aspects on LoRa Server implementation.

2.1 Webdyn ExpertLoRaWAN Configuration Steps

STEP 1

Access to the web interface of the Webdyn ExpertLoRaWAN using an Ethernet cable and the default 192.168.1.2 IP Address.

User: admin

Password: admin



The screenshot shows the web interface of the Webdyn ExpertLoRaWAN. At the top, there is a logo for 'webdyn' with a stylized 'w' in teal and grey, and the text 'webdyn' in black. Below the logo is the text 'flexitron group' with a small icon. To the right of the logo, it says 'powered by TITAN' in teal. The main content area is a light grey rectangle containing a login form. The form has two input fields: 'Username:' with the value 'admin' and 'Password:' with the value '*****'. Below the password field is a 'LOGIN' button. At the bottom of the page, there is a footer that reads 'Webdyn ExpertLoRaWan - Web Panel Control'.

STEP 2

It is needed to configure Webdyn ExpertLoRaWAN with SIM card network APN information.

Go to WAN -> Basic Setting.

Enable WAN interface and fill the “APN”, “Username” and “Password” fields with the information provided by your Mobile Operator. Please take care about “Sim PIN” (if SIM card is PIN enabled) and keep filled “Call Center” field as showed *99***1#.



The image shows the Webdyn user interface. At the top, there is a logo for 'webdyn' with 'flexitron group' underneath and 'powered by TITAN' to the right. On the left, a navigation menu lists various settings categories: Wan, LAN, Firewall, Serial Settings, External Devices, VPN, and Plugins. The 'Wan' category is selected, and its sub-menu 'Basic Settings' is highlighted with a red box.

The main content area is titled 'WAN Basic Settings'. It contains several configuration fields, some of which are highlighted with red boxes:

- Enabled WAN:** A checkbox that is checked, with the label 'Enable GSM WAN interface' to its right.
- Session Time:** A text input field containing the value '0', with the label 'Time in minutes (0 = always on)' to its right.
- APN:** A text input field containing 'movistar.es', with the label 'APN for wireless session' to its right.
- Username:** A text input field containing 'MOVISTAR', with the label 'Username for wireless session' to its right.
- Password:** A text input field containing 'MOVISTAR', with the label 'Password for wireless session' to its right.
- Call center:** A text input field containing '*99***1#', with the label 'Call center (normally *99***1#)' to its right.
- Sim Pin:** An empty text input field, with the label 'SIM user pin' to its right.
- Authentication:** A dropdown menu set to 'PAP', with the label 'Authentication method' to its right.
- IMSI:** An empty text input field, with the label 'If filled, only a valid SIM is allowed' to its right.
- Network selection:** A dropdown menu set to 'Auto', with the label 'Preferred network selection' to its right.
- DNS selection:** A dropdown menu set to 'Selected DNS Servers', with the label 'Preferred DNS1' to its right.
- DNS1:** A text input field containing '8.8.8.8', with the label 'Preferred DNS1' to its right.
- DNS2:** A text input field containing '8.8.4.4', with the label 'Preferred DNS2' to its right.
- Remote management:** A checkbox that is checked, with the label 'Enable remote management' to its right.
- Remote TCP Port:** A text input field containing '80', with the label 'TCP Port for remote http connections.' to its right.

At the bottom of the form, there is a 'SAVE CONFIG' button, which is pointed to by a red arrow.

Then click on “SAVE CONFIG” button and, important, reboot the router using menu Other->Reboot to allow router restart with new configuration and connect to internet.



webdyn

flexitron group

powered by **TITAN**

- ★ Wan
 - **Status**
 - Basic Settings
 - Keep Online
- ★ LAN
 - Basic Settings
 - DHCP Server
- ★ Firewall
 - NAT
 - Authorized IPs
- ★ Serial Settings
 - Serial Port3-485
 - Serial Port5-USB
 - SSL Certs
- ★ External Devices
 - Logger configuration
 - Temperature Sensor
 - ModBus Devices
 - Distance Sensor
 - Wavenis Concentrator
 - W-MBus Concentrator
 - GPS Receiver
 - Generic Serial Device
 - LoRa
- ★ VPN

Wan ▶ Status

Firmware version: 5.0.5.08

WAN IP: WAN IP (4g/3g/2g Network)

GSM Module: Cinterion
ELS61-E R2
REVISION 02.000
A-REVISION 01.000.02

IMEI: 354033091738537 Device identification

Network (2g/3g/4g): 4g (Movistar) Used network at this moment

Signal Strength: (-93dbm) Signal Strength (0 ... 31)



Internal temperature: Temperature of internal processor (°C)

REFRESH

STEP 3

Enable Lora Server and configure LoRa Gateway as follows:

- LoRa mode: Gateway Lora—Bridge (MQTT)
- ID: Define a unique ID for the gateway with 16 digits
- MQTT Broker: Internal



External Devices > LoRa Server

Server Enabled: ☒ Enable LoRa Server

Http Server Port: TCP port for LoRa Webserver

LoRaWAN Band: LoRaWAN regional band configuration

NET ID: Network Identifier (Ex: 010203)

JWT Secret: Password for API

External Devices > LoRa Gateway

Enabled: ☒ Enable LoRa Gateway

Latitude: Optional GPS Latitude. Ex: 40.39924

Longitude: Optional GPS Longitude. Ex: -3.71709

Altitude: Optional GPS Altitude. Ex: 609

LoRa mode: Select the mode of LoRa behaviour

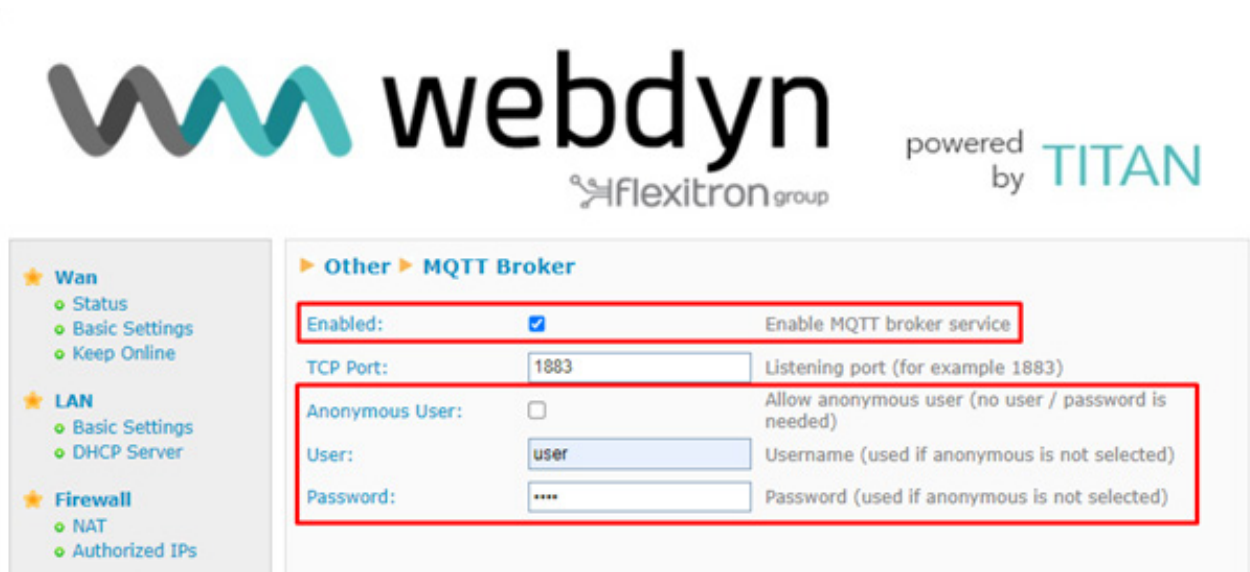
ID: Gateway ID (Ex: 010203040A0B0C0D)

MQTT Broker: Internal or external url

←

STEP 4

Enable MQTT Broker with default listening port 1883.



The screenshot shows the webdyn interface for configuring the MQTT Broker. The left sidebar contains navigation links for Wan, LAN, and Firewall. The main content area is titled "Other > MQTT Broker". It features several configuration fields: "Enabled:" with a checked checkbox and the label "Enable MQTT broker service"; "TCP Port:" with a text box containing "1883" and the label "Listening port (for example 1883)"; "Anonymous User:" with an unchecked checkbox and the label "Allow anonymous user (no user / password is needed)"; "User:" with a text box containing "user" and the label "Username (used if anonymous is not selected)"; and "Password:" with a text box containing "****" and the label "Password (used if anonymous is not selected)". Red boxes highlight the "Enabled:" checkbox, the "TCP Port:" field, and the "Anonymous User:", "User:", and "Password:" fields.

Then click on "SAVE CONFIG" button and, important, restart the router using menu Other->Reboot.

Now we can open the LoRa Server by pressing the new button that has appeared.

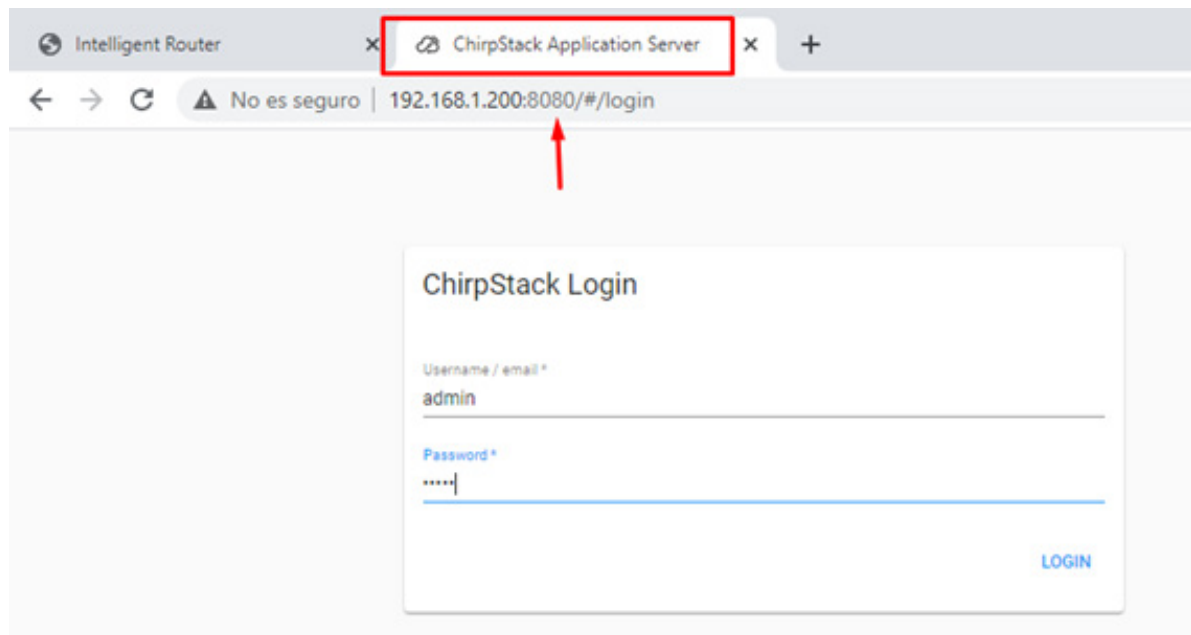


The screenshot shows the webdyn interface for configuring the LoRa Server. The left sidebar contains navigation links for Wan, LAN, Firewall, and Serial Settings. The main content area is titled "External Devices > LoRa Server". It features several configuration fields: "Server Enabled:" with a checked checkbox and the label "Enable LoRa Server"; "Http Server Port:" with a text box containing "8080" and the label "TCP port for LoRa Webserver"; "LoRaWAN Band:" with a dropdown menu showing "EU868" and the label "LoRaWAN regional band configuration"; "NET ID:" with a text box containing "000000" and the label "Network Identifier (Ex: 010203)"; and "JWT Secret:" with a text box containing "hclqr3OMDAXODk7YkJUSl" and the label "Password for API". At the bottom, there are two buttons: "SAVE CONFIG" and "OPEN LORA WEBSERVER". A red box highlights the "OPEN LORA WEBSERVER" button.

STEP 5

It is time to configure Lora Server.

New window will be opened at port 8080 (default 192.168.1.2:8080)



ChirpStack Server will be opened.

Default user: admin

Default password: admin

You can find information, guides, help and community forum at <https://www.chirpstack.io/> and <https://www.chirpstack.io/project/guides/connect-gateway/>

2.2 ChirpStack configuration STEPS

It is mandatory to follow all these steps to create a Lora Network.

- Add a Server
- Add/create a Gateway profile — connected to 1) Server
- Add/create a Service profile — must be connected to 1) Server
- Add/create a Device profile – must be connected to 1) Server
- Add/create a Gateway – must be connected to 1) Server and 2) Gateway profile
- Add/create an Application – must be connected to 3) Service Profile
- Add Devices – must be connected to 4) Device Profile
 - Repeat step 7 to add more end devices

STEP 1. Adding a server

Click on Network Server -> Add

Add a Network-server name, example WebdynExpertLoRaWAN-Server. Add the Network-server address: 127.0.0.1:8000

The screenshot shows the 'Add Network-server' form in the ChirpStack interface. The left sidebar contains a menu with 'Network-servers' highlighted. The main content area has a tabbed interface with 'GENERAL' selected. The form fields are: 'Network-server name' with the value 'WebdynExpertLoRa-Server', and 'Network-server address' with the value '127.0.0.1:8000'. An 'ADD NETWORK-SERVER' button is at the bottom right.

Enable Gateway Discovery as follows

The screenshot shows the 'Add Network-server' form with the 'GATEWAY DISCOVERY' tab selected. The 'Enable gateway discovery' checkbox is checked. Below it, the 'Interval (per day)' is set to 100, 'TX frequency (Hz)' is 0, and 'TX data-rate' is 0. The 'ADD NETWORK-SERVER' button is at the bottom right.

Check if Network Server has been created properly:

The screenshot shows the 'Network-servers' list in the ChirpStack interface. The table has two columns: 'Name' and 'Server'. The first row shows 'WebdynExpertLoRa-Server' and '127.0.0.1:8000'. The 'ADD' button is at the top right.

Name	Server
WebdynExpertLoRa-Server	127.0.0.1:8000

STEP 2. Add/create a Gateway profile

Click on Gateway-profiles -> Create

Add a name for the Gateway profile and use the Network Server created in STEP 1.

Click on the “ADD EXTRA CHANNEL” button and complete the fields of the created “Extra channel 1” with the information of the LoRa modulation in your scenario (Bandwidth, Frequency and Spreading-factors).

The screenshot shows the 'Gateway-profiles / Create' form in the ChirpStack interface. The left sidebar contains a menu with 'Gateway-profiles' highlighted. The form fields are as follows:

- Name ***: WebdynExpertLoRa-GWProfile
- Enabled channels ***: 1,2,3
- Network-server ***: WebdynExpertLoRa-Server
- Extra channel 1 (delete)**: A link to add a new channel.
- Modulation ***: LoRa
- bandwidth (kHz) ***: 125 kHz
- Frequency (Hz) ***: 868000000
- Spreading factors ***: 7,8,9,10,11,12

At the bottom right, there are two buttons: 'ADD EXTRA CHANNEL' and 'CREATE GATEWAY-PROFILE'.

Check that Gateway Profile is linked to the Network Server

The screenshot shows the 'Gateway-profiles' list view in the ChirpStack interface. The left sidebar contains a menu with 'Gateway-profiles' highlighted. The table lists the gateway profiles:

Name	Network-server
WebdynExpertLoRa-GWProfile	WebdynExpertLoRa-Server

At the top right, there are buttons for '+ CREATE' and 'HELP'. At the bottom right, there is a pagination control showing 'Rows per page: 10' and '1-1 of 1'.

STEP 3. Add/create a Service profile

Click on Service-profiles -> Create

Add a Service Profile name and use the Network Server created in STEP 1.

Check ChirpStack documentation for the other fields.

As an example:

The screenshot shows the 'Service-profiles / Create' form in the ChirpStack web interface. The left sidebar contains a menu with 'Service-profiles' highlighted. The main form area has the following fields and options:

- Service-profile name ***: WebdynExpertLoRa-ServiceProfile
- Network-server ***: WebdynExpertLoRa-Server
- ☒ **Add gateway meta-data**: GW metadata (RSSI, SNR, GW geoloc., etc.) are added to the packet sent to the application-server.
- ☒ **Enable network geolocation**: When enabled, the network-server will try to resolve the location of the devices under this service-profile. Please note that you need to have gateways supporting the fine-timestamp feature and that the network-server needs to be configured in order to provide geolocation support.
- Device-status request frequency**: 24
- Frequency to initiate an End-Device status request (request/day)**: Set to 0 to disable.
- ☒ **Report device battery level to application-server**
- ☒ **Report device link margin to application-server**
- Minimum allowed data-rate ***: 10
- Maximum allowed data-rate ***: 10

A 'CREATE SERVICE-PROFILE' button is located at the bottom right of the form.

The screenshot shows the 'Service-profiles' list view in the ChirpStack web interface. The left sidebar contains a menu with 'Service-profiles' highlighted. The main area displays a table with the following data:

Name	Network Server
WebdynExpertLoRa-ServiceProfile	WebdynExpertLoRa-Server

At the bottom right of the table, it says 'Rows per page: 10' and '1-1 of 1'.

STEP 4. Add/create a Device profile

Click on Device-profiles -> Create

Add a Device-profile name and check the LoRaWAN characteristics of the end devices you are going to use to fill the other fields. In this case we are using a Milesight device EM500-UDL.

The screenshot shows the 'Device-profiles / Create' page in the ChirpStack web interface. The left sidebar contains a menu with 'Device-profiles' highlighted. The main content area has tabs for 'GENERAL', 'JOIN (OTAA / ABP)', 'CLASS-B', 'CLASS-C', 'CODEC', and 'TAGS'. The 'GENERAL' tab is active, showing the following fields:

- Device-profile name *: MilesightDeviceProfile
- Network-server *: WebdynExpertLoRa-Server
- LoRaWAN MAC version *: 1.0.3
- LoRaWAN Regional Parameters revision *: A

A 'CREATE DEVICE-PROFILE' button is located at the bottom right of the form.

If you want to add end devices using OTAA check JOIN fields. If you will use ABP keys leave this box unmarked.

The screenshot shows the 'Device-profiles / Create' page in the ChirpStack web interface. The left sidebar contains a menu with 'Device-profiles' highlighted. The main content area has tabs for 'GENERAL', 'JOIN (OTAA / ABP)', 'CLASS-B', 'CLASS-C', 'CODEC', and 'TAGS'. The 'JOIN (OTAA / ABP)' tab is active, showing the following field:

- ☒ Device supports OTAA

A 'CREATE DEVICE-PROFILE' button is located at the bottom right of the form.

If your end devices support CLASS-B and CLASS-C communication windows, complete those sections with your specified LoRa scenario or adjust to your best performance features. These are some examples.

Device-profiles / Create

GENERAL JOIN (OTAA / ABP) **CLASS-B** CLASS-C CODEC TAGS

☒ Device supports Class-B

Class-B confirmed downlink timeout *

10

Class-B timeout (in seconds) for confirmed downlink transmissions.

Class-B ping-slot periodicity *

every 2 seconds

Class-B ping-slot periodicity.

Class-B ping-slot data-rate *

5

Class-B ping-slot frequency (Hz) *

1

CREATE DEVICE-PROFILE

Device-profiles / Create

GENERAL JOIN (OTAA / ABP) CLASS-B **CLASS-C** CODEC TAGS

☒ Device supports Class-C

Select this option when the device will operate as Class-C device immediately after activation. In case it sends a DeviceModelId mac-command when it changes to Class-C, do not select this option.

Class-C confirmed downlink timeout *

20

Class-C timeout (in seconds) for confirmed downlink transmissions.

CREATE DEVICE-PROFILE

CODEC feature is useful to decode and encode the LoRa frame payload of your end devices.

When selecting the Custom JavaScript codec functions option, you can write your own JavaScript functions or use functions provided by your end device manufacturer (read documentation / help).

In this case, the Milesight EM500-UDL decoder examples files are available at <https://github.com/Milesight-IoT/SensorDecoders>

Device-profiles / MilesightDeviceProfile

GENERAL JOIN (OTAA / ABP) CLASS-B CLASS-C **CODEC** TAGS

☒ Custom JavaScript codec functions

By defining a payload codec, ChirpStack Application Server can encode and decode the binary device payload for you.

```

1 /**
2  * Payload Decoder for Chirpstack and Milesight network server
3  *
4  * Copyright 2021 Milesight IoT
5  *
6  * @product EM500-UDL
7  */
8 function Decode(fPort, bytes) {
9   var decoded = {};
10
11   for (var i = 0; i < bytes.length; i++) {
12     var channel_id = bytes[i++];
13     var channel_type = bytes[i++];
14     // BATTERY
15     if (channel_id === 0x01 && channel_type === 0x75) {
16       The function must have the signature function Decode(fPort bytes) and must return an object. ChirpStack Application Server will convert this object to JSON.
17     }
18   }
19 }
20
21 // Encode encodes the given object into an array of bytes.
22 // - fPort contains the LoRaWAN fPort number
23 // - obj is an object, e.g. {"temperature": 22.5}
24 // - variables contains the device variables e.g. {"calibration": "3.5"} (both the key / value are of type string)
25 // The function must return an array of bytes, e.g. [225, 230, 255, 0]
26 function Encode(fPort, obj, variables) {
27   return [];
28 }

```

DELETE

STEP 5. Add/create a Gateway

Now we add the Webdyn ExpertLoRaWAN as a Gateway.

Click on Gateways -> Create

Add a Gateway name and Gateway description.

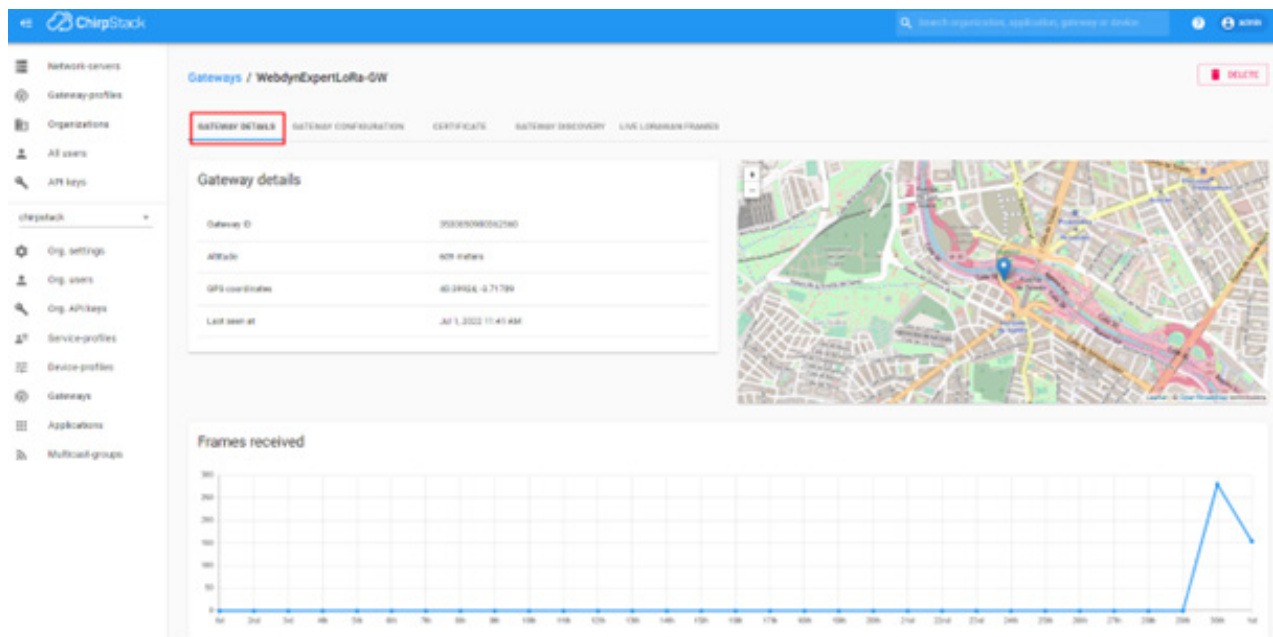
Also fill Gateway ID with the same ID configured in Webdyn ExpertLoRaWAN Configuration Step 3, in this example 3530850900362560.

Use the Gateway-profile set in STEP 2

The screenshot shows the 'Gateways / Create' form in the ChirpStack interface. The form has three tabs: GENERAL, TAGS, and METADATA. The GENERAL tab is active. The form fields are as follows:

- Gateway name: WebdynExpertLoRa-GW
- Gateway description: Webdyn ExpertLoRa Gateway
- Gateway ID: 35 30 85 09 00 36 25 60 (highlighted with a red box)
- Network-server: WebdynExpertLoRa-Server
- Gateway-profile: WebdynExpertLoRa-GWProfile
- Gateway discovery enabled: ☒ Gateway discovery enabled
- Gateway altitude (meters): 622
- Gateway location: (set to current location)

If successful, after some minutes, you will get some live information

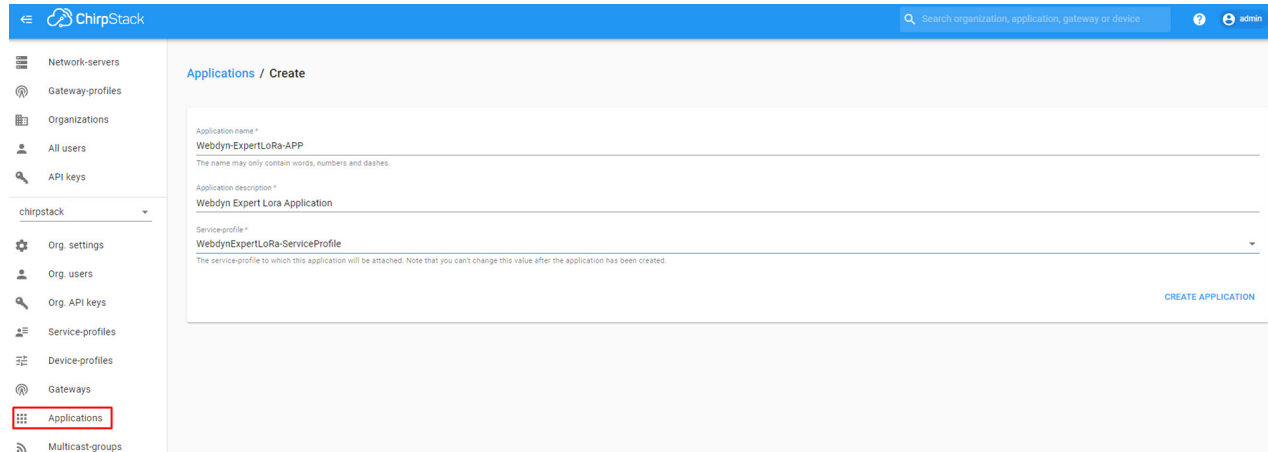


STEP 6. Add/create an Application

Next point is creating a new Application. In this section we will also add the end LoRaWAN devices and see the payload.

Click on Application -> Create

Add an Application name and Application description and use Service Profile configured in STEP 3.



The screenshot shows the ChirpStack web interface. The left sidebar contains a menu with items: Network-servers, Gateway-profiles, Organizations, All users, API keys, chirstack (selected), Org. settings, Org. users, Org. API keys, Service-profiles, Device-profiles, Gateways, Applications (highlighted with a red box), and Multicast-groups. The main content area is titled 'Applications / Create'. It contains three input fields: 'Application name *' with the value 'Webdyn-ExpertLoRa-APP', 'Application description *' with the value 'Webdyn Expert Lora Application', and 'Service-profile *' with the value 'WebdynExpertLoRa-ServiceProfile'. A 'CREATE APPLICATION' button is located at the bottom right of the form.

Now we will add the end LoRaWAN devices.

STEP 7. Adding devices

For this application note we will use two devices from different brands: Milesight (former Ursalink) and Adeunis.

You will need this information from the LoRaWAN device node:

- DEVICE EUI (DEV EUI)
- APPLICATION KEY (APP KEY)

Generally, you can get this information from your device provider, the user manual, sensor label or accessing the device through a mobile or web application.

Enter in DEVICES section inside the application and create a new device.

Example Milesight:

We have extracted the DEV EUI from the device label and used the default APP KEY provided in the user manual (<https://resource.milesight-iot.com/milesight/document/em500-series-user-guide-en.pdf>).

Parameters	Description
Device EUl	Unique ID of the device which can also be found on the label.
App EUl	Default App EUl is 24E124C0002A0001.
Application Port	The port used for sending and receiving data, default port is 85.
Join Type	OTAA and ABP mode are available.
Application Key	Appkey for OTAA mode, default is 5572404C696E6B4C6F52613230313823.
Device Address	DevAddr for ABP mode, default is the 5 th to 12 th digits of SN.
Network Session Key	Nwkskey for ABP mode, default is 5572404C696E6B4C6F52613230313823.
Application Session Key	Appskey for ABP mode, default is 5572404C696E6B4C6F52613230313823.
RX2 Data Rate	RX2 data rate to receive downlinks.
RX2 Frequency	RX2 frequency to receive downlinks. Unit: Hz
Spread Factor	If ADR is disabled, the device will send data via this spread factor.
Confirmed Mode	If the device does not receive ACK packet from network server, it will resend

ChirpStack

Applications / WebdynExpertLoRa-APP / **Devices** / Create

GENERAL VARIABLES TAGS

Device name*
Milesight-EM500-UDL
The name may only contain words, numbers and dashes.

Device description*
Milesight EM500-UDL Sensor

Device EUl*
24 E1 24 12 6A 21 74 74 MSB ↺

Device profile*
MilesightDeviceProfile

☐ Disable frame-counter validation
Note that disabling the frame-counter validation will compromise security as it enables people to perform replay-attacks.

☐ Device is disabled
ChirpStack Network Server will ignore received uplink frames and join-requests from disabled devices.

CREATE DEVICE

After the device is created, enter the APP Key in the KEYS (OTAA) section, keep Gen Application key in blank and then click on SET DEVICE-KEYS.

ChirpStack

Applications / WebdynExpertLoRa-APP / Devices / Milesight-EM500-UDL DELETE

DETAILS CONFIGURATION **KEYS (OTAA)** ACTIVATION DEVICE DATA LORAWAN FRAMES FIRMWARE

Application key*
55 72 40 4c 69 6e 6b 4c 6f 52 61 32 30 31 38 23 MSB ↺ ↻ ↵

For LoRaWAN 1.0 devices. In case your device supports LoRaWAN 1.1, update the device-profile first.

Gen Application key
.....

For LoRaWAN 1.0 devices. This key must only be set when the device implements the remote multicast setup specification / firmware updates over the air (FOTA). Else leave this field blank.

SET DEVICE-KEYS

After a while the sensor should appear enabled and you should start receiving LoRa frames.

ChirpStack

Search organization, application, gateway or device

admin

Applications / WebdynExpertLoRa-APP / Devices / Milesight-EM500-UDL

DETAILS CONFIGURATION KEYS (OTAA) ACTIVATION DEVICE DATA LORAWAN FRAMES FIRMWARE

Details

Name	Milesight-EM500-UDL
Description	Milesight EM500-UDL Sensor
Device-profile	MilesightDeviceProfile

Status

Last seen at	Jul 1, 2022 11:46 AM
State	enabled

Enqueue downlink payload

Port *

Please note that the fPort value must be > 0.

You can inspect the LoRaWAN frames on this section and check the decoded payload in the Device Data section.

ChirpStack

Search organization, application, gateway or device

admin

Applications / WebdynExpertLoRa-APP / Devices / Milesight-EM500-UDL

DETAILS CONFIGURATION KEYS (OTAA) ACTIVATION DEVICE DATA LORAWAN FRAMES FIRMWARE

HELP PAUSE DOWNLOAD CLEAR

DOWNLINK 11:52:55 AM UnconfirmedDataDown 0130c6d1

UPLINK 11:52:55 AM ConfirmedDataUp 0130c6d1

▼ info: 0 1 item

- 0: 0 14 keys
 - gatewayID: "935080900362560"
 - time: null
 - timeEndOfPSepoch: null
 - rx: -17
 - software: 3.5
 - channel: 2
 - rChain: 1
 - board: 0
 - antenna: 0
 - location: 0 5 keys
 - latitude: 40.39024
 - longitude: -3.71709
 - altitude: 609
 - source: "UNKNOWN"
 - accuracy: 0
 - finalTimestampType: "NONE"
 - correction: "YvW5d4m"
 - spinID: "5a5a6d27c5e4d22a738-039794810a031"
 - ordStatus: "CRC_OK"
- info: 0 3 keys
 - frequency: 868500000
 - modulation: "LORA"
 - loraModulationInfo: 0 4 keys
 - bandwidth: 125
 - spreadingFactor: 9
 - codeRate: "4/5"
 - polarizationInversion: false

▼ payload: 0 3 keys

- mbid: 0 2 keys
 - mtType: "ConfirmedDataUp"
 - major: "LoRaWANv1.0"
- macPayload: 0 3 keys
 - fhdr: 0 4 keys
 - devAddr: "0130c6d1"
 - fctrl: 0 5 keys
 - addr: true
 - ackReq: false
 - ack: false
 - pending: false
 - channel: false
 - fctrl: 47
 - fctrl: null
 - phys: 0 1 item
 - 0: 0 1 key
 - bytes: "2220pm"
 - mic: "028a4f0c"

ChirpStack

Applications / WebdynExpertLoRa-APP / Devices / Milesight-EM500-UDL

DETAILS CONFIGURATION KEYS (OTAA) ACTIVATION **DEVICE DATA** LORAWAN FRAMES FIRMWARE

11:54:55 AM up

```

applicationID: "1"
applicationName: "WebdynExpertLoRa-APP"
deviceName: "Milesight-EM500-UDL"
devAddr: "24e124126a217474"
• nwkID: 0 1 item
  • nwkID: 0 14 keys
    gatewayID: "3350830000000000"
    time: null
    timeDeviceEpoch: null
    rssi: -115
    loraSNR: 11.5
    channel: 0
    rxClean: 1
    board: 0
    antenna: 0
  • location: 0 5 keys
    latitude: 40.39924
    longitude: -3.71709
    altitude: 600
    source: "UNKNOWN"
    accuracy: 0
    fixTimestampType: "NONE"
    comment: "Type1A++"
    uplinkID: "7c1e1d5d0c7e435a46f4d599119f53f"
    crcStatus: "CRC_OK"
  • nwkID: 0 3 keys
    frequency: 868100000
    modulation: "LORA"
  • loraModulationInfo: 0 4 keys
    bandwidth: 125
    spreadingFactor: 9
    codeRate: "4/5"
    polarizationInversion: false
    sdr: true
    sf: 9
    ccr: 58
    PPM: 85
    dev: "AdaLora"
  • objectID: 0 1 key
    • objectID: 000
    tags: 0 0 keys
    confirmedStatus: true
    devAddr: "2130bd01"
  
```

You can repeat this step to add more end devices.

We added a temperature sensor from Adeunis under a different Device Profile.

ChirpStack

Applications / WebdynExpertLoRa-APP / Devices / **Adeunis-Temp**

DETAILS CONFIGURATION KEYS (OTAA) ACTIVATION DEVICE DATA **LORAWAN FRAMES** FIRMWARE

12:04:20 PM UnconfirmedDataUp 00ae1429

```

• nwkID: 0 1 item
  • nwkID: 0 14 keys
    gatewayID: "3350830000000000"
    time: null
    timeDeviceEpoch: null
    rssi: -107
    loraSNR: 7.2
    channel: 0
    rxClean: 1
    board: 0
    antenna: 0
  • location: 0 5 keys
    latitude: 40.39924
    longitude: -3.71709
    altitude: 600
    source: "UNKNOWN"
    accuracy: 0
    fixTimestampType: "NONE"
    comment: "Type1A++"
    uplinkID: "7c1e1d5d0c7e435a46f4d599119f53f"
    crcStatus: "CRC_OK"
  • nwkID: 0 3 keys
    frequency: 868100000
    modulation: "LORA"
  • loraModulationInfo: 0 4 keys
    bandwidth: 125
    spreadingFactor: 12
    codeRate: "4/5"
    polarizationInversion: false
    sdr: true
    sf: 9
    ccr: 58
    PPM: 85
    dev: "AdaLora"
  • objectID: 0 1 item
    • objectID: 000
    tags: 0 0 keys
    confirmedStatus: true
    devAddr: "2130bd01"
  
```

ChirpStack

Applications / WebdynExpertLoRa-APP

DEVICES APPLICATION CONFIGURATION INTEGRATIONS FUOTA

CREATE

Last seen	Device name	Device EUI	Device profile	Link margin	Battery
a minute ago	Adeunis-Temp	0018b21000004502	AdeunisDeviceProfile	5 dB	91.73%
a few seconds ago	Milesight-EM500-UDL	24e124126a217474	MilesightDeviceProfile	12 dB	

Rows per page: 10 1/2 of 2

Configuring a MQTT Client to extract the device's data

Chirpstack publishes all the data it receives from the end devices to the Webdyn ExpertLoRaWAN MQTT broker in a default topic. Therefore, it is possible to receive data from your end devices subscribing to their MQTT topic.

All events are exposed in the following default event topic: `application/[ApplicationID]/device/[DevEUI]/event/[EventType]`

For debugging, we are going to use the command-line tool “`mosquitto_sub`” which is part of the Mosquitto MQTT broker. We will use the DynDNS address configured for the Webdyn ExpertLoRaWAN and subscribe to “`application/+ /device/+ /event/up`” to receive the data from all the uplink events of all the devices in the application.

Notice that, as we configured the decoder function in the CODEC section, the Chirpstack publishes the device data with the decoded payload.

```
C:\Program Files\mosquitto>mosquitto_sub -h expertloramaster.ddns.net -t "application/+ /device/+ /event/up"
{"applicationID": "1", "applicationName": "ExpertLoRaWAN-App", "deviceName": "Ursalink-EM500", "devEUI": "24e124126a217474", "rxInfo": [{"gatewayID": "3530850900362560", "uplinkID": "c6fd0fa9-f044-471e-b2d6-4c9316dd1d5f", "name": "ExpertLoRaWAN-GW", "rssi": -43, "loRaSNR": 11.5, "location": {"latitude": 40.39924, "longitude": -3.71709, "altitude": 609}}, {"gatewayID": "3530850900362560", "uplinkID": "8393093c-7bbc-435b-beac-84c1b0f82c5c", "name": "ExpertLoRaWAN-GW", "rssi": -103, "loRaSNR": -7.8, "location": {"latitude": 40.39924, "longitude": -3.71709, "altitude": 609}}, {"gatewayID": "3530850900362560", "uplinkID": "5f48fe2e-3fed-4bdf-aebd-c3ee28b5a725", "name": "ExpertLoRaWAN-GW", "rssi": -101, "loRaSNR": -8, "location": {"latitude": 40.39924, "longitude": -3.71709, "altitude": 609}}], "txInfo": {"frequency": 868500000, "dr": 3}, "adr": true, "fCnt": 4, "fPort": 85, "data": "A4JBW==", "object": {"distance": 1856}}
```

You can also subscribe to specifics applications, devices and events. Use the Chirpstack help page for more information on the documented event types: <https://www.chirpstack.io/application-server/integrations/mqtt/>.

Note: You can also send sensor's data to third parties using other Integration methods available inside ChirpStack Applications.

