Introduction 1.1
Overview 1.2

Quick Start

Main User Interface 2.1
Create a Service Profile 2.2
Create an Application 2.3
Create a Gateway 2.4
Create a Device profile 2.5
Create a Device and Activation 2.6
Observe Device data 2.7

Basic functions

Service Profile 3.1
Device Profile 3.2
Application 3.3
Custom Codec 3.4
MQTT Authorize 3.5
Device 3.6
Import and Export Devices 3.7
Global Search 3.8
Gateway 3.9
Users 3.10

Downlink Data Test

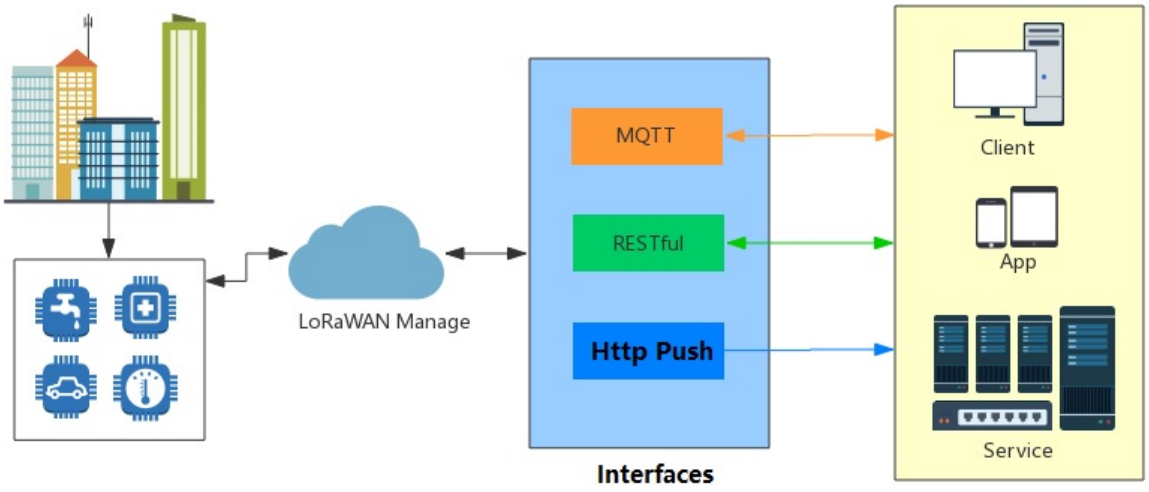
Data downlink description 4.1
Data Downlink Tool 4.2

- AdoModel LoRaWAN NS Platform User Guide
- Overview

AdoModel LoRaWAN NS Platform User Guide

Overview

LoRaWAN Manage is a universal LoRaWAN management platform developed based on the loraserver project, supporting multiple frequency bands and configurations. By managing applications, devices, and gateways, LoRaWAN network construction and maintenance functions can be achieved. The platform supports standard RESTful APIs and MQTT communication interfaces to achieve user-defined data acquisition and application development. The document mainly includes an explanation of the platform's basic business structure and an explanation of interface usage.



Copyright © AdoModel Tech Inc. All rights reserved. 2026-09-11 15:19:28

- Overview
  - 1. Users
  - 2. Network-servers
  - 3. Service-profiles
  - 4. Device-profiles
  - 5. Applications
  - 6. Devices
  - 7. Gateways

Overview

1. Users

Users are the model used in the platform to distinguish different identities, and are the top-level data model of the platform. Different users will have different `network service profiles`, `device profiles`, `applications`, `devices`, and `gateways`.

Each user will obtain a **token token** (using JWT standard) through the `login authentication` function. Through the token access interface, the platform will return the **resource** to which the user belongs.

In addition, users of the `administrator` type will have all platform permissions, be able to operate other user attributes, switch identities, or directly access designated user **resources**, and control the `system settings`. Administrator type users can only be used during **private deployment**, while public cloud administrator type users are controlled by AdoModel.

2. Network-servers

Network servers also belong to the top-level data model of the platform. In the platform architecture, by defining different network servers (server services in loraserver), users can be provided with LoRaWAN `network servers` with **different configurations** (such as LoRaWAN frequency band: US915 \ eu433...).

For normal users, when creating a `service profile`, they can (and should) retrieve the `network server list` supported by the platform for selection, while when creating a `device profile` and `gateway`, they can (and must) retrieve the `network server list` that belongs to the user.

Only administrator users have the permission to view and edit the platform's `network server list`.

3. Service-profiles

In the platform architecture, different configurations of `network services` are provided to users through an application service (i.e. the platform itself). The core is to create a `service profile`, and the system will record the routing direction of the network service data and push it to the application. Therefore, when users want to use other data models, they must first create a `service profile` to mark **whether the `network server` can be used by the user**.

Each normal user can create any number of `service profiles`, which include corresponding `network servers` (added and managed by administrator users) and device status request related configurations. When creating an `application`, the `service profile` belonging to that user can (and must) be called for selection.

As `service profile` is controlled by the routing of data, the selected `network server` when creating `device profile` and `gateway` must be accessible to the user.

4. Device-profiles

In order to reduce the coupling of LoRaWAN parameters when creating devices, the platform provides a configuration abstraction. When creating a device, the device will use the `device profile`, which mainly includes the `network server` used, the network access mode, and the attributes of LoRaWAN device types (class A, class C).

Each normal user can create any number of device profiles, and the selected `network server` during creation must be one that the user can use.

5. Applications

An application serves as a container for loading `devices`, and the device must belong to an application. Under the application, attributes such as `data subscription and push` and `MQTT authorize` can also be set for all device data within the application, which are the basic attributes for the platform to distinguish data.

Each normal user can create a limited number of applications, depending on the overall system allowance and the number allowed by the administrator for the user to add. When creating, a `Service profile` must be selected, which will use its corresponding `network server` and some network parameters.

After determining the Service profile, when calling the **device profiles list belonging to the application** under this application, the device profile using the same `network server` will be automatically presented`

6. Devices

The device is used to record the identity of the actual LoRaWAN nodes in the network, and it belongs to a certain application.

When adding a device, it is necessary to retrieve **the `device profile` list that belongs to the application to which the device belongs**. However, when the application uses a certain `service profile`, the device profile list can be filtered by the application to obtain effective `device profile`. For example, for an application that uses a US915 network server, adding a device that uses EU868 device profile is invalid.

After creating the device, several sub attributes can be operated on, such as device `keys`, obtaining `live data` or `live LoRaWAN frames`, etc.

7. Gateways

Gateway is an independent model that belongs to the platform and is only used for **monitoring**. Even if a gateway is not added to the platform, as long as the actual LoRaWAN gateway is connected to the platform according to configuration, it can still work normally.

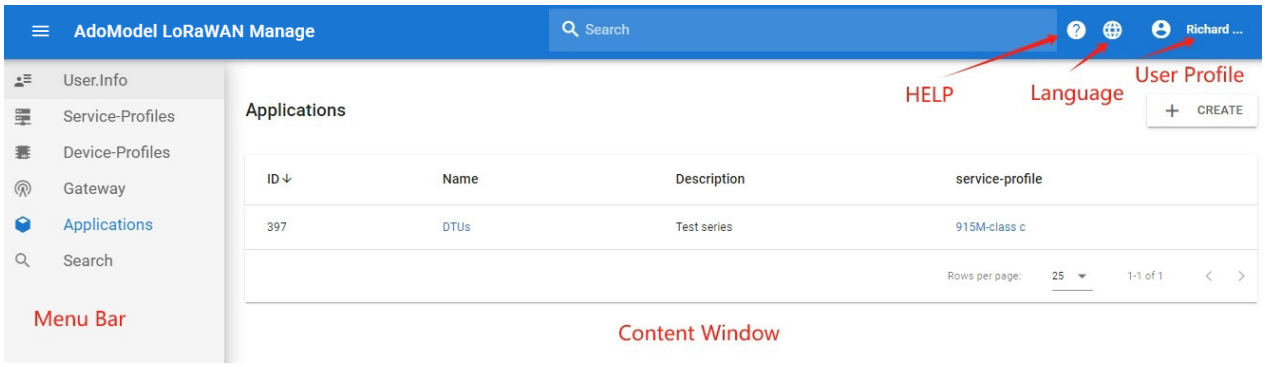
When creating a gateway, the selected 'network server' must be accessible to the user.

Copyright © AdoModel Tech Inc. All rights reserved. 2026-09-11 15:17:26

- Home User Interface

Home User Interface

After logging into the platform, the platform will default to the application list window. The main interface layout of the platform is as follows:



- Menu bar: switchable main content window
- Main content window: Display corresponding information
- Help: Switch to help document
- Language: The system supports switching between Chinese and English languages
- User menu: can be used to view personal information, update Password,log out

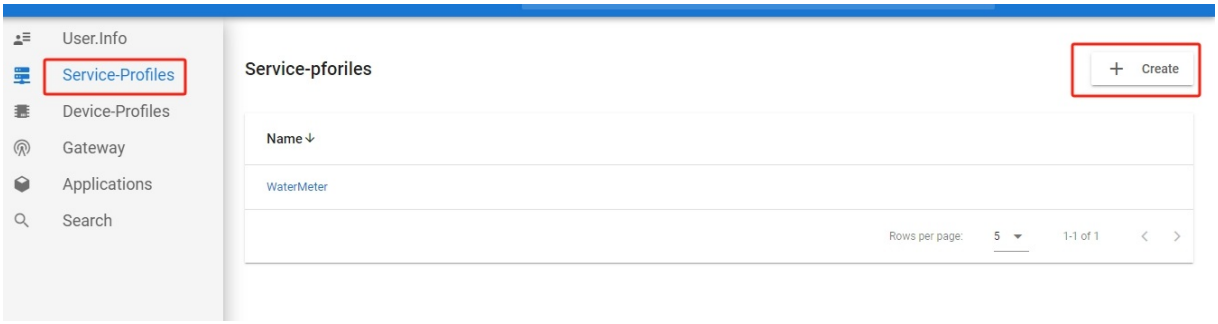
Copyright © AdoModel Tech Inc. All rights reserved. 2026-09-11 15:17:21

- Create a Service Profile

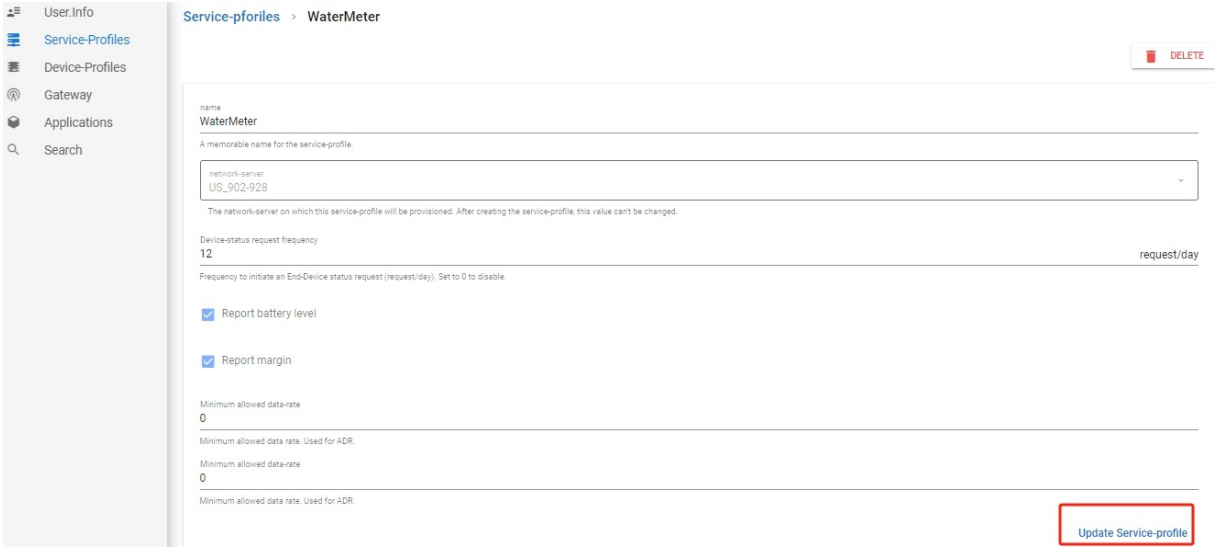
Create a Service Profile

Network service profile is a prerequisite for creating applications, gateways, devices, etc. A user must create a network service profile in order to further perform other operations.

1. Enter service profile menu



2. Click 'Create' button
3. Enter according to the form prompts and select a required network server (managed by the platform administrator). After completion, click Create.



Complete the creation of a network service configuration here.

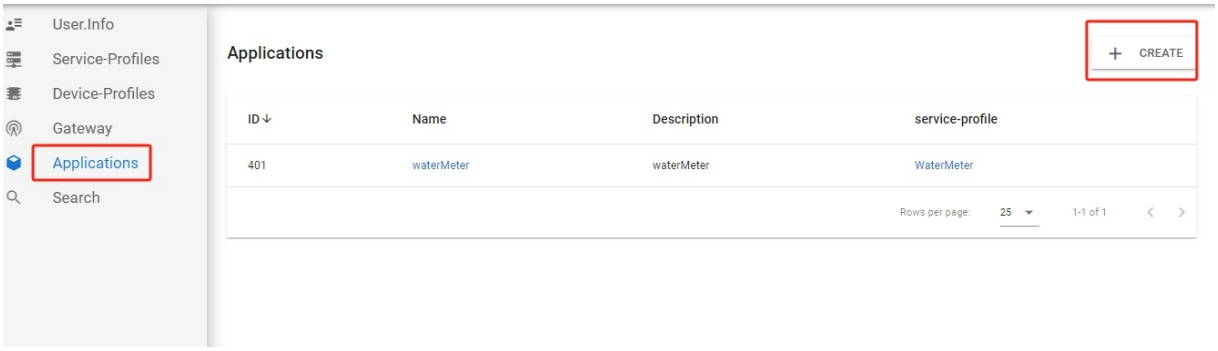
Copyright © AdoModel Tech Inc. All rights reserved. 2026-09-11 15:17:21

- Create an Application

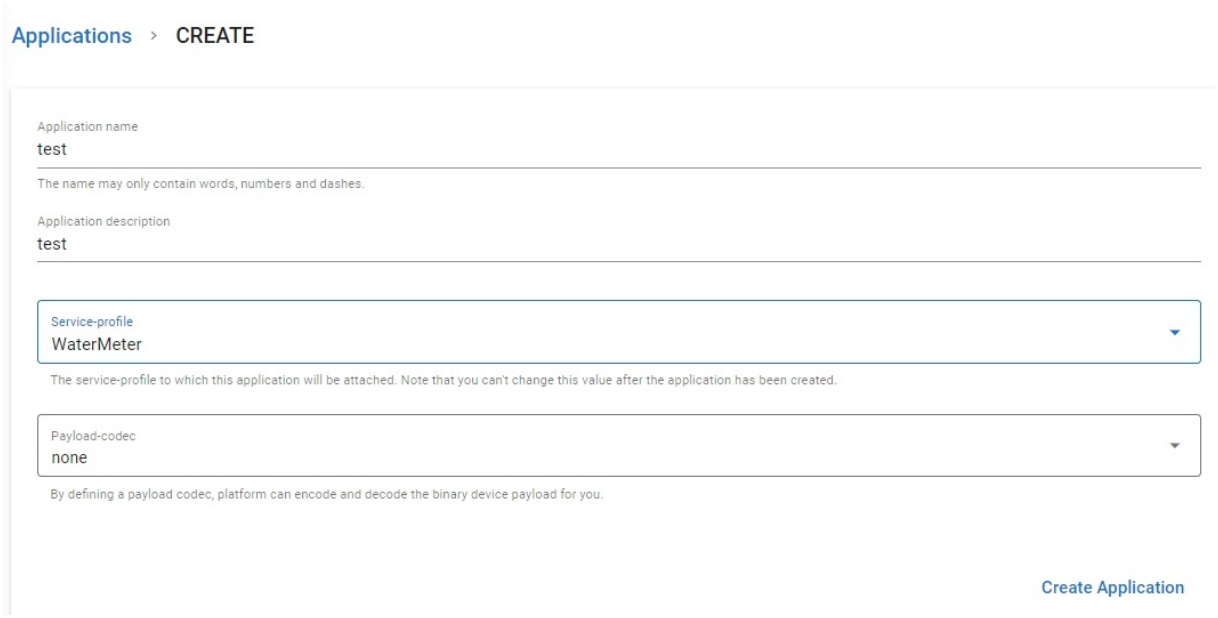
Create an Application

On the platform, all node devices must be under a certain application. After created the Service Profile, the application can be created.

1. Enter the applications menu



2. Click the create button
3. Based on the form input information and selecting a service profile, if the user has not created a service profile, they cannot obtain a list of service profile:



The payload-codec here can be selected as none. If needed, you can write it yourself. For more details, please refer to Custom Codec

Copyright © AdoModel Tech Inc. All rights reserved. 2026-09-11 15:17:22

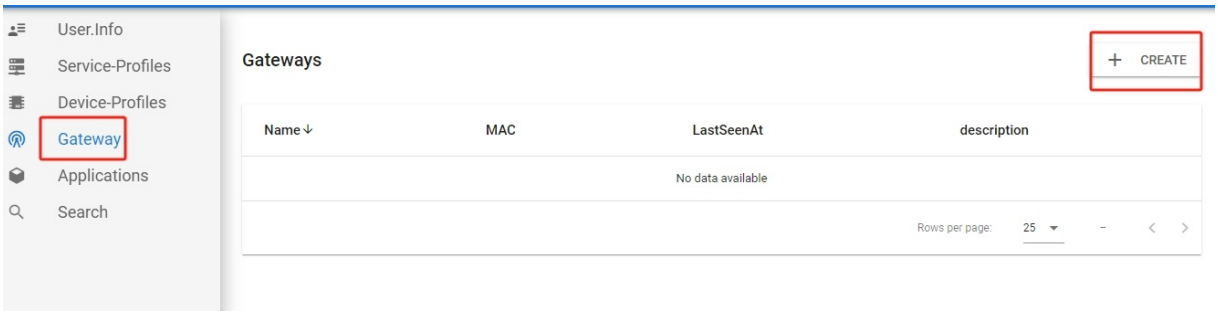
- Create a Gateway

Create a Gateway

On the platform, the purpose of creating a gateway is to view the status of the gateway and view the live LoRaWAN data frames received by the gateway, but even without create a gateway on the platform, the gateway can still function properly when it is correct configured.

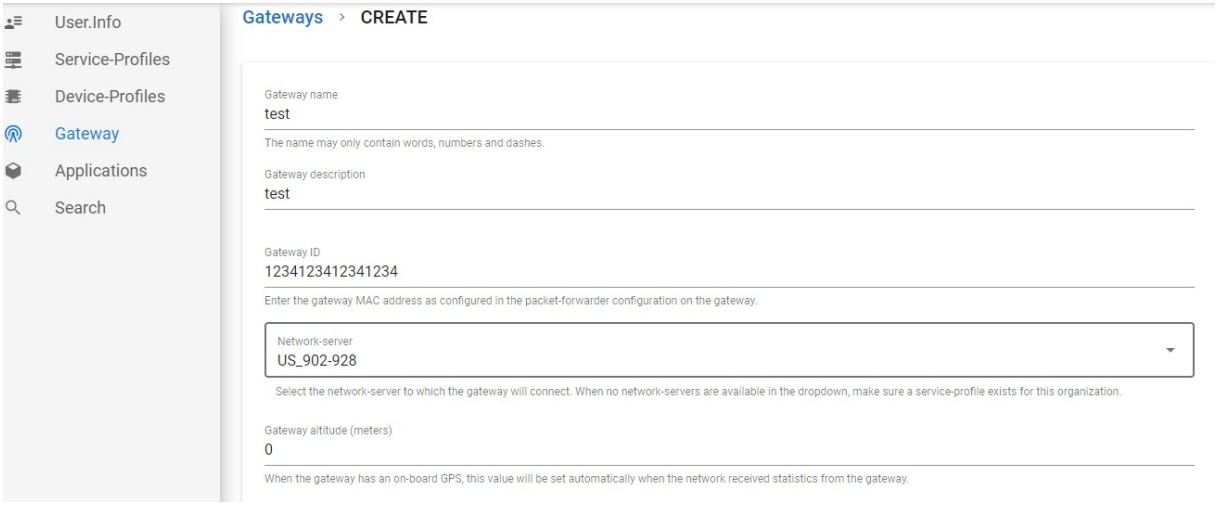
Please refer to the relevant gateway documentation for gateway access operations.

1. Enter Gateway menu



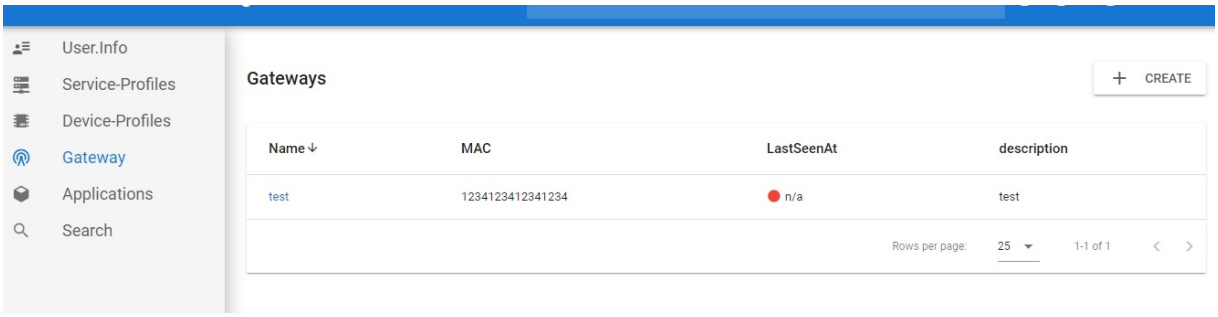
2. Click 'Create' button

3. Based on the form data, select a network server (if the user has not created any service profile, the list of network servers cannot be obtained)



Note: Once the gateway ID, also known as the gateway MAC address, is created, it cannot be modified again. If entered incorrectly, please delete it and create it again.

4.After completing the gateway creation, the gateway status can be viewed in the gateway list or details, including whether it is already online, etc.



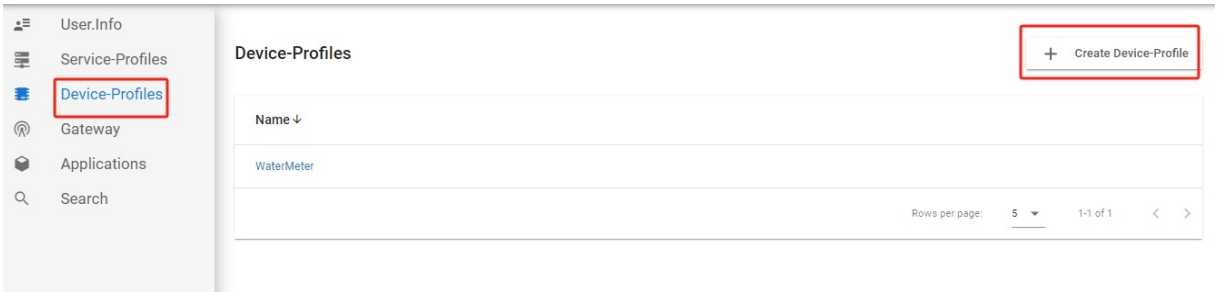
Copyright © AdoModel Tech Inc. All rights reserved. 2026-09-11 15:17:21

- Create a Device Profile

Create a Device Profile

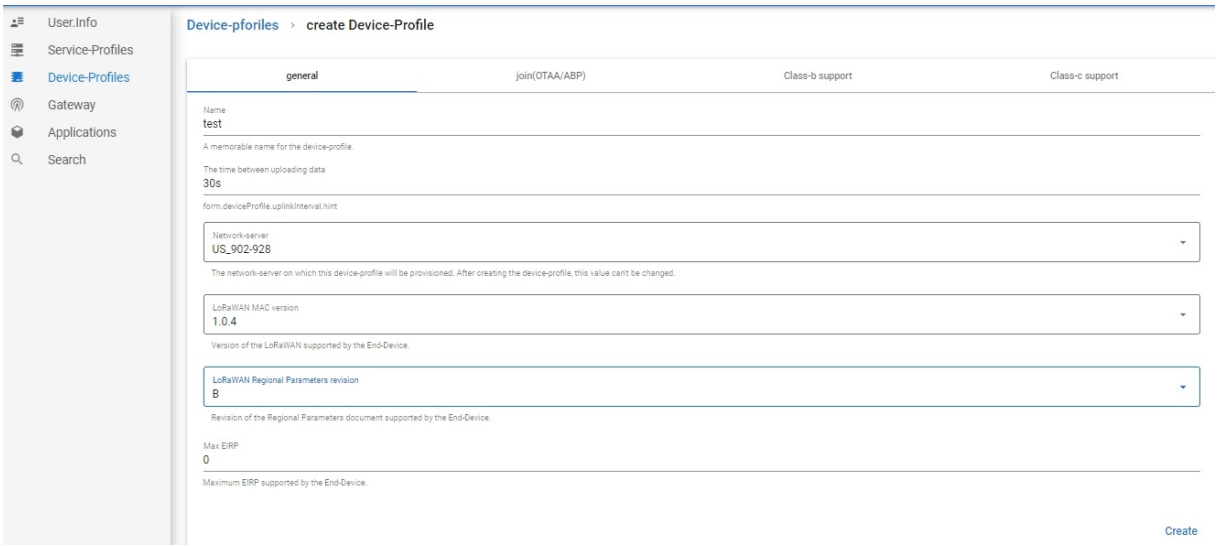
Device profile is an abstract property of a device. As long as a device profile is created and a device profile is selected during device creation, the properties of that device belong to the description in the device profile.

1. Enter device profile menu



2. Click 'Create Device-profile'

3. Based on the form data and selecting a device profile, click `Create` to complete the form



The device profile includes the device's `type` , `network mode` , `network server used` , `LoRaWAN MAC version` , and `parameter revision version` .

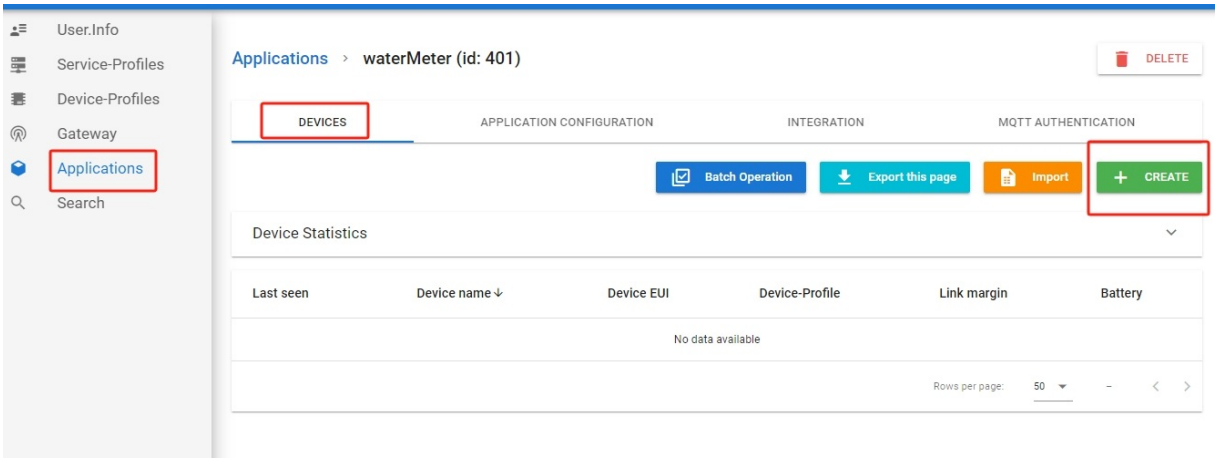
Copyright © AdoModel Tech Inc. All rights reserved. 2026-09-11 15:17:21

- Create a Device and activate

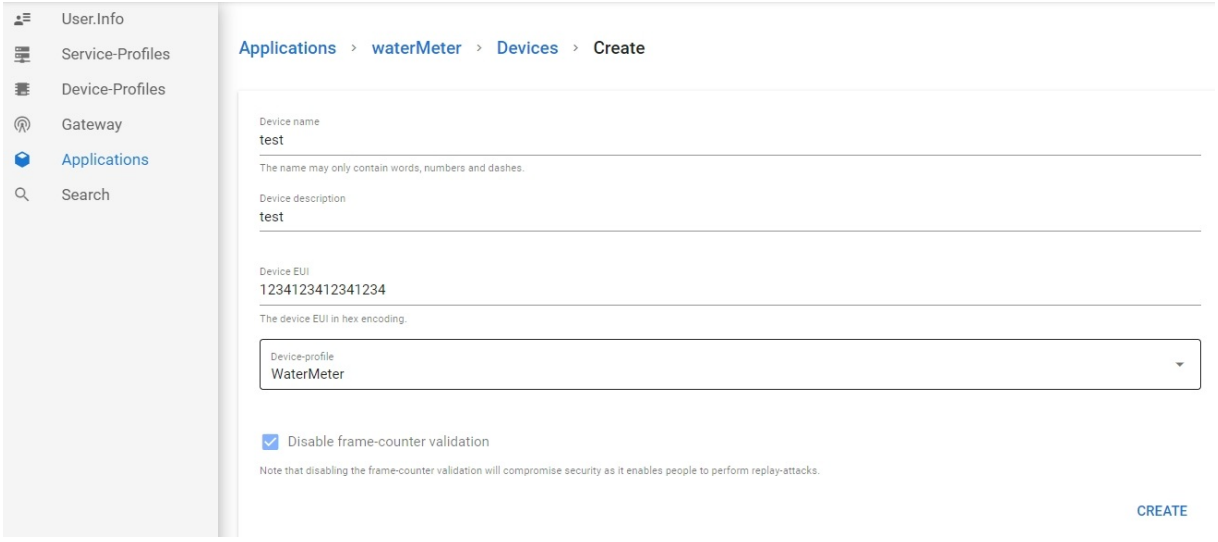
Create a Device and activate

After preparing the application and device Profile, the creation of the device, also known as the terminal node, can proceed.

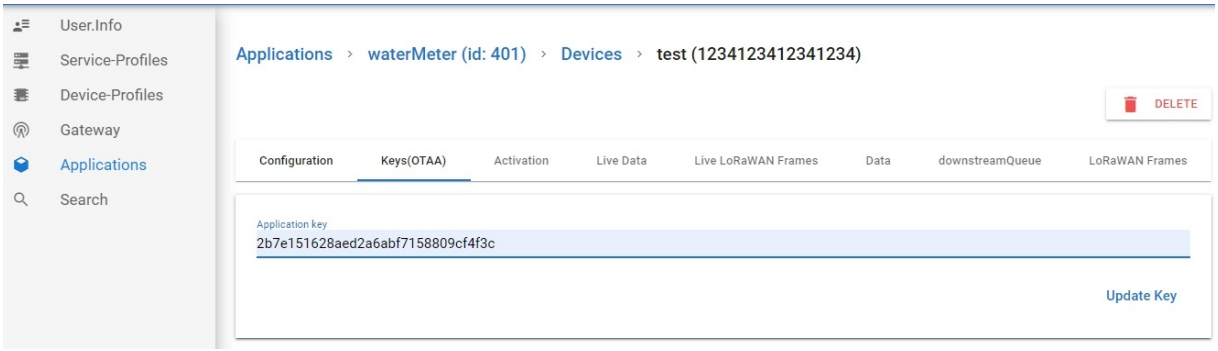
1. Enter the application list menu and select to enter an application from the application list.
2. In the application, the default first displayed node list is the current application. Click the add node button to enter the add form.



3. Based on the form input, select the required device Profile and complete the creation.



4. After creation, it will default to the internal details of the device. Depending on the type of device, Service Profile can be performed. If it is OTAA, fill in the 'application key' in the 'Device Key (OTAA)' tab, and click 'Update Key' button. If it is ABP, enter the 'Activation Status' tab for manual activation.



After activation, the registration of a device on the platform is completed.

Copyright © AdoModel Tech Inc. All rights reserved. 2026-09-11 15:17:22

- Observing live data
  - Observe real-time application data of devices
  - Observing Live LoRaWAN frames of devices

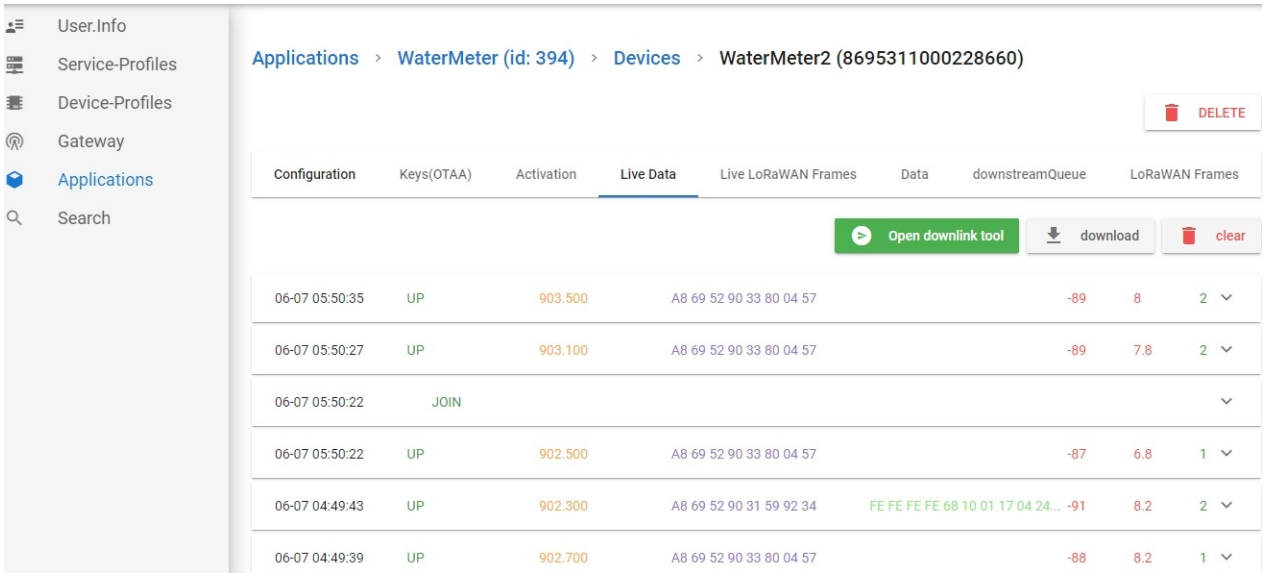
Observing live data

After completing device registration, start the device and ensure that the gateway is online, so that it can be viewed in the live data window.

Observe real-time application data of devices

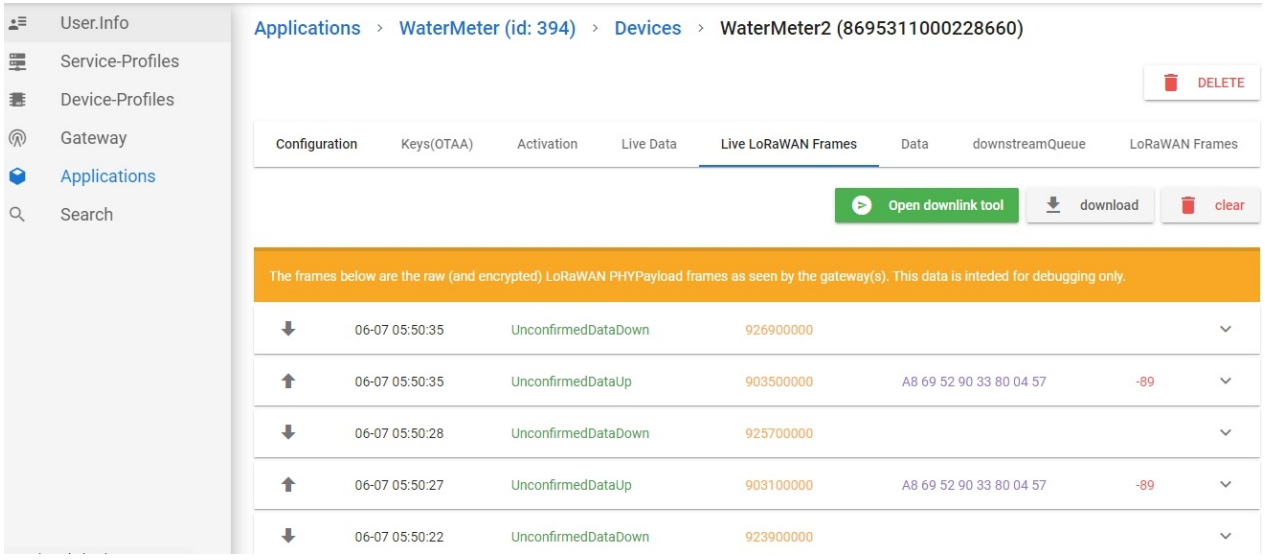
The device application data is the actual decoded data, and the detailed structure can be found in the open API.

You can enter the `Live Data` tab to view it.



Observing Live LoRaWAN frames of devices

The LoRaWAN frame contains uplink and downlink packets, which are the original frame data. The application data is not decrypted, so it can only be used for network debugging. You can view it by entering the `Live LoRaWAN data frames` tab.



Copyright © AdoModel Tech Inc. All rights reserved. 2026-09-11 15:17:21

- Network service configuration

Network service configuration

The network service configuration function is used to distinguish the network servers that users can use. One network service configuration corresponds to one type of network server on the platform, such as US_915.

The network service configuration includes the following attributes:

- **name** The name used for display and identification is only allowed in English and underlined
- **Network Server:** Once the selected `network server` is created, the application configured using this network service will only receive data from the corresponding `network server` , and the user may choose the corresponding network server when adding a `gateway` or `device configuration` .
- **Device status request frequency:** The platform will use the mac command devStatusReq specified in the LoRaWAN protocol to request the status of the device. The device will return the device's battery level and signal-to-noise ratio based on the request and store them in the platform's device properties. When set to 0, it means the platform does not send requests.
- **Minimum and maximum allowed rates** This parameter is used to set the maximum and minimum speed limit that the server can use when sending ADRs to nodes. 0 is not set, and the platform defaults to using it

Copyright © AdoModel Tech Inc. All rights reserved. 2026-09-11 15:17:28

- Device configuration

Device configuration

On the platform, a configuration abstraction can be created to represent a type of device, which is called 'device configuration'. After creating a device configuration, the actual device inherits its parameters.

The main function of device configuration is to specify the 'network server', 'network access mode', and 'type (Class A/Class C)' used by the device.

The main attributes of device configuration include:

- Name: Name of the device
- Network Server: The network server used
- LoRaWAN MAC Version: The MAC version in the LoRaWAN specification, e.g., 1.0.2 used by our company's end devices
- LoRaWAN Regional Parameters Version: The regional parameters revision in the LoRaWAN specification, e.g., Rev B used by our company's end devices

Other attributes should be filled in according to the prompts and actual needs.

Parameter description for ABP mode:

- When ABP mode is selected, the platform allows manual configuration of certain parameters, primarily RF `RX window parameters`. When the device is activated via ABP, it will `initialize` using these parameters. In practice, after the device communicates with the server, the server compares these configurations with the `network server parameters`. If there is a discrepancy, the server will issue a `MAC command` to notify the device to update them. Upon responding, the device will switch to using the `network server parameters`.
- On the platform, the `network server parameters` strictly adhere to the default values specified in the LoRaWAN standard. Especially on the public cloud platform, AdoModel will not make special modifications. For private cloud deployments, users can configure and deploy using their own custom parameters as needed.
- The platform does not provide a feature to directly modify `network server parameters`; these can only be added or updated via deployment configurations.
- In OTAA mode, these parameters are automatically configured during the join procedure. When the server sends the Join Accept message, it conveys the `network server parameters` to the device.

Network access mode:

- OTAA
- ABP

Class support:

- Class A

Class C

Copyright © AdoModel Tech Inc. All rights reserved. 2026-09-11 15:17:28

- Application

Application

Applications are the carriers of platform devices, and all devices belong to a certain application. In the application, there are mainly the following contents:

1. Device List

In the device list, you can view the current node list in the application and add new nodes

2. Application Settings

Modify the basic information of the application, such as description, name, and key, modify the [Load Data Decoder] (/feature/code.md)

3. Data push

In the platform, each application can set up an HTTP push interface. Once a device uploads data within the application, the platform will make a POST request based on the endpoint set in the application, and transfer the device data to a third-party server.

4. MQTT authentication

The platform supports API access to the MQTT protocol (details can be found in the open API), and the interface permission configuration of MQTT is achieved by creating MQTT passwords within the application. An application can create multiple MQTT passwords for different needs to connect.

Copyright © AdoModel Tech Inc. All rights reserved. 2026-09-11 15:17:31

- Custom codec
  - Simple Example

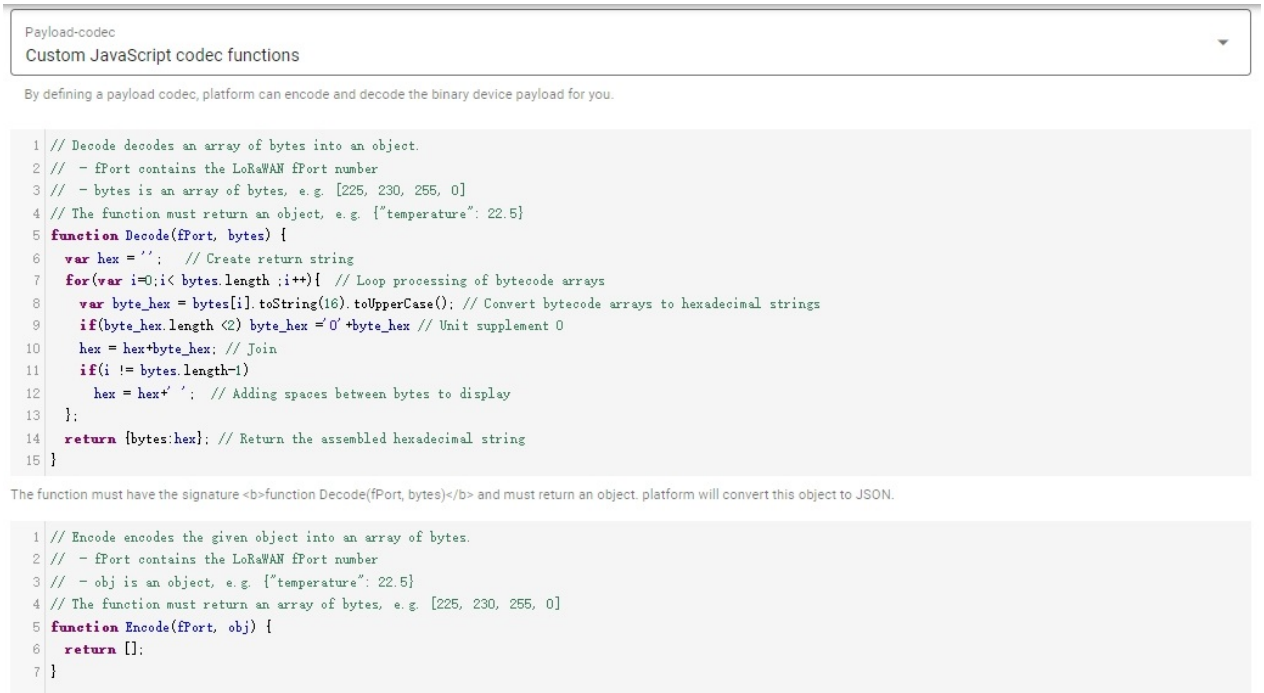
Custom codec

Codec is a basic feature provided by the platform. By default, the data uploaded by the device is the original base64 encoded string, which needs to be parsed by the user at the application level. In addition, if a custom codec is set on the platform application, the device will add fields to the data after reporting the data:

```
{
    ...
    "object": {                    // Decoded JSON object (filled in when the application has set the codec)
        "temperatureSensor": {"1": 25},
        "humiditySensor": {"1": 32}
    }
    ...
}
```

Specific data interfaces can refer to open APIs

The encoder provided by the platform consists of two custom JavaScript functions that need to be modified in the application settings:



You can achieve parsing and encoding of different `object` objects through custom code writing

In the decoding function `Decode (fPort bytes)`, bytes are the byte code array of device data. For example, `[255255255]` represents the byte code data `FF FF FF`. In the function body, by processing the fPort and bytes parameters, a JavaScript object is finally returned to complete the decoding setting. For more details, please refer to the JavaScript language.

In the encoding function `Encode (fPort, obj)`, on the contrary, when a user descends data through an API interface at the application level, if they carry an object object, the object will be encoded through this encoding function and ultimately return a bytecode array.

Note: The JavaScript versions supported by the decoder are below **ES5** . Please do not use any syntax above ES6, such as `const` , `let` , `for-in` , `for-of` , etc

Simple Example

As shown in the screenshot of the decoding function:

```
function Decode(fPort, bytes) {
  var hex = '';        // Create return string
  for(var i=0;i< bytes.length ;i++){    // Loop processing of bytecode arrays
    var byte_hex = bytes[i].toString(16).toUpperCase();    // Convert bytecode arrays to hexadecimal strings
    if(byte_hex.length <2) byte_hex ='0'+byte_hex    // Unit supplement 0
      hex = hex+byte_hex;    // Join
    if(i != bytes.length-1)
      hex = hex+' ';    // Adding spaces between bytes to display
  };
  return {bytes:hex};    // Return the assembled hexadecimal string
}
```

The function returns the hexadecimal string of the data uploaded by the device, as shown in the following figure:

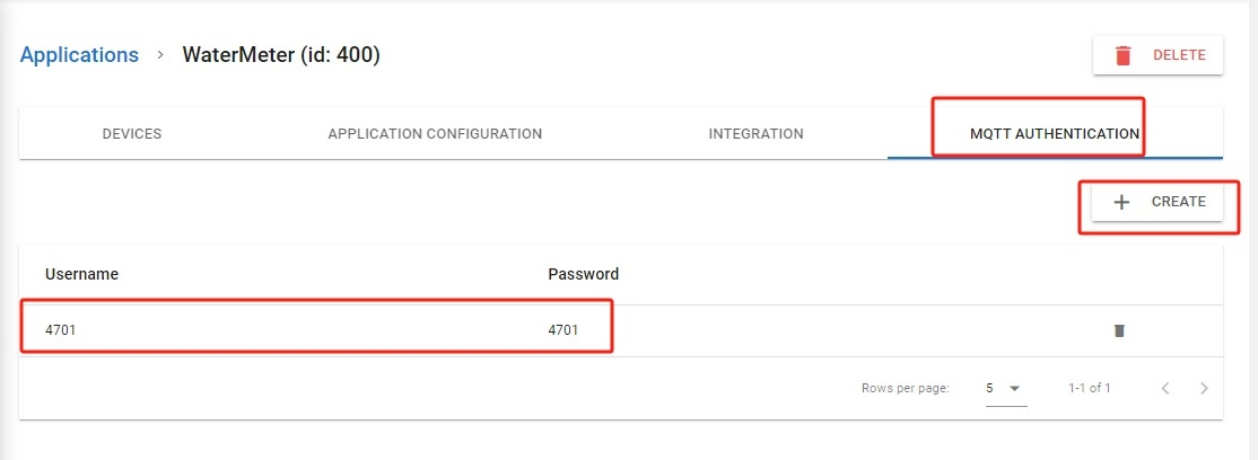


Copyright © AdoModel Tech Inc. All rights reserved. 2026-09-11 15:17:29

- MQTT Authorize

MQTT Authorize

The MQTT interface of the platform is authenticated through the application, and users must create an MQTT connection password in the application now in order to use the MQTT interface:



After creation, using the MQTT client to create a connection requires adding corresponding authentication information, such as:

```
mqtt://4701:4701@adomodelcloud.com:1883
```

please determine the account password transmission method for each MQTT SDK by yourself

Copyright © AdoModel Tech Inc. All rights reserved. 2026-09-11 15:17:28

- Device

Device

A device is a model used to bind actual node devices, which is used to record the identity of the device in the LoRaWAN network and store network access sessions. The main attributes of the device are:

- Name: The name of the identification
- Description: Device Description
- DevEUI: Unique identifier of a node in the LoRaWAN network
- Device configuration: refers to the specific network parameter configuration of the device
- Ignoring frame count verification: Does the platform ignore frame count verification for this device

  Note: Turning off frame count verification may bring the following security issues:

  - The device may receive messages from multiple gateways with the same fCnt, which may have a certain impact on the reception of application data (due to the LoRaWAN gateway broadcast mechanism)
  - The device may be vulnerable to replay attacks

The main functions in the device:

- Device Key: When the device is OTAA, the appKey can be configured
- Activation status: Check the activation status of the current device. If it is an ABP device, you can directly modify the activation status and manually (re) activate it
- Live data: Real time event data of the current device can be viewed on the UI, and detailed information can be viewed on the open API
- Live LoRaWAN data frame: The real-time LoRaWAN data frame of the current device can be viewed on the UI. Please note that the data within this data frame has not been decrypted, and this function is only for development and debugging purposes.
- Historical Data: You can view the historical data of several events such as `upload` , `ack` , `error` , and `join` on the device in the Historical Data tab, and provide API access
- Downlink testing: Can perform data downlink on devices in the `Downlink Tool` in the `Live Data` and `Live LoRaWAN Frames` windows

Copyright © AdoModel Tech Inc. All rights reserved. 2026-09-11 15:17:28

- Devices import and export
  - 1. Import/Export file format
  - 2. Import
  - 3. Export

Devices import and export

In order to facilitate users to import multiple devices, the platform encapsulates batch operation functions on the UI, which includes device export and import functions under **single application**. Users can use the factory exported device list to store in a file of `. xls` or `. xlsx`, or import through a device list made in their own format.

1. Import/Export file format

number	DevEUI	AppEUI	AppKEY	DevAddr	AppSKey	NwkSKey	name	desc
1	8695333022217510	0000000000000000	2b7e151628aed2a6abf7158809cf4f3c				8695333022217510	8695333022217510
2	8695333021104820	0000000000000000	2b7e151628aed2a6abf7158809cf4f3c				8695333021104820	8695333021104820
3	8695333021104640	0000000000000000	2b7e151628aed2a6abf7158809cf4f3c				8695333021104640	8695333021104640
4	8695333221104640	0000000000000000	2b7e151628aed2a6abf7158809cf4f3c				8695333021104640	8695333021104640
5	8695333021095060	0000000000000000	2b7e151628aed2a6abf7158809cf4f3c				8695333021095060	8695333021095060
6	8695443022217510	0000000000000000	2b7e151628aed2a6abf7158809cf4f3c				8695443022217510	8695443022217510
7	8695443021104820	0000000000000000	2b7e151628aed2a6abf7158809cf4f3c				8695443021104820	8695443021104820
8	8695443021104640	0000000000000000	2b7e151628aed2a6abf7158809cf4f3c				8695443021104640	8695443021104640
9	8695443221104640	0000000000000000	2b7e151628aed2a6abf7158809cf4f3c				8695443021104640	8695443021104640
10	8695443021095060	0000000000000000	2b7e151628aed2a6abf7158809cf4f3c				8695443021095060	8695443021095060

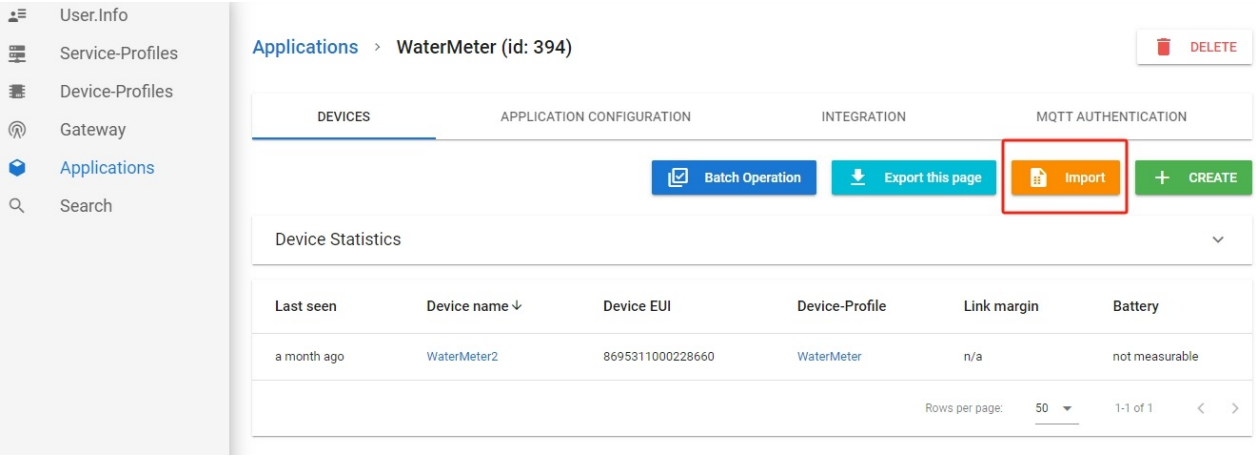
As file format in the figure

Field	description	must
number	serial number **cannot exist same serial number**	true
DevEUI	device EUI	true
AppEUI	AppEUIit is a field that has no effect temporarily, reserved	false
AppKEY	The appKey used by OTAA. If the device is about to access the network in OTAA mode, fill it in, otherwise the activation will fail	false
DevAddr	DevAddr in ABP mode, if the device is about to connect to the network in ABP mode, fill it in, otherwise the activation will fail	false
AppSKey	AppSKey in ABP mode, if the device is about to connect to the network in ABP mode, fill it in, otherwise the activation will fail	false
NwkSKey	NwkSKey in ABP mode, if the device is about to connect to the network in ABP mode, fill it in, otherwise the activation will fail	false
name	Device name in the platform, **will be named as devEUI if not fill.**	false
desc	description of the device	false

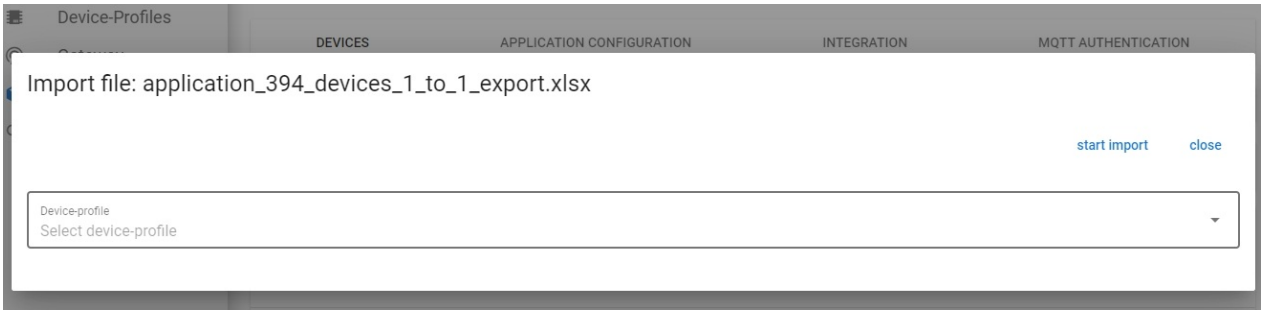
Note: In the list file, the `OTAA` or `ABP` modes are not distinguished, and mode selection can only be made by selecting `Device profile` during import. The UI will automatically add, activate (ABP), or update keys (OTAA) based on the parameters in the list.

2. Import

First, enter the application and on the device list page, you can see the import and export menus



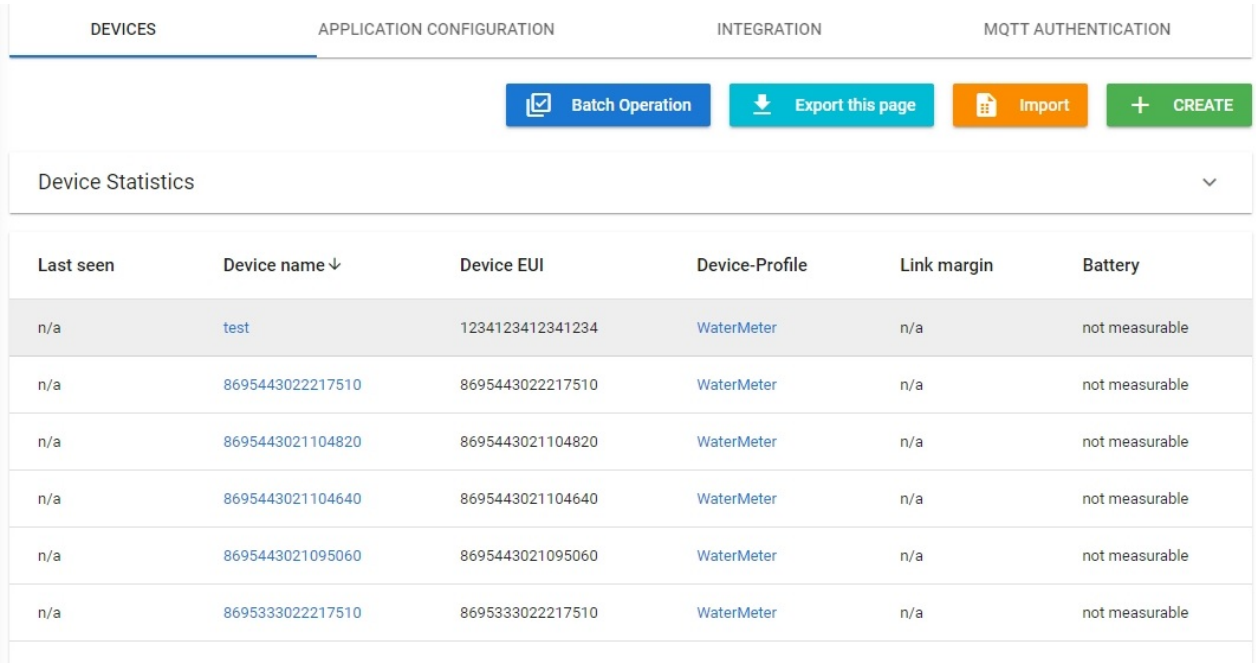
Click on `Import` and select the uploaded `xls` or `xlsx` file. The UI will directly read the file and an import window will pop up:



In the window, select a `Device profile` to determine the node configuration of the imported device. You can select some devices and click on `Start Import` to complete the operation. The UI will automatically complete the operation:

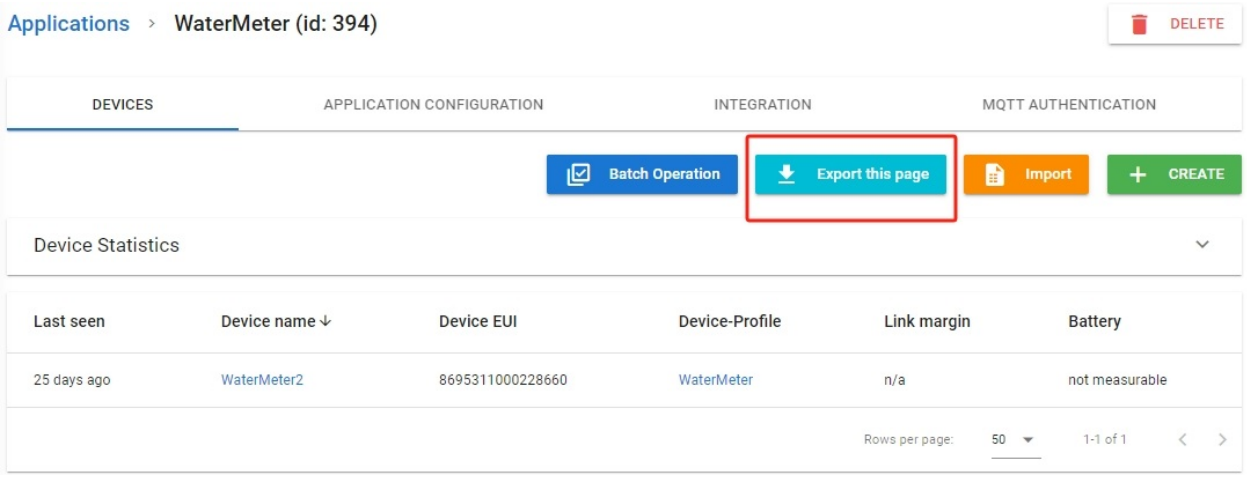


After successfully import, refresh the UI or go back to Application menu and select related application, imported devices can display:



3. Export

Provide export of `current page` devices in the device list view under the application. You can export more devices at once by selecting **to display more devices on a single page**



Click `Export Current Page`, the platform will automatically generate a table file for download.

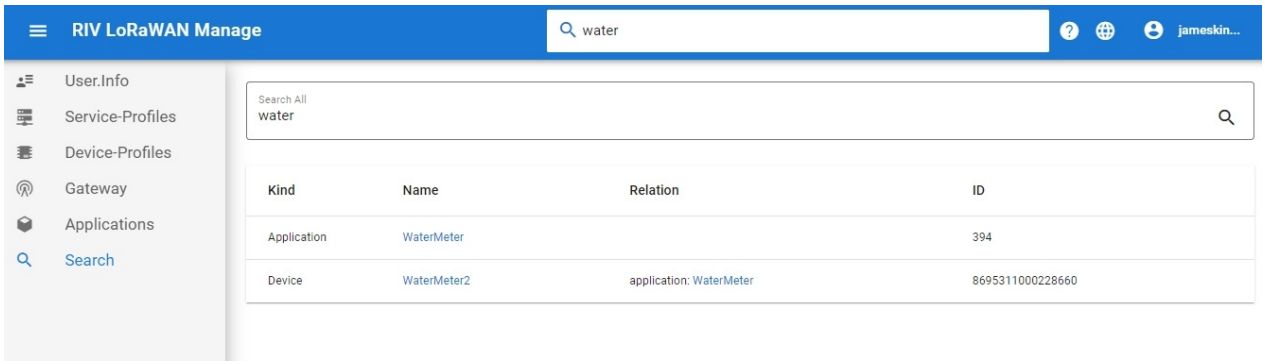
Note: If the device is currently connected to OTAA network, the parameters related to ABP activation in the exported list will be empty. If it is in ABP mode, the 'appKey' will be empty

Copyright © AdoModel Tech Inc. All rights reserved. 2026-09-11 15:17:28

- Global Search

Global Search

On the platform, users can directly search for `applications` , `devices` , `gateways` , and `users (only administrators)` through keywords, and the results will be presented in a table with their relevant connections displayed.



Copyright © AdoModel Tech Inc. All rights reserved. 2026-09-11 15:17:28

- Gateway

Gateway

The addition of a gateway on the platform is only for management and monitoring purposes, and cannot actually control the gateway. The gateway is identified by the gateway ID (MAC address, defined by the packet forwarder software within the gateway) and the network server used by the gateway.

Function of gateway

- Check gateway status
- Modify gateway parameters
- Gateway LoRaWAN data frame: All data frames received and sent by the gateway will be displayed. Please note that since the node identification in the LoRaWAN data frame is not devEUI but the device address code, the message will not contain devEUI.

Gateway access platform

At present, the platform provides support for 5 frequency bands by default. The gateway needs to modify the configuration file in the packet forwarder and restart it to complete gateway access:

```
// global_conf.json / local_conf.json
{
    ...
     "gateway_conf": {
        "gateway_ID": "0102030405060708",
        "server_address": "adomodelcloud.com",
        "serv_port_up": 915,
        "serv_port_down": 915,
      ...
    }
    ...
}
```

The default ports for the frequency bands supported by the platform are:

Frequency	Port
CN_470-510	470
EU_433	433
EU_863-870	868
US_902-928	915
AS_923	923
AU_915	1915
IN_865	865

Private deployment can be changed on its own.

Copyright © AdoModel Tech Inc. All rights reserved. 2026-09-11 15:17:28

- Users

Users

As a model for identifying different organizations on the platform, users can modify their user name and organization name within the platform, as well as view the system usage limit and validity period of the current user. They can also modify their login password on their own.

At the same time, in open APIs, API authentication is also obtained through the user's account and password.



The system usage and validity of users are configured by the platform administrator. If the usage is insufficient or exceeds the validity period, you can contact the system administrator.

Copyright © AdoModel Tech Inc. All rights reserved. 2026-09-11 15:17:28

- Data downlink description

Data downlink description

The platform performs LoRaWAN data downlink through RESTful or MQTT interfaces, and provides corresponding device downlink testing tool on the UI. However, for LoRaWAN downlink, there is a fixed logical specification. When calling the downlink interface on the platform, the actual data is not actually issued. The platform caches the command, calculates the downlink timing according to the rules of LoRaWAN device types, and only sends the data packet to the gateway at the appropriate time.

Class A situation:

The basic rule of class A is that the receiving window will only open after uplink

Send data to the platform, cache the data in the queue, and wait for the device to upload. After the device uploads data once, the platform will actually send this command from the queue to the gateway. Multiple data are continuously sent to the platform, and the platform caches the data to a queue and waits for the device to upload. After the device uploads data once, the platform sends the earliest command from the queue to the gateway, identifies pending, and requests the device to continue uploading to receive subsequent data.

Class C situation:

Class C is ready for downlink at any time, so the platform will directly distribute the data. Additionally, if the distribution type is confirmation required, the platform will temporarily cache this message until it receives confirmation from the node, or clear it according to the response timeout setting, and finally trigger the ACK event data.

Attention:

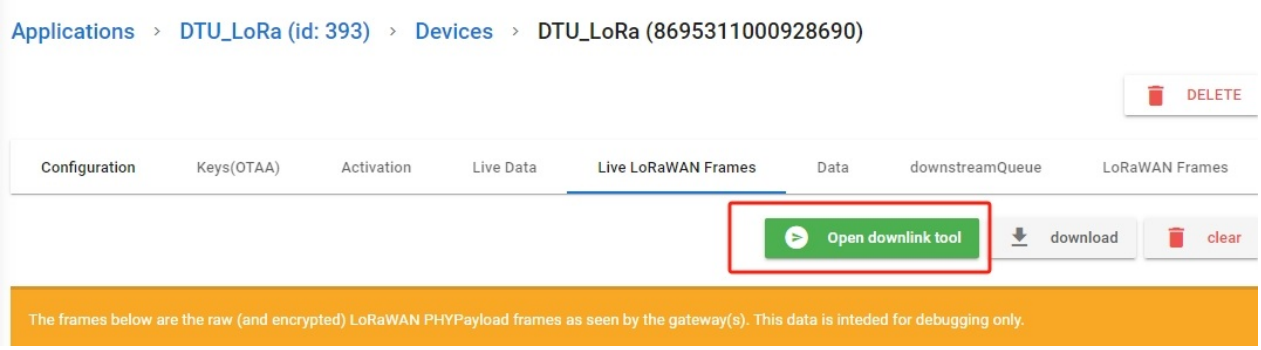
LoRaWAN products themselves are not suitable for real-time and frequent downlink control. There are many factors in the actual network that can affect stability. When designing LoRaWAN applications, careful consideration should be given to the downlink and uplink processes.

Copyright © AdoModel Tech Inc. All rights reserved. 2026-09-11 15:17:27

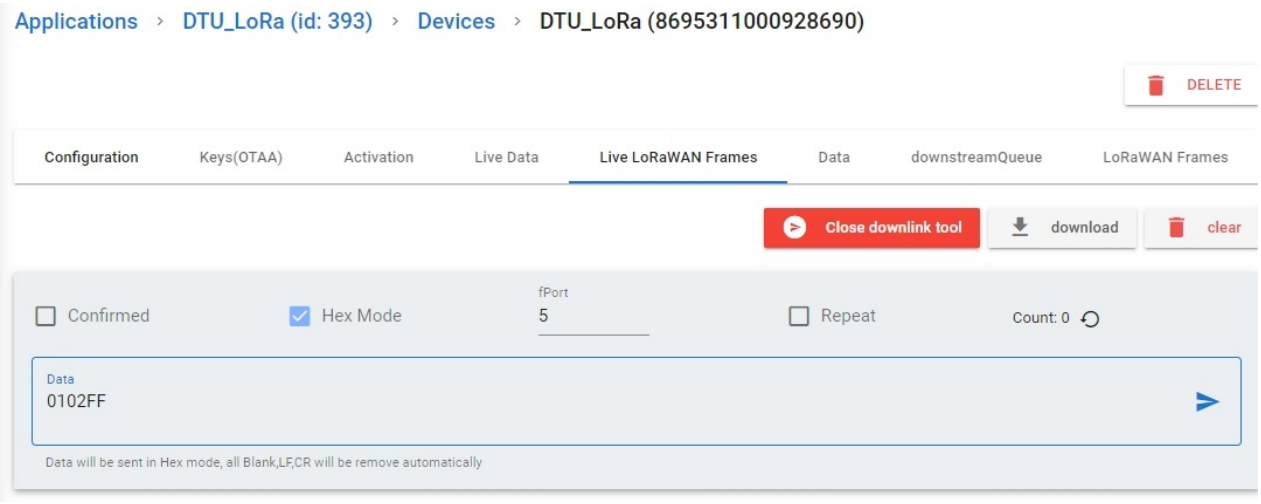
- Downlink Test Tool

Downlink Test Tool

In `Live Data` and `Live LoRaWAN Frames` tab of Deviceyou can open downlink tool



Fill parameters and data according to requirement, then the server will schedule a downlink packet to the device:



Parameters

- Confirmed: Refers to whether the scheduled downlink packet is a confirmed type packet in LoRaWAN
- Hex mode: The server-side API allows the use of 'string' or 'hex' mode for downlink data. When in hex mode, only strings in hex format are allowed to be input, which can refer to the open API
- fPortRefers to the fPort field in LoRaWAN packets
- Repeat: The tool allows setting a `loop interval` for loop issuance testing

Copyright © AdoModel Tech Inc. All rights reserved. 2026-09-11 15:17:27