



# Milesight Gateway

## UG56/UG65/UG67

### HTTP API Specification

Date	Applicable Firmware Version	Description
---	---	Initial version
Aug. 11, 2023	60.0.0.42/56.0.0.3	Add packet forwarder, NS-general, etc.
May 28, 2024	60.0.0.42-r5/56.0.0.4	Update login account information
Apr. 14, 2025	60.0.0.45/56.0.0.6	Add FUOTA
July 15, 2025	60.0.0.46-r2/56.0.0.6-r2	Update API login info, add BACnet server settings

## Contents

Overview .....	5
1. API Authentication .....	5
1.1 API Login .....	5
1.2 Change User Password .....	7
2. Packet Forwarder .....	8
2.1 Get Packet Forwarder Configuration .....	8
2.2 Set Packet Forwarder Configuration .....	11
2.3 Delete Packet Forwarder Retransmission Data .....	14
3. Network Server-General .....	15
3.1 Get General Configuration .....	15
3.2 Set General Configuration .....	16
4. Application .....	18
4.1 Get Applications .....	18
4.2 Create Application .....	20
4.3 Update Application .....	21
4.4 Delete Application .....	22
4.5 Get Data Transmission Integration .....	23
4.6 Create Data Transmission Integration .....	26
4.7 Update Data Transmission Integration .....	32
4.8 Delete Data Transmission Integration .....	34
5. Payload Codec .....	35
5.1 Get All Payload Codec Content .....	35
5.2 Get Payload Codec ID and Name .....	37
5.3 Get Payload Codec ID by Device EUI .....	38
5.4 Get Payload Codec by ID .....	40
5.5 Create a Custom Payload Codec .....	42
5.6 Update a Custom Payload Codec .....	45
5.7 Delete a Custom Payload Codec .....	49
5.8 Payload Codec Test .....	49
5.9 Get Payload Codec Setting .....	52

5.10 Update Payload Codec Online .....	53
5.11 Get Payload Codec Local Import Result .....	54
6. Profile .....	55
6.1 Get Profile .....	55
6.2 Create Profile .....	58
6.3 Update Profile .....	60
6.4 Delete Profile .....	61
7. Device .....	62
7.1 Get Device .....	62
7.2 Create Device .....	66
7.3 Update Device .....	67
7.4 Delete Device .....	69
8. Multicast Group .....	70
8.1 Get Multicast Group .....	70
8.2 Get Multicast Group by ID .....	72
8.3 Create Multicast Group .....	73
8.4 Update Multicast Group .....	75
8.5 Delete Multicast Group .....	76
8.6 Get Devices in Multicast Group .....	77
8.7 Add a Device to Multicast Group .....	78
8.8 Add Multiple Devices to Multicast Group .....	79
8.9 Remove Devices from Multicast Group .....	80
8.10 Remove a Device from Multicast Group by Device EUI .....	81
9. Gateway Fleet .....	82
9.1 Get Gateway List .....	82
9.2 Get Gateway by Gateway ID .....	83
9.3 Create Gateway .....	85
9.4 Update Gateway .....	86
9.5 Delete Gateway .....	87
10. Packets .....	88
10.1 Get Packets .....	88
10.2 Delete All Stored Packets .....	90

11. Send Downlink .....	91
11.1 Send Downlink Command to Device .....	91
11.2 Get Device Downlink Command in Queue .....	92
11.3 Delete Device Downlink Command from Queue .....	93
11.4 Send Downlink Command to Multicast Group .....	94
11.5 Get Multicast Downlink Command in Queue .....	95
11.6 Delete Multicast Downlink Command from Queue .....	95
12. FUOTA .....	96
12.1 Add FUOTA Task .....	96
12.2 Get FUOTA Task List .....	99
12.3 Update FUOTA Task .....	103
12.4 Delete FUOTA Task .....	106
12.5 Retry FUOTA Task .....	107
12.6 Get Payload Size of Datarate .....	108
12.7 Get Device List from FUOTA Task .....	109
12.8 Get Device List from Firmware Info .....	110
12.9 Get Official Firmware Model List .....	113
12.10 Get Official Firmware Info .....	113
12.11 Return Code List .....	115
13. BACnet Server .....	116
13.1 Get BACnet Server Configuration .....	116
13.2 Set BACnet Server Configuration .....	117
13.3 Get Available BACnet Object List .....	119
13.4 Get Existed BACnet Object List .....	123
13.5 Add BACnet Object .....	127
Appendix .....	131
Return Code List .....	131
Default RX2/Ping Slot Settings .....	131

# Overview

This document specifies the parameters and configuration files for Milesight gateways. Each API call needs to be accompanied with **Bearer Token** and this token can be required by [Login API](#). This token is only valid in 24 hours and need to require again after 24 hours.

The screenshot shows the 'Auth' tab selected in a REST client interface. The 'Type' dropdown is set to 'Bearer Token'. The 'Token' field contains a long alphanumeric string: 'slnVzZXJuYW1lIjoieYWRtaW4ifQ.xqkCofVboZ81bxmWdNjnUolaMfj1BqpR6e0X0z2-upE'.

## 1. API Authentication

### 1.1 API Login

**Request Method:** POST

**Request Address:**

`https://{gatewayIP}:8080/api/internal/login`

#### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

Body

Parameter	Type	Required	Description
username	string	Yes	60.0.0.42-r4 and before: apiuser 60.0.0.42-r5/56.0.0.4 and later: admin 60.0.0.46-r2/56.0.0.6-r2 and later: HTTP API account username
password	string	Yes	HTTP API login password (encrypted)

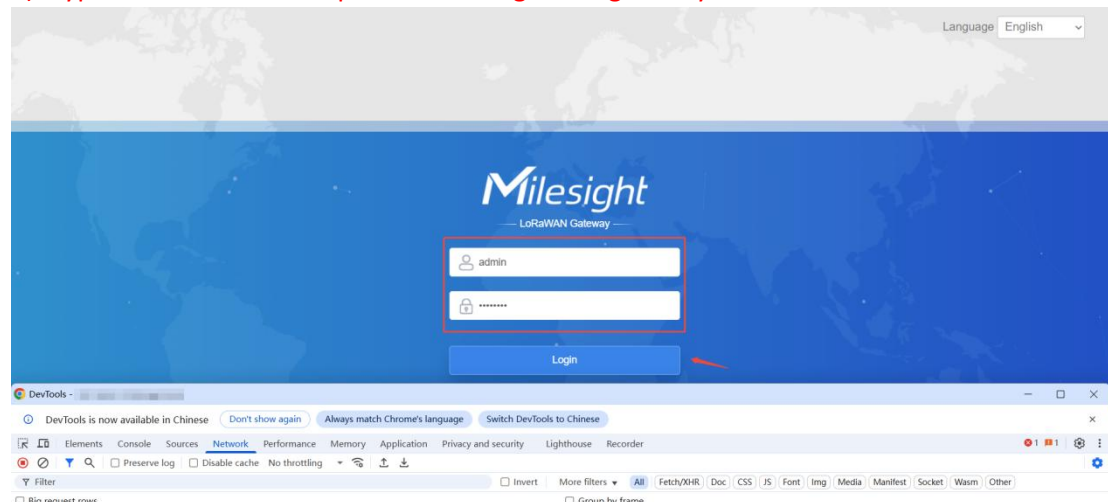
#### Response Parameters

Parameter	Type	Description
-----------	------	-------------

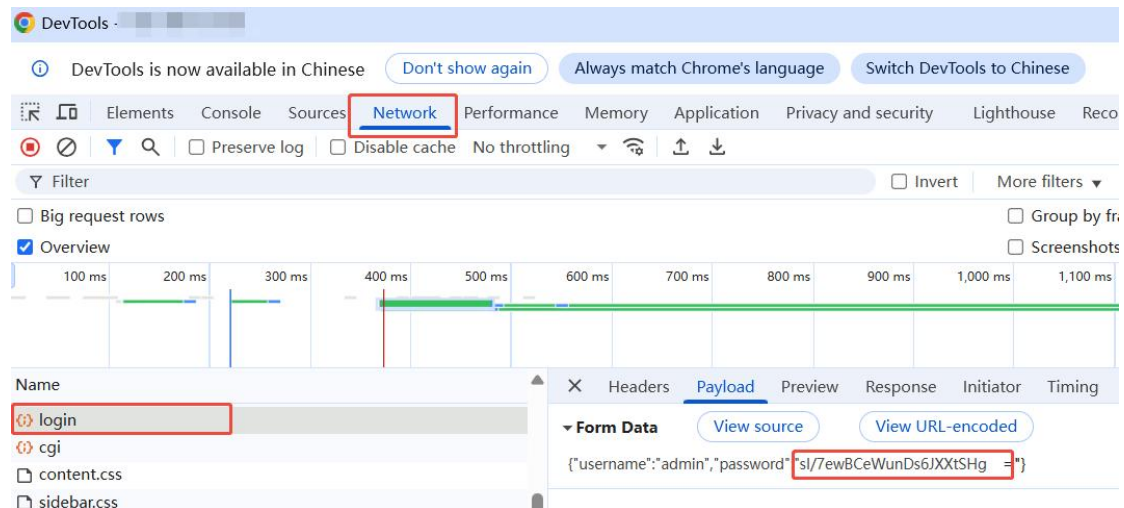
jwt	String	Token used for subsequent requests when success.
error	String	Error content
code	Integer	Error code

**Note:** for firmware 60.0.0.42-r5/56.0.0.4 and later, the encrypted password can be found via below steps:

- 1) Open the web GUI of the gateway via any browser.
- 2) Open the Developer Tool of this browser.
- 3) Type the username and password to log in the gateway.



- 4) Find the password for HTTP API authentication in the Developer tool.



### Request Example:

Firmware version 60.0.0.42-r4/56.0.0.3-r1 and before:

```
{
  "username": "apiuser",
  "password": "password"
}
```

Firmware version 60.0.0.42-r5/56.0.0.4 and later:

```
{
```

```
{
  "username": "admin",
  "password": "sl/7ewBCeWunDs6JXXtSHg=="
}
```

### Response Example

```
{
  "jwt":
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiJsbnR5cC1zZXJ2ZXIiLCJleHAiOiJlY2O
  DY1NDE1NDAsImIzcyI6ImxvcmEtYXBwLXNlcnZlcilIm5iZiI6MTY4NjQ1NTE0MCwic3ViIjoidXNl
  cilslVzZXJlIjoiYXBpdXNlciJ9.-IzLE9vbglPH954LhpvhzB2r-D128iRGD4B1uQ68Uts"
}
```

## 1.2 Change User Password

**Note:** this API is used to change the password of apiuser account and only works with firmware version 60.0.0.42-r4/56.0.0.3-r1 and before.

**Request Method:** PUT

**Request Address:**

```
https://{gatewayIP}:8080/api/users/apiuser/password
```

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

Body

Parameter	Type	Required	Description
password	String	Yes	

### Response Parameters

**Success:**

```
{}
```

**Failure:**

Parameter	Type	Description
error	String	Error content
code	Integer	Error code

### Request Example

```
https://192.168.23.150:8080/api/users/apiuser/password
```

Body

```
{
  "password": "123456"
}
```

```
}
```

### Response Example

```
{}
```

## 2. Packet Forwarder

### 2.1 Get Packet Forwarder Configuration

**Request Method:** GET

**Request Address:**

```
https://{gatewayIP}:8080/api/packet-forwarder/network-servers
```

#### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

#### Response Parameters

Parameter	Type	Description
gatewayEui	String	The devEUI of gateway
gatewayId	String	The ID of gateway
radioType	String	The radio signal of gateway
loriotSupport	boolean	If this gateway supports Lorient: true or false
status	number	The connection status, 1: connected of at least one destination, 0 : all disconnected
dataRetransmission	boolean	Enable: true, disable: false
pendingData	number	The number of packets to be retransmitted
servs	object []	
id	Number	Id number of server destination ID
servEnabled	boolean	Enable: true, disable: false
servAddr	String	localhost or the IP address of network server
servType	String	The type of network server
servPortUp	Number	The port number of server uplink
servPortDown	Number	The port number of server downlink
servMqttPort	Number	The port number of MQTT for embedded NS
syncFreqPlan	boolean	Enable: true, disable: false
connected	Number	0: disconnected 1: connected
servGwKey	String	The key of gateway,keep as null
Region	String	Region parameter for Everynet sever



Band	String	Band parameter for Everynet sever
username	String	The email address to access the Everynet server
password	String	The password to access the Everynet server
CupsUri	String	CUPS URI for basic station server
TcUri	String	LNS URI for basic station server

### Response Example

```
{
  "gatewayEui": "24E124FFFEF0DE07",
  "gatewayId": "24E124FFFEF0DE07",
  "radioType": "",
  "loriotSupport": true,
  "status": 1,
  "dataRetransmission": false,
  "pendingData": 0,
  "servs":[
    {
      "id":0,
      "servEnabled":false,
      "servAddr":"localhost",
      "servType":"ursalink",
      "servPortUp":1700,
      "servPortDown":1700,
      "servMqttPort":1883,
      "syncFreqPlan":false,
      "connected":0,
      "servGwKey": "",
      "Region": "",
      "Band": "",
      "username": "",
      "password": "",
      "CupsUri": "",
      "TcUri": ""
    },
    {
      "id":1,
      "servEnabled":true,
      "servAddr":"ap4pro.loriot.io",
      "servType":"loriot",
      "servPortUp":1780,
      "servPortDown":1780,
      "servMqttPort":0,
      "syncFreqPlan":true,
      "connected":1,
      "servGwKey": "",
      "Region": ""
    }
  ]
}
```

```
"Band":"","  
"username":"","  
"password":"","  
"CupsUri":"","  
"TcUri":"","  
},  
{"id":2,  
"servEnabled":false,  
"servAddr":"eu1.cloud.thethings.network",  
"servType":"semtech",  
"servPortUp":1700,  
"servPortDown":1700,  
"servMqttPort":0,  
"syncFreqPlan":false,  
"connected":0,  
"servGwKey":"","  
"Region":"","  
"Band":"","  
"username":"","  
"password":"","  
"CupsUri":"","  
"TcUri":"","  
},  
{  
"id":3,  
"servEnabled":false,  
"servAddr":"192.168.45.12",  
"servType":"chirpstack_generic",  
"servPortUp":1700,  
"servPortDown":1700,  
"servMqttPort":18883,  
"syncFreqPlan":false,  
"connected":0,  
"servGwKey":"","  
"Region":"","  
"Band":"","  
"username":"","  
"password":"","  
"CupsUri":"","  
"TcUri":"","  
}  
{"id":4,  
"servEnabled":false,  
"servAddr":"","
```

```
"servType":"basicstation",
"servPortUp":0,
"servPortDown":0,
"servMqttPort":0,
"syncFreqPlan":false,
"connected":0,
"servGwKey": "",
"Region": "",
"Band": "",
"username": "",
"password": "",
"CupsUri": "",
"TcUri": "ws:\\\\192.168.45.135"
}
{
  id":5,
  "servEnabled":true,
  "servAddr":"localhost",
  "servType":"everynet",
  "servPortUp":1690,
  "servPortDown":1690,
  "servMqttPort":0,
  "syncFreqPlan":true,
  "connected":0,
  "servGwKey": "",
  "Region":"eu",
  "Band":"eu868",
  "username":"test@milesight.com",
  "password":"123456",
  "CupsUri": "",
  "TcUri": ""
}
]
```

## 2.2 Set Packet Forwarder Configuration

**Request Method:** POST

**Request Address:**

`https://{gatewayIP}:8080/api/packet-forwarder/network-servers`

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

## Body

Parameter	Type	Required	Description
dataRetransmission	boolean	Yes	Enable: true, disable: false
GatewayId	number	Yes	The gateway id
servs	object []	Yes	
id	Number	Yes	Id number of different network server
servEnabled	boolean	Yes	Enable: true, disable: false
servAddr	String	Yes	localhost or the IP address of network server
servType	String	Yes	Supported type of server. Embedded NS: ursalink Semtech: semtech Everynet: everynet Loriot: loriot
servPortUp	Number	Yes	The port number of server uplink, range 1-65535
servPortDown	Number	Yes	The port number of server downlink, range 1-65535
servMqttPort	Number	Yes	The MQTT port number of embedded NS, range 1-65535
syncFreqPlan	boolean	Yes	Enable: true, disable: false
servGwKey	String	Yes	The key of gateway, keep as null
Region	String	Yes	1) Keep as null when server type is not Everynet 2) When server type is Everynet, supported value: US(USA): us AP(ASIA Pacific): ap ATC(Brazil): atc custom: user defined
Band	String	Yes	1) Keep as null when server type is not Everynet 2) When server type is Everynet, supported value: EU868/US902/Australia915/China470/AS923/ SouthKorea920/China779/EU433/India865/R U864/AS923_2/AS923_3/AS923_4
username	String	Yes	The email address to access the Everynet server
password	String	Yes	The password to access the Everynet server

**Note: when adding a new packet forwarder destination, it is necessary to include current packet forwarder settings into request. Otherwise, the current settings will be overwritten by your new settings.**

## Response Parameters

### Success:

```
{}
```

### Failure:

Parameter	Type	Description
error	String	Error content
code	Integer	Error code

### Request Example:

```
{
  "gatewayId": "24E124FFFEF0DE07",
  "dataRetransmission": false,
  "servs":[
    {
      "id":0,
      "servEnabled":false,
      "servAddr":"localhost",
      "servType":"ursalink",
      "servPortUp":1700,
      "servPortDown":1700,
      "servMqttPort":1883,
      "syncFreqPlan":false,
      "connected":0,
      "servGwKey":"",
      "Region":"",
      "Band":"",
      "username":"",
      "password":""
    },
    {
      "id":1,
      "servEnabled":true,
      "servAddr":"ap4pro.loriot.io",
      "servType":"loriot",
      "servPortUp":1780,
      "servPortDown":1780,
      "servMqttPort":0,
      "syncFreqPlan":true,
      "connected":1,
      "servGwKey":"",
      "Region":"",
      "Band":"",
      "username":"",
      "password":""
    }
  ],
}
```

```
{
  "id":2,
  "servEnabled":false,
  "servAddr":"eu1.cloud.thethings.network",
  "servType":"semtech",
  "servPortUp":1700,
  "servPortDown":1700,
  "servMqttPort":0,
  "syncFreqPlan":false,
  "connected":0,
  "servGwKey":"",
  "Region":"",
  "Band":"",
  "username":"",
  "password":""
},
{
  "id":3,
  "servEnabled":true,
  "servAddr":"localhost",
  "servType":"everynet",
  "servPortUp":1690,
  "servPortDown":1690,
  "servMqttPort":0,
  "syncFreqPlan":true,
  "connected":0,
  "servGwKey":"",
  "Region":"eu",
  "Band":"eu868",
  "username":"test@milesight.com",
  "password":"123456"
}
}
```

## 2.3 Delete Packet Forwarder Retransmission Data

**Request Method:** DELETE

**Request Address:**

<https://{gatewayIP}:8080/api/packet-forwarder/network-servers/backupdata>

**Request Parameters**

Header

Parameter	Required	Description
Content-Type	Yes	application/json

**Response Parameters****Success:**

{}

**Failure:**

Parameter	Type	Description
error	String	Error content
code	Integer	Error code

## 3. Network Server-General

### 3.1 Get General Configuration

**Request Method:** GET**Request Address:**

https://{gatewayIP}:8080/api/network-server/settings

**Request Parameters**

Header

Parameter	Required	Description
Content-Type	Yes	application/json

**Response Parameters**

Parameter	Type	Description
ursalinkCloud	boolean	Whether platform mode is enabled
cloudPlatform	String	Platform mode type, null means Milesight IoT Cloud mode
smartOfficeUrl	String	Yeastar Workplace (on-premises) server address
ursalinkPktFwdEnable	boolean	Whether embedded NS option of packet forwarder is enabled
channelPlan	Number	3: CN470 0: EU868, 8: IN865, 9: RU864, 1: US915, 2: AU915, 4: AS923-1, 5: KR920, 10: AS923-2, 11: AS923-3, 12: AS923-4
channelMask	string	
netId	string	
joinDelay	Number	
rx1Delay	Number	

rx1DrOffset	Number	RX1 Datarate offset
rx2Dr	Number	RX2 Datarate
rx2Frequency	Number	
leaseTime	string	
logLevel	Number	0: panic, 1: fatal, 2: error, 3: warning, 4:info, 5: debug
enable	boolean	Whether NS is enabled
additionalPlan	Object []	
id	number	
freq	string	
maxDatarate	number	
minDatarate	number	

### Response Example

```
{
  "ursalinkCloud": false,
  "cloudPlatform": "",
  "smartOfficeUrl": "",
  "ursalinkPktFwdEnable": true,
  "channelPlan": 0,
  "channelMask": "",
  "netId": "010203",
  "joinDelay": 5,
  "rx1Delay": 1,
  "rx1DrOffset": 0,
  "rx2Dr": 0,
  "rx2Frequency": 869525000,
  "leaseTime": "8760-0-0",
  "logLevel": 4,
  "enable": true,
  "additionalPlan": []
}
```

## 3.2 Set General Configuration

**Request Method:** POST

**Request Address:**

<https://{gatewayIP}:8080/api/network-server/settings>

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json



## Body

Parameter	Type	Required	Description
enable	boolean	Yes	Whether enable NS
ursalinkCloud	boolean	Yes	Whether enable platform mode Enable: true, disable: false
cloudPlatform	String	Yes	Platform type: smart-office, smart-office-on-premise When this value is null, it means Milesight IoT Cloud mode
smartOfficeUrl	String	Yes	Yeastar Workplace (on-premises) server address
channelPlan	Number	Yes	3: CN470 0: EU868, 8: IN865, 9: RU864, 1: US915, 2: AU915, 4: AS923-1, 5: KR920, 10: AS923-2, 11: AS923-3, 12: AS923-4
channelMask	string	Yes	<b>Examples:</b> 1, 6: Enabling Channel 1 and Channel 6 1-7: Enabling Channel 1 to Channel 7 1-7, 15: Enabling Channel 1 to Channel 7 and Channel 15
netId	string	Yes	
joinDelay	Number	Yes	Range 1-255s
rx1Delay	Number	Yes	Range 1-15s
leaseTime	string	Yes	hh-mm-ss
logLevel	Number	Yes	0: panic, 1: fatal, 2: error, 3: warning, 4: info, 5: debug
additionalPlan	Object []	No	
id	number	Yes	
action	number		1: add, 2: modify, 3: delete
freq	string	Yes	
maxDatarate	number	Yes	0: DR0 (SF12, 125kHz) 1: DR0 (SF11, 125kHz) 2: DR0 (SF10, 125kHz) 3: DR0 (SF9, 125kHz) 4: DR0 (SF8, 125kHz) 5: DR0 (SF7, 125kHz) 6: DR0 (SF7, 250kHz)
minDatarate	number	Yes	0: DR0 (SF12, 125kHz) 1: DR0 (SF11, 125kHz) 2: DR0 (SF10, 125kHz) 3: DR0 (SF9, 125kHz) 4: DR0 (SF8, 125kHz) 5: DR0 (SF7, 125kHz) 6: DR0 (SF7, 250kHz)

## Response Parameters

Parameter	Type	Description
-----------	------	-------------

Error	string	When sending this message with success, it can only reply content "request accepted" and the network server program will reboot. User needs re-get the configuration to check the result.
Code	number	

### Request Example

```
{
  "enable": true,
  "ursalinkCloud": false,
  "cloudPlatform": "",
  "smartOfficeUrl": "",
  "channelPlan": 0,
  "channelMask": "",
  "netId": "010203",
  "joinDelay": 5,
  "rx1Delay": 1,
  "rx1DrOffset": 0,
  "rx2Dr": 0,
  "rx2Frequency": 869525000,
  "leaseTime": "8760-0-0",
  "logLevel": 4,
  "additionalPlan": []
}
```

### Response Example

```
{
  "error": "request accept",
  "code": 17
}
```

## 4. Application

### 4.1 Get Applications

**Request Method:** GET

**Request Address:**

<https://{gatewayIP}:8080/api/urapplications?limit=10&offset=0&organizationID=1>

Params

Parameter	Type	Required	Description
search	String	No	Application name
limit	String	Yes	Max number of applications to return in the result-test.
offset	String	Yes	Offset in the result-set (for pagination).
organizationID	String	Yes	1

## Request Parameters

### Header

Parameter	Required	Description
Content-Type	Yes	application/json

## Response Parameters

Parameter	Type	Description
totalCount	String	The total number of the application already created.
disable	Boolean	
result	Object[]	
id	String	The ID of the application.
name	String	The name of the application.
description	String	The the description of the application.
organizationID	String	
serviceProfileID	String	The unique ID of this service profile.
payloadCodec	String	Payload codec (invalid after firmware 60.0.0.41/56.0.0.3).
payloadDecoderScript	String	Payload decoder script(invalid after firmware 60.0.0.41/56.0.0.3).
payloadEncoderScript	String	Payload encoder script(invalid after firmware 60.0.0.41/56.0.0.3).
using	Boolean	Have device or not in this application.
kind	Array	Data transmission integrations associated with this application.

## Request Example

```
https://192.168.23.150:8080/api/urapplications?limit=10&offset=0&organizationID=1
```

## Response Example

```
{
  "totalCount": "1",
  "disable": false,
  "result": [
    {
      "id": "1",
```

```
{
  "name": "cloud",
  "description": "cloud",
  "organizationID": "1",
  "serviceProfileID": "f6f7d81d-647f-4c7f-8409-3e5218c0c523",
  "payloadCodec": "",
  "payloadEncoderScript": "",
  "payloadDecoderScript": "",
  "using": true,
  "kinds": [
    "URBACnetIP",
    "URHTTP",
    "URMQTT"
  ]
}
```

## 4.2 Create Application

**Request Method:** POST

**Request Address:**

https://{gatewayIP}:8080/api/urapplications

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

Body

Parameter	Type	Required	Description
name	String	Yes	User-define
description	String	Yes	User-define
organizationID	String	Yes	Default: 1
serviceProfileID	String	Yes	Default: f6f7d81d-647f-4c7f-8409-3e5218c0c523

### Response Parameters

Parameter	Type	Description
id	String	The new application id

### Request Example

```
{
  "name": "cloud1",
  "description": "cloud1",
  "organizationID": "1",
  "serviceProfileID": "f6f7d81d-647f-4c7f-8409-3e5218c0c523"
}
```

### Response Example

```
{
  "id": "6"
}
```

## 4.3 Update Application

**Request Method:** PUT

**Request Address:**

https://{gatewayIP}:8080/api/urapplications/{id}

Params

Parameter	Type	Required	Description
id	String	Yes	The ID of application

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

Body

Parameter	Type	Required	Description
name	String	Yes	User-define
description	String	Yes	User-define
organizationID	String	Yes	Default: 1
serviceProfileID	String	Yes	The unique ID of this service profile, it can be searched by get method

### Response Parameters

**Success:**

```
{}
```

**Failure:**

Parameter	Type	Description
error	String	Error content
code	Integer	Error code

**Request Example**

```
https://192.168.23.150:8080/api/urapplications/1
```

**Body**

```
{
  "name": "cloud1",
  "description": "cloud1",
  "organizationID": "1",
  "serviceProfileID": "f6f7d81d-647f-4c7f-8409-3e5218c0c523"
}
```

**Response Example**

```
{}
```

## 4.4 Delete Application

**Request Method:** DELETE**Request Address:**

```
https://{gatewayIP}:8080/api/urapplications/{id}
```

**Params**

Parameter	Type	Required	Description
id	String	Yes	The ID of application

**Request Parameters****Header**

Parameter	Required	Description
Content-Type	Yes	application/json

**Response Parameters****Success:**

```
{}
```

**Failure:**

Parameter	Type	Description
error	String	Error content
code	Integer	Error code

**Request Example**

```
https://192.168.23.150:8080/api/urapplications/6
```

**Response Example**

{}

## 4.5 Get Data Transmission Integration

**Method:** GET**Request Address:**

https://{gatewayIP}:8080/api/urapplications/{id}/integrations/{data-transmission-type}

Params

Parameter	Type	Required	Description
id	String	Yes	The ID of application
data-transmission-type	String	Yes	Option: http, mqtt <b>Note:</b> when getting a HTTPS integration, select http

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

### Response Parameters

**HTTP/HTTPS:**

Parameter	Type	Description
id	String	The ID of application
httpType	String	http or https
headers	Object[]	
key	String	
value	String	
dataUpURL	String	Uplink data URL
joinNotificationURL	String	Join Notification URL
ackNotificationURL	String	ACK Notification URL
errorNotificationURL	String	Error Notification URL

**MQTT:**

Parameter	Type	Description
id	String	The ID of application
host	String	Broker address
port	Integer	Broker port
useAuth	Boolean	Whether to enable user credentials
username	String	
password	String	
connectTimeout	Integer	Connection timeout, unit: s

keepAliveInterval	Integer	Keep alive interval, unit: s
clientID	String	Client ID
useTLS	Boolean	Whether to enable TLS
CACert	String	
CAName	String	
TLSCert	String	
certName	String	
TLSKey	String	
keyName	String	
uplinkTopic	String	Uplink data topic
upQoS	Integer	
joinTopic	String	Join notification topic
joinQoS	Integer	
ackTopic	String	ACK notification topic
ackQoS	Integer	
errorTopic	String	Error notification topic
errorQoS	Integer	
TLSMode	Integer	0 : CA signed certificates 1 : Self signed certificates
downlinkTopic	String	Downlink topic
downlinkQoS	Integer	
connectStatus	Boolean	Whether connected
mcDownlinkTopic	String	Multicast downlink data topic
mcDownlinkQoS	Integer	
requestTopic	String	Request data topic
requestQoS	Integer	
responseTopic	String	Response data topic
responseQoS	Integer	

### Request Example

<https://192.168.23.150:8080/api/urapplications/1/integrations/http>

### Response Example

#### HTTP:

```
{
  "id": "1",
  "headers": [],
  "dataUpURL": "http://192.168.1.1",
  "joinNotificationURL": "",
  "ackNotificationURL": "",
  "errorNotificationURL": "",
  "httpType": "http"
}
```



```
}
```

**MQTT:**

```
{  
  "id": "1",  
  "host": "broker.emqx.io",  
  "port": 8883,  
  "useAuth": false,  
  "username": "",  
  "password": "",  
  "connectTimeout": 30,  
  "keepAliveInterval": 60,  
  "clientId": "ug6545191",  
  "useTLS": true,  
  "CACert": "",  
  "CAName": "",  
  "TLSCert": "",  
  "certName": "",  
  "TLSKey": "",  
  "keyName": "",  
  "uplinkTopic": "",  
  "upQoS": 0,  
  "joinTopic": "",  
  "joinQoS": 0,  
  "ackTopic": "",  
  "ackQoS": 0,  
  "errorTopic": "",  
  "errorQoS": 0,  
  "TLSMode": 0,  
  "downlinkTopic": "",  
  "downlinkQoS": 0,  
  "connectStatus": true,  
  "mcDownlinkTopic": "",  
  "mcDownlinkQoS": 0,  
  "requestTopic": "",  
  "requestQoS": 0,  
  "responseTopic": "",  
  "responseQoS": 0  
}
```

## 4.6 Create Data Transmission Integration

**Method:** POST

### 1) MQTT

**Request Address:**

`https://{gatewayIP}:8080/api/urapplications/{id}/integrations/mqtt`

Params

Parameter	Type	Required	Description
id	String	Yes	The ID of application

**Note:** one application only support to add this type integration once.

**Request Parameters**

Header

Parameter	Required	Description
Content-Type	Yes	application/json

Body

Parameter	Type	Required	Description
host	String	No	Broker address
port	Integer	No	Broker port
useAuth	Boolean	No	Whether to enable user credentials
username	String	No	
password	String	No	
connectTimeout	Integer	No	Connection timeout, unit: s
keepAliveInterval	Integer	No	Keep alive interval, unit: s
clientId	String	No	A unique client ID
useTLS	Boolean	No	Whether to enable TLS
CACert	String	No	CA file content
CAName	String	No	CA file name
TLSCert	String	No	Client certificate file content
certName	String	No	Client certificate file name
TLSKey	String	No	Client key file content
keyName	String	No	Client certificate file name
uplinkTopic	String	No	Uplink data topic
upQoS	Integer	No	Option: 0,1,2
joinTopic	String	No	Join notification topic
joinQoS	Integer	No	Option: 0,1,2
ackTopic	String	No	ACK notification topic
ackQoS	Integer	No	Option: 0,1,2
errorTopic	String	No	Error notification topic

errorQoS	Integer	No	Option: 0,1,2
TLSMode	Integer	No	0 : CA signed certificates 1 : Self signed certificates
downlinkTopic	String	No	Downlink topic
downlinkQoS	Integer	No	Option: 0,1,2
mcDownlinkTopic	String	No	Multicast downlink data topic
mcDownlinkQoS	Integer	No	Option: 0,1,2
requestTopic	String	No	Request data topic
requestQoS	Integer	No	Option: 0,1,2
responseTopic	String	No	Response data topic
responseQoS	Integer	No	Option: 0,1,2 0: QoS 0 1: QoS 1 2: QoS 2

## Response Parameters

**Success:**

 $\{$ 

### Failure:

Parameter	Type	Description
error	String	Error content
code	Integer	Error code

## Request Example

<https://192.168.23.150:8080/api/urapplications/11/integrations/mqtt>

## Body

```
{
  "host": "broker.emqx.io",
  "port": 8883,
  "useAuth": false,
  "username": "",
  "password": "",
  "connectTimeout": 30,
  "keepAliveInterval": 60,
  "clientId": "ug6545191",
  "useTLS": true,
  "TLSMode": 1,
  "CACert": "-----BEGIN
CERTIFICATE-----\nMIIDQTCCAimgAwIBAgITBmyfz5m/jAo54vB4ikPmljZbyjANBgkqhkiG9w0B
AQsF\nnADA5MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQW1hem9uMRkwFwYDVQQDEx
BBbWF6\nnb24gUm9vdCBDQSAxMB4XDTE1MDUyNjAwMDAwMFoXDTE1MDUyNjAwMDAw
MFowOTEL\nnMAkGA1UEBhMCVVMxMDUyNjAwMDAwMFoXDTE1MDUyNjAwMDAwMFoXDTE1MDUyNjAwMDAw
m9uIFJv\nnb3QgQ0EgMTCCASlwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALJ4gHHKeN
```

```
Xj\nc9HgFB0fW7Y14h29Jlo91ghYPI0hAEvrAlhtOgQ3pOsQTNroBvo3bSMgHFzZM\n9O6II8
c 6zf1tRn4SWiw3te5djgdYZ6k/ol2peVKVuRF4fn9tBb6dNqcmzU5L/qw\nlIFAGbHrQgLKm
a/sRxmPUDgH3KKHOVj4utWp
UhnMJbulHheb4mjUcAwhmahRWa6\nVOujw5H5SNz/0egwLX0tdHA114gk957EWW67c4cX8j
JGKLhD
rcdqsq08p8kDi1L\n93FcXmn/6pUCyziKrlA4b9v7LWlBxcceVOF34GfID5yHI9Y/QCB/IIDeGEw
OyQm\nnjgSubJrlqg0CAwEAAaNCMEAwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBA
MC\nAAYYwHQYDVR0OBBYEFIQYzIU07LwMIJQuCFmcx7IQTgoIMA0GCSqGSib3DQEBcWUA\n
A4IBAQC8jdaQZChGsV2USggNiMOruYou6r4IK5lpDB/G/wkjUu0yKGX9rbxenDI\nnU5PMCCjJm
CXPI6T53iHTfIUJrU6adTrCC2qJeHZERxhlbl1BjJt/msv0tadQ1wUs\nnN
gDS63pYaACbvXy8MWy7Vu33PqUXHeeE6V/Uq2V8viTO96LXFvKWlJbYK8U90w\nno/ufQJVtM
VT8QtPHRh8jrdkPSHCA2XV4cdFyQzR1bldZwgJcJmApzyMZFo6IQ6XU\nn5Msl yMRQ
hDKXJioaldXgjUkK642M4UwtBV8ob2xJNDd2ZhwLnoQdeXeGADbkpy\nnrqXRfboQnoZsG4q5W
TP468SQvG5\n-----END CERTIFICATE-----",
  "CAName": "AmazonRootCA1 (4).pem",
  "TlSCert": "-----BEGIN
CERTIFICATE-----\nMIIDWjCCAKKgAwIBAgIVAPty/swo3PehJhh6CfCx12SsxShoMA0GCSqGSib3
DQEB\nCwUAME0xSzBJBgNVBAsMQkFtYXpviBXZWlGU2VydmljZXMGtZ1BbWF6b24uY29t\nnEluYy4gTD1TZWF0dGxlfFNUPVdhc2hpbmd0b24gQz1VUzAeFw0yMzA1MjkwNjMw\nnNTdaFw
00OTEyMzEyMzU5NTlaMB4xHDAaBgNVBAMME0FXUyBjb1QgQ2VydGlmaWNh\nndGUwggEi
MA0GCSqGSib3DQEBAAQAA4IBDwAwggEKAoIBAQC5yymzegf2HqAjPZ6F\nnOO7G95cCMbsN
qT2HhtF3wSKxV2Z3Uj4zCEReaFmHwg7ppFciyutQefrHbe
GSGk8\nfl98qQ3q1z8cDy1222kt7aB3r5/MeHU1d9hTk2rTDO4IridXsDgOxbwuVglp37eH\nnX/N
evqVescjO4EjlqKvjclp 2f9SsZiGVA9ilfosrcmPyYX0laPftz338CXoaGGr\nnrS gb8 c
bF9vPyZrGxDPSy37f3wDmuVNAS2dE 4SJ9cJdbWqTevacyYYlwF3smG\nna67brqzrRiAd7sUlsv9
/a2Q/boT0rmiBKXphi9ZVvbJlITxqsr4VDePsi77Uf1p\nnw
nTAGmMBAAGjYDBEMB8GA1UdIwQYMBaAFGyF3xnSijZiJpDAvKAEE4OQKnHlMB0G\nnA1UdDg
QWBBSNMZUVY726CxCNkyw7d8b5zKjrsDAMBgNVHRMBAf8EAjAAMA4GA1Ud\nnDwEB/wQE
AwIHgDANBgkqhkiG9w0BAQsFAAOCAQEAY8YlxbtZEtm5E
M2W35AjCV\nnyMBA3av8xUTc1USUol2WRwuN8z7jBv/EE7Z7SbSY8nuocdevi7z33n6SgSbXX
K/\nOM3oNdC5bKKbELSGUlxUzSltHzEF5tVySRMXw5Bou4uZZwejHYai5waSB1kEzoVH\nnCftYC
TCr/eR4/DDjSrBbLJf72rdrvCCRHLxlc0/gV1Palb8ipgdy8Oo jagop3CN\nngh6Y
B8eUVdHjoBWG7lt46YB2XRnkOAoHmou3x
2zgL4XtHVIqSeoofVII/Ui9mH\nnQZaLnkipn0e/WTkPDe6yNt44uprWiMjKl1xAEVfRptnNoeK
RzTIMoeYvTILHg: =\n-----END CERTIFICATE-----\n",
  "certName":
"c4f6e6184319b7b67ed55ace8e4ab11c47ccc1bf4e764142b328b190ab5d572d-certificate.pe
m.crt",
  "TlSKey": "-----BEGIN RSA PRIVATE
KEY-----\nMIIEpgIBAAKCAQEAucsp3oH9h6glz2ehTjuxveXAJG7Dak9h4bRd8EisVdmd1l
\nMwHEXmhZh8IO6aRXlSrrUHn6x23vhkhpPH5ffKkN6tc/HA8tdtPLe2gd6
fzHh1\nnNXfYU5Nq0wzuCK4nV7A4DsW8LIYCKd
3h1/zXr6IXrHlzuBI5air43Naftn/UrGY\nnhlQPYiH6LK3Jj8mF9JWj37c99/Al6Ghhq60voG/PnPmx
f
bz8maxsQz0st
```

```

398A5r\nlTQETnRPuEifXCXW1qk3r2nMmGCMBdR5hmuu266s60YgHe7FJbL/fv2tkP26E9K5\n
ogSI6YYvWVb24yCE8arK FQ3j7CO
1H9acPp0wIDAQABAoIBAQCnjqKGMYSRVclz\nnN4apAHjFNe1jaZqYTQMpWGoOcSPKtMnITex
AG6NjvuPAEPYlw
X0B2GE3ntZKYTe\n7hroUWzVYUTOUhOugE3mikO4D548sp3xPzIme9phyycQCIDwMiidoHEEC
FpZf4SE\nfQsRa0rVPDJPVNGn9bD08i My93
WTPfjiACDNENH8JlSkKHQQYq8q18gmSjK/Z1\n1I4vtn0B7sVC/LGyrC1koxeVAZAomnC/B9fkHI
X9fMKCwZ0HFyaYKJFi9bAZqZop\nnqHc
jzHkGuvTRHHTkyrZDBgliwqE58UQz0vEpH80b2zq7yXZesV2SAJk1H5cGPxo\nneUMInlp5AoGBA
NpAnD KZQXGawmnKWz
RtamYKlkbifHg4QoSqk2dkli17Bi3e\nngrNWvw32jdYola9gAAxIHh5hbjTeB
LKWteLQMgcvvUhsT6BRuhfXmPiPXOSsjcm\nneilAYRYHOVNua85n3pZA53m81L5FrmVzYLjioh
Mf3yMdNFpryoSkDgZFAoGBANnt\nnZAClaS9H4tk3tAjZcWvNp6Xo5SO
JOnEpxda4fHI5uumM0JcRJSphqOxk QC9UH\nnITqWWZ1ojvoZdh
xpTF9ezPXmQU1ePbJDJU71YFNYq8/Z0kHwARVsommBB/54bXQ\nn6qKZCqezpP3qShdczq
S1U1sPB0/PDbZmpj26903AoGBAMvsJpGwC6gTtps0Vx6B\nnj0qr9gpJo6sDXbyzslaB9RJRYzDo6
Mbzl8mpCi1v9zjbmXgeusZ58RXIJPBCPk8\nnPRI8AfF3 mwr6OIQBGLGCPmmv
gmnuGsz9zJle1IHoj32T7oCi 1wRlb/2Mw1Dvy\nnS Mzzl
0Hdc/SFVb9cNtTbVJAoGBAIO3fGVhPYoJPfAiPde5XcrlAvNKHeG9pDrC\nnIXqG8GZJk5kWJ17OF
Ryl/cJRh0UxgCVKwOfDVfjQ6JH7b6zqVzkZaazE48UFM86d\nnZ9CPMJWNa1cW0FuQZ79
IE/6YBWLPSg66xJ4KDDvc1c4C5ZGc0LV41TAQ1YzSX\nnw1wu6hJzAoGBAMnZsAUtbcCVgK5X
OBLTCJ1hlwMwB X2WktvMWpsSly55RcQG3uQ\nn67CYu7DKRktmcfFp5sJgr4 EV4pSb8
adTs4jSdR7YczkBg5VqgMffzXh10WIIcM\nnoV5nqvg0uVpYNeHKpg9JfQpo5fbyNvdC5BsuA4h5
AbCypj7qSjdZaCx2\nn-----END RSA PRIVATE KEY-----\n",
    "keyName": "c4f6e6184319b7b67ed55ace8e4ab11c47ccc1bf4e764142b328b190ab5d5
72d-private.pem.key",
    "uplinkTopic": "",
    "upQoS": 0,
    "joinTopic": "",
    "joinQoS": 0,
    "ackTopic": "",
    "ackQoS": 0,
    "errorTopic": "",
    "errorQoS": 0,
    "downlinkTopic": "",
    "downlinkQoS": 0,
    "connectStatus": true,
    "mcDownlinkTopic": "",
    "mcDownlinkQoS": 0,
    "requestTopic": "",
    "requestQoS": 0,
    "responseTopic": "",
    "responseQoS": 0
}

```

**Response Example**

```
{}
```

**2) HTTP/HTTPS****Request Address:**

```
https://{gatewayIP}:8080/api/urapplications/{id}/integrations/http
```

**Params**

Parameter	Type	Required	Description
id	String	Yes	The ID of application

**Note: one application only support to add this type integration once.**

**Request Parameters****Header**

Parameter	Required	Description
Content-Type	Yes	application/json

**Body**

Parameter	Type	Required	Description
httpType	String	No	http or https
headers	Object[]	No	
key	String	No	
value	String	No	
dataUpURL	String	No	Uplink data URL
joinNotificationURL	String	No	Join Notification URL
ackNotificationURL	String	No	ACK Notification URL
errorNotificationURL	String	No	Error Notification URL

**Response Parameters****Success:**

```
{}
```

**Failure:**

Parameter	Type	Description
error	String	Error content
code	Integer	Error code

**Request Example**

```
https://192.168.23.150:8080/api/urapplications/11/integrations/http
```

**Body**

```
{
  "headers": [
```

```

{
  "key": "1",
  "value": "3"
},
{
  "key": "2",
  "value": "3"
}
],
"dataUpURL": "http://192.168.1.1",
"joinNotificationURL": "",
"ackNotificationURL": "",
"errorNotificationURL": "",
"httpType": "https"
}

```

### Response Example

```
{}
```

## 3) BACnet/IP

**Note: This only works with firmware v60.0.0.45/56.0.0.5 and before.**

### Request Address:

```
https://{gatewayIP}:8080/api/urapplications/{id}/integrations/bacnetip
```

### Params

Parameter	Type	Required	Description
id	String	Yes	The ID of application

**Note: one application only support to add this type integration once.**

### Request Parameters

#### Header

Parameter	Required	Description
Content-Type	Yes	application/json

### Response Parameters

#### Success:

```
{}
```

#### Failure:

Parameter	Type	Description
error	String	Error content
code	Integer	Error code

### Request Example

```
https://192.168.23.150:8080/api/urapplications/11/integrations/bacnetip
```

### Response Example

```
{}
```

## 4.7 Update Data Transmission Integration

**Method:** PUT

### 1) MQTT

#### Request Address:

```
https://{gatewayIP}:8080/api/urapplications/{id}/integrations/mqtt
```

Params

Parameter	Type	Required	Description
id	String	Yes	The ID of application

#### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

Body

Parameter	Type	Required	Description
host	String	No	Broker address
port	Integer	No	Broker port
useAuth	Boolean	No	Whether to enable user credentials
username	String	No	
password	String	No	
connectTimeout	Integer	No	Connection timeout, unit: s
keepAliveInterval	Integer	No	Keep alive interval, unit: s
clientId	String	No	A unique client ID
useTLS	Boolean	No	Whether to enable TLS
CACert	String	No	CA file content
CAName	String	No	CA file name
TLSCert	String	No	Client certificate file content
certName	String	No	Client certificate file name
TLSKey	String	No	Client key file content
keyName	String	No	Client certificate file name
uplinkTopic	String	No	Uplink data topic
upQoS	Integer	No	Option: 0,1,2
joinTopic	String	No	Join notification topic



joinQoS	Integer	No	Option: 0,1,2
ackTopic	String	No	ACK notification topic
ackQoS	Integer	No	Option: 0,1,2
errorTopic	String	No	Error notification topic
errorQoS	Integer	No	Option: 0,1,2
TLSMode	Integer	No	0 : CA signed certificates 1 : Self signed certificates
downlinkTopic	String	No	Downlink topic
downlinkQoS	Integer	No	Option: 0,1,2
mcDownlinkTopic	String	No	Multicast downlink data topic
mcDownlinkQoS	Integer	No	Option: 0,1,2
requestTopic	String	No	Request data topic
requestQoS	Integer	No	Option: 0,1,2
responseTopic	String	No	Response data topic
responseQoS	Integer	No	Option: 0,1,2 0: QoS 0 1: QoS 1 2: QoS 2

### Response Parameters

Name	Type	Description
Success	json	{}
failed	json	error code

### Request Example

```
https://192.168.23.150:8080/api/urapplications/11/integrations/mqtt
```

#### Body

```
{
  "port": 1883
}
```

### Response Example

```
{}
```

## 2) HTTP/HTTPS

### Request Address:

```
https://{gatewayIP}:8080/api/urapplications/{id}/integrations/http
```

#### Params

Parameter	Type	Required	Description
id	String	Yes	The ID of application

## Request Parameters

### Header

Parameter	Required	Description
Content-Type	Yes	application/json

### Body

Parameter	Type	Required	Description
httpType	String	No	http or https
headers	Object[]	No	
key	String	No	
value	String	No	
dataUpURL	String	No	Uplink data URL
joinNotificationURL	String	No	Join Notification URL
ackNotificationURL	String	No	ACK Notification URL
errorNotificationURL	String	No	Error Notification URL

## Response Parameters

### Success:

{}

### Failure:

Parameter	Type	Description
error	String	Error content
code	Integer	Error code

## Request Example

```
https://192.168.23.150:8080/api/urapplications/11/integrations/http
```

### Body

```
{
  "dataUpURL": "http://192.168.2.1"
}
```

## Response Example

{}

## 4.8 Delete Data Transmission Integration

**Request Method:** DELETE

**Request Address:**

```
https://{gatewayIP}:8080/api/urapplications/{id}/integrations/{data-transmission-type}
```

## Request Parameters

### Header

Parameter	Required	Description
Content-Type	Yes	application/json

### Params

Parameter	Type	Required	Description
id	String	Yes	The ID of application
data-transmission-type	String	Yes	Option: http, mqtt Note: when deleting a HTTPS integration, select http

## Request Example

<https://192.168.23.150:8080/api/urapplications/6/integrations/http>

# 5. Payload Codec

## 5.1 Get All Payload Codec Content

**Request Method:** GET

**Request Address:**

<https://{gatewayIP}:8080/api/payloadcodecs?type=default&limit=10&offset=0>

### Params

Parameter	Type	Required	Description
type	string	Yes	Option: default, custom
limit	String	Yes	Max number of payload codecs to return in the result-test.
offset	String	Yes	Offset in the result-set (for pagination).
search	String	No	Payload codec name

## Request Parameters

### Header

Parameter	Required	Description
Content-Type	Yes	application/json

## Response Parameters

Parameter	Type	Description
totalCount	String	Total count of payload codec library

result	Object[]	Return payload codec content based on limit value
id	String	Payload codec id
name	String	Payload codec name
description	String	
templateID	String	Template ID of this payload codec
devEUIPrefix	String	Device EUI of this type of payload codec matched
decoderScript	String	Decoder content
encoderScript	String	Encoder content
testEnabled	Boolean	Whether to enable Payload Codec test
fPort	Integer	

**Request Example:**

```
https://192.168.23.150:8080/api/payloadcodecs?limit=1&offset=0&type=default
```

**Response Example:**

```
{
  "totalCount": "94",
  "result": [
    {
      "id": "1",
      "name": "AM102",
      "description": "Indoor Ambience Monitoring Sensor",
      "templateID": "0",
      "devEUIPrefix": "24e124725",
      "encoderScript": "/*\n * Payload Encoder for Milesight Network Server\n *\n *
Copyright 2023 Milesight IoT\n *\n * @product AM102\n */\nfunction Encode(fPort, obj)
{\n  var encoded = [];\n  return encoded;\n}\n",
      "decoderScript": "/*\n * Payload Decoder for Milesight Network Server\n *\n *
Copyright 2023 Milesight IoT\n *\n * @product AM102\n */\nfunction Decode(fPort, bytes)
{\n  return milesight(bytes);\n}\n\nfunction milesight(bytes) {\n  var decoded =
{};\n\n  for (var i = 0; i < bytes.length; ) {\n    var channel_id = bytes[i++];\n    var
channel_type = bytes[i++];\n\n    // IPSO VERSION\n    if (channel_id === 0xff &&
channel_type === 0x01) {\n      decoded.ipso_version =
readProtocolVersion(bytes[i]);\n      i += 1;\n    } // HARDWARE
VERSION\n    else if (channel_id === 0xff && channel_type === 0x09)
{\n      decoded.hardware_version = readHardwareVersion(bytes.slice(i, i +
2));\n      i += 2;\n    } // FIRMWARE VERSION\n    else if (channel_id
=== 0xff && channel_type === 0x0a) {\n      decoded.firmware_version =
readFirmwareVersion(bytes.slice(i, i + 2));\n      i += 2;\n    } // DEVICE
STATUS\n    else if (channel_id === 0xff && channel_type === 0x0b)
{\n      decoded.device_status = 1;\n      i += 1;\n    } // LORAWAN
CLASS TYPE\n    else if (channel_id === 0xff && channel_type === 0x0f)
{\n      decoded.lorawan_class = bytes[i];\n      i += 1;\n    } //
SERIAL NUMBER\n    else if (channel_id === 0xff && channel_type === 0x16)
```

```

{
    decoded.sn = readSerialNumber(bytes.slice(i, i + 8));
    i += 8;
}
// BATTERY
else if (channel_id === 0x01 && channel_type === 0x75) {
    decoded.battery = bytes[i];
    i += 1;
}
// TEMPERATURE
else if (channel_id === 0x03 && channel_type === 0x67) {
    // °C
    decoded.temperature = readInt16LE(bytes.slice(i, i + 2)) / 10;
    i += 2;
}
// HUMIDITY
else if (channel_id === 0x04 && channel_type === 0x68) {
    decoded.humidity = bytes[i] / 2;
    i += 1;
}
// HISTORY
else if (channel_id === 0x20 && channel_type === 0xce) {
    var data = {};
    data.timestamp = readUInt32LE(bytes.slice(i, i + 4));
    data.temperature = readInt16LE(bytes.slice(i + 4, i + 6)) / 10;
    data.humidity = bytes[i + 6] / 2;
    i += 7;
    decoded.history = decoded.history || [];
    decoded.history.push(data);
} else {
    break;
}
}
return decoded;
}

function readUInt16LE(bytes) {
    var value = (bytes[1] << 8) + bytes[0];
    return value & 0xffff;
}

function readInt16LE(bytes) {
    var ref = readUInt16LE(bytes);
    return ref > 0x7fff ? ref - 0x10000 : ref;
}

function readUInt32LE(bytes) {
    var value = (bytes[3] << 24) + (bytes[2] << 16) + (bytes[1] << 8) + bytes[0];
    return (value & 0xffffffff) >>> 0;
}

function readInt32LE(bytes) {
    var ref = readUInt32LE(bytes);
    return ref > 0x7fffffff ? ref - 0x100000000 : ref;
}

function readProtocolVersion(bytes) {
    var major = (bytes & 0xf0) >> 4;
    var minor = bytes & 0x0f;
    return "\v" + major + "." + minor;
}

function readHardwareVersion(bytes) {
    var major = bytes[0] & 0xff;
    var minor = (bytes[1] & 0xff) >> 4;
    return "\v" + major + "." + minor;
}

function readFirmwareVersion(bytes) {
    var major = bytes[0] & 0xff;
    var minor = bytes[1] & 0xff;
    return "\v" + major + "." + minor;
}

function readSerialNumber(bytes) {
    var temp = [];
    for (var idx = 0; idx < bytes.length; idx++) {
        temp.push(("0" + (bytes[idx] & 0xff).toString(16)).slice(-2));
    }
    return temp.join("");
}

"testEnabled": false,
"fPort": 0
}
]
}

```

## 5.2 Get Payload Codec ID and Name

**Request Method:** GET

**Request Address:**

<https://{gatewayIP}:8080/api/payloadcodecs-short>

Params

Parameter	Type	Required	Description
type	string	No	Option: default, custom

### Request Parameters

#### Header

Parameter	Required	Description
Content-Type	Yes	application/json

### Response Parameters

Parameter	Type	Description
totalCount	String	Total count of this type payload codec library
result	Object[]	
id	String	Payload codec id
name	String	Payload codec name

### Request Example:

```
https://192.168.23.150:8080/api/payloadcodecs-short
```

### Response Example:

```
{
  "totalCount": "95",
  "result": [
    {
      "id": "1",
      "name": "AM102"
    },
    {
      "id": "2",
      "name": "AM102L"
    },
    .....
  ]
}
```

## 5.3 Get Payload Codec ID by Device EUI

**Request Method:** GET

**Request Address:**

```
https://{gatewayIP}:8080/api/payloadcodecs/{devEUI}/device
```

Params

Parameter	Type	Required	Description
devEUI	string	Yes	Milesight device EUI

## Request Parameters

### Header

Parameter	Required	Description
Content-Type	Yes	application/json

## Response Parameters

Parameter	Type	Description
totalCount	String	Total count of payload codec to suit this device
resultDefault	Object[]	Result in default payload codec
id	String	Payload codec id
name	String	Payload codec name
resultCustom	Object[]	Result in custom payload codec
bestMatchID	String	The best payload codec to match this device

## Request Example:

```
https://192.168.23.150:8080/api/payloadcodecs/24E124136E091081/device
```

## Response Example:

```
{
  "totalCount": "9",
  "resultDefault": [
    {
      "id": "20",
      "name": "EM300-CL"
    },
    {
      "id": "21",
      "name": "EM300-DI"
    },
    {
      "id": "22",
      "name": "EM300-DI-Hall"
    },
    {
      "id": "23",
      "name": "EM300-MCS"
    },
    {
      "id": "24",
      "name": "EM300-MLD"
    },
    {

```

```
{
  "id": "25",
  "name": "EM300-SLD"
},
{
  "id": "26",
  "name": "EM300-TH"
},
{
  "id": "27",
  "name": "EM300-ZLD"
}
],
"resultCustom": [
  {
    "id": "95",
    "name": "TS201"
  }
],
"bestMatchID": "20"
}
```

## 5.4 Get Payload Codec by ID

**Request Method:** GET

**Request Address:**

`https://{gatewayIP}:8080/api/payloadcodecs/{id}`

Params

Parameter	Type	Required	Description
id	string	Yes	The ID of every payload codec can be searched via previous get methods

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

### Response Parameters

Parameter	Type	Description
id	String	
name	String	Payload codec name



description	String	
templateID	String	Template ID of this payload codec
devEUIPrefix	String	Device EUI of this type of payload codec matched
decoderScript	String	Decoder content
encoderScript	String	Encoder content
testEnabled	Boolean	Whether to enable Payload Codec test
fPort	Integer	

**Request Example:**

```
https://192.168.23.150:8080/api/payloadcodecs/1
```

**Response Example:**

```
{
  "id": "1",
  "name": "AM102",
  "description": "Indoor Ambience Monitoring Sensor",
  "templateID": "0",
  "devEUIPrefix": "24e124725",
  "encoderScript": "/*\n * Payload Encoder for Milesight Network Server\n *\n *
Copyright 2023 Milesight IoT\n * @product AM102\n */\nfunction Encode(fPort, obj)
{\n  var encoded = [];\n  return encoded;\n}\n",
  "decoderScript": "/*\n * Payload Decoder for Milesight Network Server\n *\n *
Copyright 2023 Milesight IoT\n * @product AM102\n */\nfunction Decode(fPort, bytes)
{\n  return milesight(bytes);\n}\n\nfunction milesight(bytes) {\n  var decoded =
{};\n  for (var i = 0; i < bytes.length; ) {\n    var channel_id = bytes[i++];\n    var
channel_type = bytes[i++];\n    // IPSO VERSION\n    if (channel_id === 0xff &&
channel_type === 0x01) {\n      decoded.ipso_version =
readProtocolVersion(bytes[i]);\n      i += 1;\n    } // HARDWARE
VERSION\n    else if (channel_id === 0xff && channel_type === 0x09)
{\n      decoded.hardware_version = readHardwareVersion(bytes.slice(i, i +
2));\n      i += 2;\n    } // FIRMWARE VERSION\n    else if (channel_id
=== 0xff && channel_type === 0x0a) {\n      decoded.firmware_version =
readFirmwareVersion(bytes.slice(i, i + 2));\n      i += 2;\n    } // DEVICE
STATUS\n    else if (channel_id === 0xff && channel_type === 0x0b)
{\n      decoded.device_status = 1;\n      i += 1;\n    } // LORAWAN
CLASS TYPE\n    else if (channel_id === 0xff && channel_type === 0x0f)
{\n      decoded.lorawan_class = bytes[i];\n      i += 1;\n    } //
SERIAL NUMBER\n    else if (channel_id === 0xff && channel_type === 0x16)
{\n      decoded.sn = readSerialNumber(bytes.slice(i, i + 8));\n      i +=
8;\n    } // BATTERY\n    else if (channel_id === 0x01 && channel_type ===
0x75) {\n      decoded.battery = bytes[i];\n      i += 1;\n    } //
TEMPERATURE\n    else if (channel_id === 0x03 && channel_type === 0x67)
{\n      // °C\n      decoded.temperature = readInt16LE(bytes.slice(i, i + 2)) /
10;\n      i += 2;\n    } // HUMIDITY\n    else if (channel_id === 0x04
```

```

&& channel_type === 0x68) {\n        decoded.humidity = bytes[i] / 2;\n        i +=\n1;\n    }\n    // HISTORY\n    else if (channel_id === 0x20 && channel_type ===\n0xce) {\n        var data = {};\n        data.timestamp = readUInt32LE(bytes.slice(i, i\n+ 4));\n        data.temperature = readInt16LE(bytes.slice(i + 4, i + 6)) /\n10;\n        data.humidity = bytes[i + 6] / 2;\n        i +=\n7;\n\n        decoded.history = decoded.history ||\n[];\n        decoded.history.push(data);\n    } else\n{\n        break;\n    }\n}\n\nreturn decoded;\n}\n\nfunction\nreadUInt16LE(bytes) {\n    var value = (bytes[1] << 8) + bytes[0];\n    return value &\n0xffff;\n}\n\nfunction readInt16LE(bytes) {\n    var ref = readUInt16LE(bytes);\n    return\nref > 0x7fff ? ref - 0x10000 : ref;\n}\n\nfunction readUInt32LE(bytes) {\n    var value =\n(bytes[3] << 24) + (bytes[2] << 16) + (bytes[1] << 8) + bytes[0];\n    return (value &\n0xffffffff) >>> 0;\n}\n\nfunction readInt32LE(bytes) {\n    var ref =\nreadUInt32LE(bytes);\n    return ref > 0x7fffffff ? ref - 0x100000000 : ref;\n}\n\nfunction\nreadProtocolVersion(bytes) {\n    var major = (bytes & 0xf0) >> 4;\n    var minor = bytes &\n0x0f;\n    return "\\v\\" + major + "\\." + minor;\n}\n\nfunction readHardwareVersion(bytes)\n{\n    var major = bytes[0] & 0xff;\n    var minor = (bytes[1] & 0xff) >> 4;\n    return "\\v\\" +\nmajor + "\\." + minor;\n}\n\nfunction readFirmwareVersion(bytes) {\n    var major =\nbytes[0] & 0xff;\n    var minor = bytes[1] & 0xff;\n    return "\\v\\" + major + "\\." +\nminor;\n}\n\nfunction readSerialNumber(bytes) {\n    var temp = [];\n    for (var idx = 0;\nidx < bytes.length; idx++) {\n        temp.push(("0\\n" + (bytes[idx] &\n0xff).toString(16)).slice(-2));\n    }\n    return temp.join("\\n");\n}\n\n\n    "testEnabled": false,\n    "fPort": 0\n}\n

```

## 5.5 Create a Custom Payload Codec

**Request Method:** POST

**Request Address:**

<https://{gatewayIP}:8080/api/payloadcodecs>

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

Body

Parameter	Type	Required	Description
name	String	No	Payload codec name
description	String	No	
templateID	String	No	Template ID of payload codec, 0=none
decoderScript	String	No	Decoder content

encoderScript	String	No	Encoder content
testEnabled	Boolean	No	Whether to enable Payload Codec test
fPort	Integer	No	

### Response Parameters

Parameter	Type	Description
id	String	Payload codec ID when create with success
error	String	Error content
code	Integer	error code, return 200 when success

### Request Example

```
{
  "name": "TS201",
  "description": "test",
  "templateID": "0",
  "encoderScript": "",
  "decoderScript": "/*\n * Payload Decoder\n *\n * Copyright 2024 Milesight IoT\n *\n *
@product TS201\n *\n\n// Chirpstack v3\nfunction Decode(fPort, bytes) {\n  return
milesightDeviceDecode(bytes);\n}\n\nfunction milesightDeviceDecode(bytes) {\n  var
decoded = {};\n\n  for (var i = 0; i < bytes.length; ) {\n    var channel_id =
bytes[i++];\n    var channel_type = bytes[i++];\n\n    // IPSO VERSION\n    if
(channel_id === 0xff && channel_type === 0x01) {\n      decoded.ipso_version =
readProtocolVersion(bytes[i]);\n      i += 1;\n    }\n    // HARDWARE
VERSION\n    else if (channel_id === 0xff && channel_type === 0x09)
{\n      decoded.hardware_version = readHardwareVersion(bytes.slice(i, i +
2));\n      i += 2;\n    }\n    // FIRMWARE VERSION\n    else if (channel_id
=== 0xff && channel_type === 0x0a) {\n      decoded.firmware_version =
readFirmwareVersion(bytes.slice(i, i + 2));\n      i += 2;\n    }\n    // DEVICE
STATUS\n    else if (channel_id === 0xff && channel_type === 0x0b)
{\n      decoded.device_status = \"on\";\n      i += 1;\n    }\n    //
LORAWAN CLASS TYPE\n    else if (channel_id === 0xff && channel_type === 0x0f)
{\n      decoded.lorawan_class = readLoRaWANType(bytes[i]);\n      i +=
1;\n    }\n    // SERIAL NUMBER\n    else if (channel_id === 0xff &&
channel_type === 0x16) {\n      decoded.sn = readSerialNumber(bytes.slice(i, i +
8));\n      i += 8;\n    }\n    // TSL VERSION\n    else if (channel_id ===
0xff && channel_type === 0xff) {\n      decoded.tsl_version =
readTslVersion(bytes.slice(i, i + 2));\n      i += 2;\n    }\n    //
BATTERY\n    else if (channel_id === 0x01 && channel_type === 0x75)
{\n      decoded.battery = bytes[i];\n      i += 1;\n    }\n    //
TEMPERATURE\n    else if (channel_id === 0x03 && channel_type === 0x67)
{\n      decoded.temperature = readInt16LE(bytes.slice(i, i + 2)) / 10;\n      i +=
2;\n    }\n    // TEMPERATURE THRESHOLD ALARM\n    else if (channel_id ===
0x83 && channel_type === 0x67) {\n      var data = {};\n      data.temperature =
readInt16LE(bytes.slice(i, i + 2)) / 10;\n      data.temperature_alarm =
```

```

readAlarmType(bytes[i + 2]);\n          i += 3;\n\n          decoded.temperature =
data.temperature;\n          decoded.event = decoded.event ||
[];\n          decoded.event.push(data);\n          }\n          // TEMPERATURE MUTATION
ALARM\n          else if (channel_id === 0x93 && channel_type === 0x67) {\n          var
data = {};\n          data.temperature = readInt16LE(bytes.slice(i, i + 2)) /
10;\n          data.temperature_mutation = readInt16LE(bytes.slice(i + 2, i + 4)) /
10;\n          data.temperature_alarm = readAlarmType(bytes[i + 4]);\n          i +=
5;\n\n          decoded.temperature = data.temperature;\n          decoded.event =
decoded.event || [];\n          decoded.event.push(data);\n          }\n          //
TEMPERATURE ERROR\n          else if (channel_id === 0xb3 && channel_type === 0x67)
{\n          var data = {};\n          data.temperature_error =
readErrorType(bytes[i]);\n          i += 1;\n\n          decoded.event = decoded.event
|| [];\n          decoded.event.push(data);\n          }\n          // HISTORY
DATA\n          else if (channel_id === 0x20 && channel_type === 0xce) {\n          var
timestamp = readUInt32LE(bytes.slice(i, i + 4));\n          var event = bytes[i +
4];\n          var temperature = readInt16LE(bytes.slice(i + 5, i + 7)) / 10;\n          i +=
7;\n\n          var read_status = readStatus((event >>> 4) & 0x0f);\n          var
event_type = readType(event & 0x0f);\n          var data =
{};\n          data.timestamp = timestamp;\n          data.read_status =
read_status;\n          data.event_type = event_type;\n          data.temperature =
temperature;\n\n          decoded.history = decoded.history ||
[];\n          decoded.history.push(data);\n          } else
{\n          break;\n          }\n          }\n\n          return decoded;\n\n}\n\nfunction
readUInt8(bytes) {\n          return bytes & 0xff;\n\n}\n\nfunction readInt8(bytes) {\n          var ref =
readUInt8(bytes);\n          return ref > 0x7f ? ref - 0x100 : ref;\n\n}\n\nfunction
readUInt16LE(bytes) {\n          var value = (bytes[1] << 8) + bytes[0];\n          return value &
0xffff;\n\n}\n\nfunction readInt16LE(bytes) {\n          var ref = readUInt16LE(bytes);\n          return
ref > 0x7fff ? ref - 0x10000 : ref;\n\n}\n\nfunction readUInt32LE(bytes) {\n          var value =
(bytes[3] << 24) + (bytes[2] << 16) + (bytes[1] << 8) + bytes[0];\n          return (value &
0xffffffff) >>> 0;\n\n}\n\nfunction readInt32LE(bytes) {\n          var ref =
readUInt32LE(bytes);\n          return ref > 0x7fffffff ? ref - 0x100000000 : ref;\n\n}\n\nfunction
readFloatLE(bytes) {\n          var bits = (bytes[3] << 24) | (bytes[2] << 16) | (bytes[1] << 8) |
bytes[0];\n          var sign = bits >>> 31 === 0 ? 1.0 : -1.0;\n          var e = (bits >>> 23) &
0xff;\n          var m = e === 0 ? (bits & 0x7fffff) << 1 : (bits & 0x7fffff) | 0x800000;\n          var f =
sign * m * Math.pow(2, e - 150);\n          return f;\n\n}\n\nfunction readProtocolVersion(bytes)
{\n          var major = (bytes & 0xf0) >> 4;\n          var minor = bytes & 0x0f;\n          return "\"v\" +
major + \".\" + minor;\n\n}\n\nfunction readHardwareVersion(bytes) {\n          var major =
bytes[0] & 0xff;\n          var minor = (bytes[1] & 0xff) >> 4;\n          return "\"v\" + major + \".\" +
minor;\n\n}\n\nfunction readFirmwareVersion(bytes) {\n          var major = bytes[0] &
0xff;\n          var minor = bytes[1] & 0xff;\n          return "\"v\" + major + \".\" +
minor;\n\n}\n\nfunction readTslVersion(bytes) {\n          var major = bytes[0] & 0xff;\n          var
minor = bytes[1] & 0xff;\n          return "\"v\" + major + \".\" + minor;\n\n}\n\nfunction
readSerialNumber(bytes) {\n          var temp = [];\n          for (var idx = 0; idx < bytes.length; idx++)
{\n          temp.push(("0" + (bytes[idx] & 0xff).toString(16)).slice(-2));\n          }\n          return

```

```
temp.join("\\");\n}\n\nfunction readLoRaWANType(type) {\n  switch (type) {\n    case 0x00:\n      return \"ClassA\";\n    case 0x01:\n      return \"ClassB\";\n    case 0x02:\n      return \"ClassC\";\n    case 0x03:\n      return \"ClassCtoB\";\n    default:\n      return \"Unknown\";\n  }\n}\n\nfunction readAlarmType(type) {\n  switch (type) {\n    case 0x00:\n      return \"Threshold Alarm Release\";\n    case 0x01:\n      return \"Threshold Alarm\";\n    case 0x02:\n      return \"Mutation Alarm\";\n    default:\n      return \"Unknown\";\n  }\n}\n\nfunction readErrorType(type) {\n  switch (type) {\n    case 0x00:\n      return \"Read Error\";\n    case 0x01:\n      return \"Overload\";\n    default:\n      return \"Unknown\";\n  }\n}\n\nfunction readHistoryEvent(type) {\n  switch (type) {\n    case 0x00:\n      return \"Time Update\";\n    case 0x01:\n      return \"Periodic\";\n    case 0x02:\n      return \"Alarm(Threshold or Mutation)\";\n    case 0x03:\n      return \"Alarm Release\";\n    case 0x04:\n      return \"Read Error\";\n    case 0x05:\n      return \"Overload\";\n    default:\n      return \"Unknown\";\n  }\n}\n\nfunction readStatus(type) {\n  switch (type) {\n    case 0x00:\n      return \"Success\";\n    case 0x01:\n      return \"Read Error\";\n    case 0x02:\n      return \"Overload\";\n    default:\n      return \"Unknown\";\n  }\n}\n\nfunction readType(type) {\n  switch (type) {\n    case 0x00:\n      return \"\";\n    case 0x01:\n      return \"Periodic\";\n    case 0x02:\n      return \"Alarm(Threshold or Mutation)\";\n    case 0x03:\n      return \"Alarm Release\";\n    default:\n      return \"Unknown\";\n  }\n}\n\n,\"testEnabled\": false,\n  \"fPort\": 1\n}
```

### Response Example

```
{
  \"id\": \"96\",
  \"error\": \"\",
  \"code\": 200
}
```

## 5.6 Update a Custom Payload Codec

**Request Method:** PUT

**Request Address:**

<https://{gatewayIP}:8080/api/payloadcodecs/{id}>

Params

Parameter	Type	Required	Description
-----------	------	----------	-------------

id	string	Yes	Payload codec ID which need to update
----	--------	-----	---------------------------------------

## Request Parameters

### Header

Parameter	Required	Description
Content-Type	Yes	application/json

### Body

Parameter	Type	Required	Description
name	String	Yes	Payload codec name
description	String	Yes	
templateID	String	Yes	Template ID of payload codec, 0=none
decoderScript	String	Yes	Decoder content
encoderScript	String	Yes	Encoder content
testEnabled	Boolean	Yes	Whether to enable Payload Codec test
fPort	Integer	Yes	

**Note: when updating the parameter of a payload codec, it is necessary to include all settings into request.**

## Response Parameters

Parameter	Type	Description
error	String	Error content
code	Integer	error code, return 200 when success

## Request Example

<https://192.168.23.150:8080/api/payloadcodecs/96>

### Body

```
{
  "name": "my-code",
  "description": "test",
  "templateID": "0",
  "encoderScript": "",
  "decoderScript": "/*\n * Payload Decoder\n * \n * Copyright 2024 Milesight IoT\n * \n *
@product TS201\n * /\n\n\n// Chirpstack v3\nfunction Decode(fPort, bytes) {\n  return
milesightDeviceDecode(bytes);\n}\n\nfunction milesightDeviceDecode(bytes) {\n  var
decoded = {};\n\n  for (var i = 0; i < bytes.length; ) {\n\n    var channel_id =
bytes[i++];\n\n    var channel_type = bytes[i++];\n\n\n    // IPSO VERSION\n    if
(channel_id === 0xff && channel_type === 0x01) {\n\n      decoded.ipso_version =
readProtocolVersion(bytes[i]);\n\n      i += 1;\n\n    }\n\n    // HARDWARE
VERSION\n    else if (channel_id === 0xff && channel_type === 0x09)
{\n\n      decoded.hardware_version = readHardwareVersion(bytes.slice(i, i +
2));\n\n      i += 2;\n\n    }\n\n    // FIRMWARE VERSION\n    else if (channel_id
=== 0xff && channel_type === 0x0a) {\n\n      decoded.firmware_version =
readFirmwareVersion(bytes.slice(i, i + 2));\n\n      i += 2;\n\n    }\n\n    // DEVICE
```

```

STATUS\n        else if (channel_id === 0xff && channel_type === 0x0b)
{\n            decoded.device_status = \"on\";\n                i += 1;\n        }\n        //
LORAWAN CLASS TYPE\n        else if (channel_id === 0xff && channel_type === 0x0f)
{\n            decoded.lorawan_class = readLoRaWANType(bytes[i]);\n                i +=
1;\n        }\n        // SERIAL NUMBER\n        else if (channel_id === 0xff && channel_type
=== 0x16) {\n            decoded.sn = readSerialNumber(bytes.slice(i, i + 8));\n                i +=
8;\n        }\n        // TSL VERSION\n        else if (channel_id === 0xff && channel_type ===
0xff) {\n            decoded.tsl_version = readTslVersion(bytes.slice(i, i + 2));\n                i +=
2;\n        }\n        // BATTERY\n        else if (channel_id === 0x01 && channel_type ===
0x75) {\n            decoded.battery = bytes[i];\n                i += 1;\n        }\n        //
TEMPERATURE\n        else if (channel_id === 0x03 && channel_type === 0x67)
{\n            decoded.temperature = readInt16LE(bytes.slice(i, i + 2)) / 10;\n                i +=
2;\n        }\n        // TEMPERATURE THRESHOLD ALARM\n        else if (channel_id ===
0x83 && channel_type === 0x67) {\n            var data = {};\n                data.temperature =
readInt16LE(bytes.slice(i, i + 2)) / 10;\n                data.temperature_alarm =
readAlarmType(bytes[i + 2]);\n                i += 3;\n        }\n        decoded.temperature =
data.temperature;\n        decoded.event = decoded.event ||
[];\n        decoded.event.push(data);\n        }\n        // TEMPERATURE MUTATION
ALARM\n        else if (channel_id === 0x93 && channel_type === 0x67) {\n            var
data = {};\n                data.temperature = readInt16LE(bytes.slice(i, i + 2)) /
10;\n                data.temperature_mutation = readInt16LE(bytes.slice(i + 2, i + 4)) /
10;\n                data.temperature_alarm = readAlarmType(bytes[i + 4]);\n                i +=
5;\n        }\n        decoded.temperature = data.temperature;\n        decoded.event =
decoded.event || [];\n        decoded.event.push(data);\n        }\n        //
TEMPERATURE ERROR\n        else if (channel_id === 0xb3 && channel_type === 0x67)
{\n            var data = {};\n                data.temperature_error =
readErrorType(bytes[i]);\n                i += 1;\n        }\n        decoded.event = decoded.event ||
[];\n        decoded.event.push(data);\n        }\n        // HISTORY DATA\n        else if
(channel_id === 0x20 && channel_type === 0xce) {\n            var timestamp =
readUInt32LE(bytes.slice(i, i + 4));\n            var event = bytes[i + 4];\n            var
temperature = readInt16LE(bytes.slice(i + 5, i + 7)) / 10;\n            i += 7;\n            var
read_status = readStatus((event >>> 4) & 0x0f);\n            var event_type = readType(event
& 0x0f);\n            var data = {};\n            data.timestamp =
timestamp;\n            data.read_status = read_status;\n            data.event_type =
event_type;\n            data.temperature = temperature;\n            decoded.history =
decoded.history || [];\n            decoded.history.push(data);\n        } else
{\n            break;\n        }\n        }\n        return decoded;\n    }\n\nfunction readUInt8(bytes)
{\n    return bytes & 0xff;\n}\n\nfunction readInt8(bytes) {\n    var ref =
readUInt8(bytes);\n    return ref > 0x7f ? ref - 0x100 : ref;\n}\n\nfunction readUInt16LE(bytes)
{\n    var value = (bytes[1] << 8) + bytes[0];\n    return value & 0xffff;\n}\n\nfunction
readInt16LE(bytes) {\n    var ref = readUInt16LE(bytes);\n    return ref > 0x7fff ? ref -
0x10000 : ref;\n}\n\nfunction readUInt32LE(bytes) {\n    var value = (bytes[3] << 24) +
(bytes[2] << 16) + (bytes[1] << 8) + bytes[0];\n    return (value & 0xffffffff) >>>
0;\n}\n\nfunction readInt32LE(bytes) {\n    var ref = readUInt32LE(bytes);\n    return ref >

```



```

0x7fffffff ? ref - 0x100000000 : ref;\n}\n\nfunction readFloatLE(bytes) {\n  var bits =
(bytes[3] << 24) | (bytes[2] << 16) | (bytes[1] << 8) | bytes[0];\n  var sign = bits >>> 31 ===
0 ? 1.0 : -1.0;\n  var e = (bits >>> 23) & 0xff;\n  var m = e === 0 ? (bits & 0x7fffff) << 1 : (bits
& 0x7fffff) | 0x800000;\n  var f = sign * m * Math.pow(2, e - 150);\n  return
f;\n}\n\nfunction readProtocolVersion(bytes) {\n  var major = (bytes & 0xf0) >> 4;\n  var
minor = bytes & 0x0f;\n  return \"v\" + major + \".\" + minor;\n}\n\nfunction
readHardwareVersion(bytes) {\n  var major = bytes[0] & 0xff;\n  var minor = (bytes[1] &
0xff) >> 4;\n  return \"v\" + major + \".\" + minor;\n}\n\nfunction
readFirmwareVersion(bytes) {\n  var major = bytes[0] & 0xff;\n  var minor = bytes[1] &
0xff;\n  return \"v\" + major + \".\" + minor;\n}\n\nfunction readTslVersion(bytes) {\n  var
major = bytes[0] & 0xff;\n  var minor = bytes[1] & 0xff;\n  return \"v\" + major + \".\" +
minor;\n}\n\nfunction readSerialNumber(bytes) {\n  var temp = [];\n  for (var idx = 0; idx <
bytes.length; idx++) {\n    temp.push((\"0\" + (bytes[idx] &
0xff).toString(16)).slice(-2));\n  }\n  return temp.join(\"\\");\n}\n\nfunction
readLoRaWANType(type) {\n  switch (type) {\n    case 0x00:\n      return
\"ClassA\";\n    case 0x01:\n      return \"ClassB\";\n    case
0x02:\n      return \"ClassC\";\n    case 0x03:\n      return
\"ClassCtoB\";\n    default:\n      return \"Unknown\";\n  }\n}\n\nfunction
readAlarmType(type) {\n  switch (type) {\n    case 0x00:\n      return \"Threshold
Alarm Release\";\n    case 0x01:\n      return \"Threshold Alarm\";\n    case
0x02:\n      return \"Mutation Alarm\";\n    default:\n      return
\"Unknown\";\n  }\n}\n\nfunction readErrorType(type) {\n  switch (type) {\n    case
0x00:\n      return \"Read Error\";\n    case 0x01:\n      return
\"Overload\";\n    default:\n      return \"Unknown\";\n  }\n}\n\nfunction
readHistoryEvent(type) {\n  switch (type) {\n    case 0x00:\n      return \"Time
Update\";\n    case 0x01:\n      return \"Periodic\";\n    case
0x02:\n      return \"Alarm(Threshold or Mutation)\";\n    case
0x03:\n      return \"Alarm Release\";\n    case 0x04:\n      return \"Read
Error\";\n    case 0x05:\n      return
\"Overload\";\n    default:\n      return \"Unknown\";\n  }\n}\n\nfunction
readStatus(type) {\n  switch (type) {\n    case 0x00:\n      return
\"Success\";\n    case 0x01:\n      return \"Read Error\";\n    case
0x02:\n      return \"Overload\";\n    default:\n      return
\"Unknown\";\n  }\n}\n\nfunction readType(type) {\n  switch (type) {\n    case
0x00:\n      return \"\";\n    case 0x01:\n      return
\"Periodic\";\n    case 0x02:\n      return \"Alarm(Threshold or
Mutation)\";\n    case 0x03:\n      return \"Alarm
Release\";\n    default:\n      return \"Unknown\";\n  }\n}\n\",
  \"testEnabled\": false,
  \"fPort\": 1
}

```

### Response Example

```
{
```



```
"error": "",  
"code": 200  
}
```

## 5.7 Delete a Custom Payload Codec

**Request Method:** DELETE

**Request Address:**

`https://{gatewayIP}:8080/api/payloadcodecs/{id}`

Params

Parameter	Type	Required	Description
id	string	Yes	Payload codec ID which need to delete

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

### Response Parameters

Parameter	Type	Description
error	String	Error content
code	Integer	error code, return 200 when success

### Request Example

`https://192.168.23.150:8080/api/payloadcodecs/96`

### Response Example

```
{  
  "error": "",  
  "code": 200  
}
```

## 5.8 Payload Codec Test

**Request Method:** POST

**Request Address:**

`https://{gatewayIP}:8080/api/payloadcodecs-test`

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

## Body

Parameter	Type	Required	Description
type	String	No	Option: encode, decode
fPort	Integer	No	
data	String	No	Raw data need to encode or decode
script	String	No	Encoder or decoder content

## Response Parameters

Parameter	Type	Description
result	String	Encode or decode result

## Request Example

```
{
  "type": "decode",
  "fport": 85,
  "data": "01756403671101",
  "script": "/*\n * Payload Decoder\n *\n * Copyright 2024 Milesight IoT\n *\n *
@product TS201\n */\n\n// Chirpstack v3\nfunction Decode(fPort, bytes) {\n  return
milesightDeviceDecode(bytes);\n}\n\nfunction milesightDeviceDecode(bytes) {\n  var
decoded = {};\n  for (var i = 0; i < bytes.length; ) {\n    var channel_id =
bytes[i++];\n    var channel_type = bytes[i++];\n    // IPSO VERSION\n    if
(channel_id === 0xff && channel_type === 0x01) {\n      decoded.ipso_version =
readProtocolVersion(bytes[i]);\n      i += 1;\n    }\n    // HARDWARE
VERSION\n    else if (channel_id === 0xff && channel_type === 0x09)
{\n      decoded.hardware_version = readHardwareVersion(bytes.slice(i, i +
2));\n      i += 2;\n    }\n    // FIRMWARE VERSION\n    else if (channel_id
=== 0xff && channel_type === 0x0a) {\n      decoded.firmware_version =
readFirmwareVersion(bytes.slice(i, i + 2));\n      i += 2;\n    }\n    // DEVICE
STATUS\n    else if (channel_id === 0xff && channel_type === 0x0b)
{\n      decoded.device_status = \"on\";\n      i += 1;\n    }\n    //
LORAWAN CLASS TYPE\n    else if (channel_id === 0xff && channel_type === 0x0f)
{\n      decoded.lorawan_class = readLoRaWANType(bytes[i]);\n      i +=
1;\n    }\n    // SERIAL NUMBER\n    else if (channel_id === 0xff &&
channel_type === 0x16) {\n      decoded.sn = readSerialNumber(bytes.slice(i, i +
8));\n      i += 8;\n    }\n    // TSL VERSION\n    else if (channel_id ===
0xff && channel_type === 0xff) {\n      decoded.tsl_version =
readTslVersion(bytes.slice(i, i + 2));\n      i += 2;\n    }\n    //
BATTERY\n    else if (channel_id === 0x01 && channel_type === 0x75)
{\n      decoded.battery = bytes[i];\n      i += 1;\n    }\n    //
TEMPERATURE\n    else if (channel_id === 0x03 && channel_type === 0x67)
{\n      decoded.temperature = readInt16LE(bytes.slice(i, i + 2)) / 10;\n      i +=
2;\n    }\n    // TEMPERATURE THRESHOLD ALARM\n    else if (channel_id ===
```

```

0x83 && channel_type === 0x67) {\n          var data = {};\n          data.temperature =
readInt16LE(bytes.slice(i, i + 2)) / 10;\n          data.temperature_alarm =
readAlarmType(bytes[i + 2]);\n          i += 3;\n          decoded.temperature =
data.temperature;\n          decoded.event = decoded.event ||
[];\n          decoded.event.push(data);\n          }\n          // TEMPERATURE MUTATION
ALARM\n          else if (channel_id === 0x93 && channel_type === 0x67) {\n          var
data = {};\n          data.temperature = readInt16LE(bytes.slice(i, i + 2)) /
10;\n          data.temperature_mutation = readInt16LE(bytes.slice(i + 2, i + 4)) /
10;\n          data.temperature_alarm = readAlarmType(bytes[i + 4]);\n          i +=
5;\n          decoded.temperature = data.temperature;\n          decoded.event =
decoded.event || [];\n          decoded.event.push(data);\n          }\n          //
TEMPERATURE ERROR\n          else if (channel_id === 0xb3 && channel_type === 0x67)
{\n          var data = {};\n          data.temperature_error =
readErrorType(bytes[i]);\n          i += 1;\n          decoded.event = decoded.event
|| [];\n          decoded.event.push(data);\n          }\n          // HISTORY
DATA\n          else if (channel_id === 0x20 && channel_type === 0xce) {\n          var
timestamp = readUInt32LE(bytes.slice(i, i + 4));\n          var event = bytes[i +
4];\n          var temperature = readInt16LE(bytes.slice(i + 5, i + 7)) / 10;\n          i +=
7;\n          var read_status = readStatus((event >>> 4) & 0x0f);\n          var
event_type = readType(event & 0x0f);\n          var data =
{};\n          data.timestamp = timestamp;\n          data.read_status =
read_status;\n          data.event_type = event_type;\n          data.temperature =
temperature;\n          decoded.history = decoded.history ||
[];\n          decoded.history.push(data);\n          } else
{\n          break;\n          }\n          }\n          return decoded;\n          }\n\nfunction
readUInt8(bytes) {\n    return bytes & 0xff;\n          }\n\nfunction readInt8(bytes) {\n    var ref =
readUInt8(bytes);\n    return ref > 0x7f ? ref - 0x100 : ref;\n          }\n\nfunction
readUInt16LE(bytes) {\n    var value = (bytes[1] << 8) + bytes[0];\n    return value &
0xffff;\n          }\n\nfunction readInt16LE(bytes) {\n    var ref = readUInt16LE(bytes);\n    return
ref > 0x7fff ? ref - 0x10000 : ref;\n          }\n\nfunction readUInt32LE(bytes) {\n    var value =
(bytes[3] << 24) + (bytes[2] << 16) + (bytes[1] << 8) + bytes[0];\n    return (value &
0xffffffff) >>> 0;\n          }\n\nfunction readInt32LE(bytes) {\n    var ref =
readUInt32LE(bytes);\n    return ref > 0x7fffffff ? ref - 0x100000000 : ref;\n          }\n\nfunction
readFloatLE(bytes) {\n    var bits = (bytes[3] << 24) | (bytes[2] << 16) | (bytes[1] << 8) |
bytes[0];\n    var sign = bits >>> 31 === 0 ? 1.0 : -1.0;\n    var e = (bits >>> 23) &
0xff;\n    var m = e === 0 ? (bits & 0x7fffff) << 1 : (bits & 0x7fffff) | 0x800000;\n    var f =
sign * m * Math.pow(2, e - 150);\n    return f;\n          }\n\nfunction readProtocolVersion(bytes)
{\n    var major = (bytes & 0xf0) >> 4;\n    var minor = bytes & 0x0f;\n    return \"v\" +
major + \".\" + minor;\n          }\n\nfunction readHardwareVersion(bytes) {\n    var major =
bytes[0] & 0xff;\n    var minor = (bytes[1] & 0xff) >> 4;\n    return \"v\" + major + \".\" +
minor;\n          }\n\nfunction readFirmwareVersion(bytes) {\n    var major = bytes[0] &
0xff;\n    var minor = bytes[1] & 0xff;\n    return \"v\" + major + \".\" +
minor;\n          }\n\nfunction readTslVersion(bytes) {\n    var major = bytes[0] & 0xff;\n    var
minor = bytes[1] & 0xff;\n    return \"v\" + major + \".\" + minor;\n          }\n\nfunction

```

```

readSerialNumber(bytes) {\n  var temp = [];\n  for (var idx = 0; idx < bytes.length; idx++)\n  {\n    temp.push(("0" + (bytes[idx] & 0xff).toString(16)).slice(-2));\n  }\n  return\n  temp.join("\\");\n}\n\nfunction readLoRaWANType(type) {\n  switch (type) {\n    case\n    0x00:\n      return "ClassA";\n    case 0x01:\n      return\n      "ClassB";\n    case 0x02:\n      return "ClassC";\n    case\n    0x03:\n      return "ClassCtoB";\n    default:\n      return\n      "Unknown";\n  }\n}\n\nfunction readAlarmType(type) {\n  switch (type)\n  {\n    case 0x00:\n      return "Threshold Alarm Release";\n    case\n    0x01:\n      return "Threshold Alarm";\n    case 0x02:\n      return\n      "Mutation Alarm";\n    default:\n      return\n      "Unknown";\n  }\n}\n\nfunction readErrorType(type) {\n  switch (type) {\n    case\n    0x00:\n      return "Read Error";\n    case 0x01:\n      return\n      "Overload";\n    default:\n      return "Unknown";\n  }\n}\n\nfunction\n  readHistoryEvent(type) {\n    switch (type) {\n      case 0x00:\n        return "Time\n        Update";\n      case 0x01:\n        return "Periodic";\n      case\n      0x02:\n        return "Alarm(Threshold or Mutation)";\n      case\n      0x03:\n        return "Alarm Release";\n      case 0x04:\n        return "Read\n        Error";\n      case 0x05:\n        return\n        "Overload";\n      default:\n        return "Unknown";\n    }\n  }\n}\n\nfunction\n  readStatus(type) {\n    switch (type) {\n      case 0x00:\n        return\n        "Success";\n      case 0x01:\n        return "Read Error";\n      case\n      0x02:\n        return "Overload";\n      default:\n        return\n        "Unknown";\n    }\n  }\n}\n\nfunction readType(type) {\n  switch (type) {\n    case\n    0x00:\n      return "";\n    case 0x01:\n      return\n      "Periodic";\n    case 0x02:\n      return "Alarm(Threshold or\n      Mutation)";\n    case 0x03:\n      return "Alarm\n      Release";\n    default:\n      return "Unknown";\n  }\n}\n}

```

### Response Example

```

{
  "result": "{\"battery\":100,\"temperature\":27.3}"
}

```

## 5.9 Get Payload Codec Setting

**Request Method:** GET

**Request Address:**

<https://{gatewayIP}:8080/api/payloadcodecs-setting>

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

**Response Parameters**

Parameter	Type	Description
onlineUpgradeEnabled	Boolean	Whether to update online
version	String	Payload codec library version
updatedAt	String	Payload codec latest updated time
upgradeType	String	Obtaining type

**Request Example:**

https://192.168.23.150:8080/api/payloadcodecs-setting

**Response Example:**

```
{
  "onlineUpgradeEnabled": true,
  "version": "1.1.4",
  "updatedAt": "2024-05-21T18:16:48.026106+08:00",
  "upgradeType": "online"
}
```

## 5.10 Update Payload Codec Online

**Request Method:** GET

**Request Address:**

https://{gatewayIP}:8080/api/payloadcodecs-upgrade

**Request Parameters**

Header

Parameter	Required	Description
Content-Type	Yes	application/json

**Response Parameters**

Parameter	Type	Description
version	String	Payload codec library version after update
error	String	Error content
code	Integer	Error code, return 200 when success

**Request Example:**

https://192.168.23.150:8080/api/payloadcodecs-upgrade

**Response Example:**

```
{
  "version": "",
  "error": "version is newest",
  "code": 10003
}
```

## 5.11 Get Payload Codec Local Import Result

**Request Method:** GET

**Request Address:**

`https://{gatewayIP}:8080/api/payloadcodecs-import`

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

### Response Parameters

Parameter	Type	Description
succCount	Integer	The count of payload codec imported to gateway with success
failCount	Integer	The count of payload codec imported to gateway with failure
error	String	Error content

**Request Example:**

`https://192.168.23.150:8080/api/payloadcodecs-import`

**Response Example:**

```
{
  "succCount": 0,
  "failCount": 0,
  "error": "stat /tmp/lora_payloadcodec.zip: no such file or directory"
}
```

## 6. Profile

### 6.1 Get Profile

**Result Method:** GET

**Request Address:**

<https://{gatewayIP}:8080/api/urprofiles>

Params

Parameter	Type	Required	Description
limit	String	Yes	Max number of profiles to return in the result-test.
offset	String	Yes	Offset in the result-set (for pagination).
organizationID	String	Yes	1
applicationID	String	Yes	1
profileID	String	No	Search the profile content of this ID

#### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

#### Response Parameters

Parameter	Type	Description
totalCount	String	Total number of profiles.
disable	Boolean	Whether Milesight IoT Cloud is disabled
result	Array []	
profile	Object[]	
profileID	String	Profile ID
supporsClassB	Boolean	Whether to support Class B
classBTimeout	Integer	Class B ACK timeout
pingSlotPeriod	Integer	PingSlot periodicity for Class B
pingSlotDR	Integer	PingSlot Datarate for Class B
pingSlotFreq	Integer	PingSlot Frequency for Class B, unit: Hz
supportsClassC	Boolean	Whether to support Class C
classCTimeout	Integer	Class C ACK timeout
macVersion	String	LoRaWAN MAC version
regParamsRevision	String	Regional Parameters Revision
rxDROffset1	Integer	RX1 Datarate offset
rxDataRate2	Integer	RX2 Datarate
rxFreq2	Integer	RX2 channel frequency, unit: Hz
factoryPresetFreqs	Object[]	Frequency list

maxEIRP	Integer	Max output power, 0 means maximum
maxDutyCycle	Integer	0
supportsJoin	Boolean	Whether to support OTAA
rfRegion	String	
supports32bitFCnt	Boolean	
enableUplinkChannels	Array []	Enabled uplink channels
name	String	Profile name
organizationID	String	1
networkServerID	String	1
using	Boolean	Whether this profile is used by devices
isDefault	Boolean	Whether this profile is default setting
channelPlan	String	

### Request Example

```
https://192.168.23.150:8080/api/urprofiles?limit=10&offset=0&organizationID=1&applicationID=1
```

### Response Example

```
{
  "totalCount": "8",
  "disable": false,
  "result": [
    {
      "profile": {
        "profileID": "eba41845-c435-4912-8d65-e167ca0e92e4",
        "supportsClassB": false,
        "classBTimeout": 10,
        "pingSlotPeriod": 128,
        "pingSlotDR": 3,
        "pingSlotFreq": 869525000,
        "supportsClassC": false,
        "classCTimeout": 10,
        "macVersion": "1.0.2",
        "regParamsRevision": "B",
        "rxDROffset1": 0,
        "rxDataRate2": 0,
        "rxFreq2": 869525000,
        "factoryPresetFreqs": [],
        "maxEIRP": 0,
        "maxDutyCycle": 0,
        "supportsJoin": false,
        "rfRegion": "EU868",
        "supports32bitFCnt": true,

```



```

        "enableUplinkChannels": []
    },
    "name": "ClassA-ABP",
    "organizationID": "1",
    "networkServerID": "1",
    "using": false,
    "isDefault": true
},
{
    "profile": {
        "profileID": "04c34d5a-11eb-4504-8180-fc197c9a6752",
        "supportsClassB": false,
        "classBTimeout": 10,
        "pingSlotPeriod": 128,
        "pingSlotDR": 3,
        "pingSlotFreq": 869525000,
        "supportsClassC": false,
        "classCTimeout": 10,
        "macVersion": "1.0.2",
        "regParamsRevision": "B",
        "rxDROffset1": 0,
        "rxDataRate2": 0,
        "rxFreq2": 869525000,
        "factoryPresetFreqs": [],
        "maxEIRP": 0,
        "maxDutyCycle": 0,
        "supportsJoin": true,
        "rfRegion": "EU868",
        "supports32bitFCnt": true,
        "enableUplinkChannels": []
    },
    "name": "ClassA-OTAA",
    "organizationID": "1",
    "networkServerID": "1",
    "using": true,
    "isDefault": true
},
.....
],
"channelPlan": "EU868"
}

```

## 6.2 Create Profile

**Request Method:** POST

**Request Address:**

https://{gatewayIP}:8080/api/urprofiles

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

Body

Parameter	Type	Required	Description
profile	Object[]	Yes	
supportsClassB	Boolean	No	Whether to support Class B
classBTimeout	Integer	No	Class B ACK timeout, range: 0-10s
pingSlotPeriod	Integer	No	PingSlot periodicity for Class B Option: 32: every second 64: every 2 second 128: every 4 second 256: every 8 second 512: every 16 second 1024: every 32 second 2048: every 64 second 4096: every 128 second
pingSlotDR	Integer	No	PingSlot Datarate for Class B, refer to <a href="#">Default RX2/PingSlot Settings</a>
pingSlotFreq	Integer	No	PingSlot Frequency for Class B, unit: Hz refer to <a href="#">Default RX2/PingSlot Settings</a>
supportsClassC	Boolean	No	Whether to support Class C
classCTimeout	Integer	No	Class C ACK timeout, range: 0-10s
macVersion	String	No	LoRaWAN MAC version: 1.0.0, 1.0.1, 1.0.2, 1.1.0
regParamsRevision	String	No	Regional Parameters Revision: A or B
rxDROffset1	Integer	No	RX1 Datarate offset, range: 0-5
rxDataRate2	Integer	No	RX2 Datarate, refer to <a href="#">Default RX2/PingSlot Settings</a>
rxFreq2	Integer	No	RX2 channel frequency, unit: Hz refer to <a href="#">Default RX2/PingSlot Settings</a>
factoryPresetFreqs	Array []	No	Frequency list
maxEIRP	Integer	No	Max output power, 0 means maximum
maxDutyCycle	Integer	No	0
supportsJoin	Boolean	No	Whether to support OTAA

rfRegion	String	No	Chanel plan
supports32bitFCnt	Boolean	Yes	true
enableUplinkChannels	Array []	No	Uplink channel number <b>Note:</b> only CN470/US915/AU915 supports this parameter, and at least 2 channels need to add for US915/AU915.
name	String64	Yes	Profile name, only supports letter, number, “-” and “_”, the first character must be letter
organizationID	String	Yes	1

### Response Parameters

Parameter	Type	Description
profileID	String	Return this ID when creating a profile with success
error	String	Error content
code	Integer	Error code, return 200 when success

### Request Example

```
{
  "profile": {
    "supportsClassB": false,
    "classBTimeout": 10,
    "pingSlotPeriod": 128,
    "pingSlotDR": 3,
    "pingSlotFreq": 869525000,
    "supportsClassC": false,
    "classCTimeout": 10,
    "macVersion": "1.0.2",
    "regParamsRevision": "B",
    "rxDROffset1": 0,
    "rxDataRate2": 0,
    "rxFreq2": 869525000,
    "factoryPresetFreqs": [],
    "maxEIRP": 0,
    "maxDutyCycle": 0,
    "supportsJoin": true,
    "rfRegion": "EU868",
    "supports32bitFCnt": true,
    "enableUplinkChannels": []
  },
  "name": "profile1",
  "organizationID": "1"
}
```

### Response Example

```
{
  "profileID": "545c6b85-6daf-4eec-99a6-198e76b47cb8"
}
```

## 6.3 Update Profile

**Request Method:** PUT

**Request Address:**

https://{gatewayIP}:8080/api/urprofiles/{profileID}

Params

Parameter	Type	Required	Description
profileID	string	Yes	The profile ID need to update, this ID can be searched by get method

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

Body

Parameter	Type	Required	Description
name	String64	No	Profile name, only supports letter, number, "-" and "_", the first character must be letter
profile	Object[]	No	
supporsClassB	Boolean	No	Whether to support Class B
classBTimeout	Integer	No	Class B ACK timeout, range: 0-10s
pingSlotPeriod	Integer	No	PingSlot periodicity for Class B Option: 32: every second 64: every 2 second 128: every 4 second 256: every 8 second 512: every 16 second 1024: every 32 second 2048: every 64 second 4096: every 128 second
pingSlotDR	Integer	No	PingSlot Datarate for Class B, refer to <a href="#">Default RX2/PingSlot Settings</a>
pingSlotFreq	Integer	No	PingSlot Frequency for Class B, unit: Hz refer to <a href="#">Default RX2/PingSlot Settings</a>
supportsClassC	Boolean	No	Whether to support Class C
classCTimeout	Integer	No	Class C ACK timeout, range: 0-10s
macVersion	String	No	LoRaWAN MAC version: 1.0.0, 1.0.1, 1.0.2, 1.1.0
regParamsRevision	String	No	Regional Parameters Revision: A or B

rxDROffset1	Integer	No	RX1 Datarate offset, range: 0-5
rxDataRate2	Integer	No	RX2 Datarate, refer to <a href="#">Default RX2/PingSlot Settings</a>
rxFreq2	Integer	No	RX2 channel frequency, unit: Hz refer to <a href="#">Default RX2/PingSlot Settings</a>
factoryPresetFreqs	Array []	No	Frequency list
maxEIRP	Integer	No	Max output power, 0 means maximum
maxDutyCycle	Integer	No	0
supportsJoin	Boolean	No	Whether to support OTAA
rfRegion	String	No	Chanel plan
supports32bitFCnt	Boolean	Yes	true
enableUplinkChannels	Array []	No	Uplink channel number <b>Note:</b> only CN470/US915/AU915 supports this parameter, and at least 2 channels need to add for US915/AU915.

### Response Parameters

#### Success:

```
{}
```

#### Failure:

Parameter	Type	Description
error	String	Error content
code	Integer	Error code

### Request Example

```
https://192.168.23.150:8080/api/urprofiles/545c6b85-6daf-4eec-99a6-198e76b47cb8
```

#### Body

```
{
  "name": "profile2"
}
```

### Response Example

```
{}
```

## 6.4 Delete Profile

**Request Method:** DELETE

#### Request Address:

```
https://{gatewayIP}:8080/api/urprofiles/{profileID}
```

#### Params

Parameter	Type	Required	Description
-----------	------	----------	-------------

profileID	string	Yes	The profile ID need to delete, this ID can be searched by get method
-----------	--------	-----	--

### Request Parameters

#### Header

Parameter	Required	Description
Content-Type	Yes	application/json

### Response Parameters

#### Success:

{}

#### Failure:

Parameter	Type	Description
error	String	Error content
code	Integer	Error code

### Request Example

https://192.168.23.150:8080/api/urprofiles/545c6b85-6daf-4eec-99a6-198e76b47cb8

### Response Example

{}

## 7. Device

### 7.1 Get Device

**Request Method:** GET

**Request Address:**

https://{gatewayIP}:8080/api/urdevices

#### Params

Parameter	Type	Required	Description
limit	String	Yes	Max number of payload codecs to return in the result-test.
offset	String	Yes	Offset in the result-set (for pagination).
OrganizationID	String	Yes	1
applicationID	String	Yes	Devices under this application ID, 0 means all devices
search	String	No	Search device via device name or dev EUI

### Response Parameters

Parameter	Type	Description
devTotalCount	String	Total device number of selected application
deviceResult	Array[]	
devEUI	String	Device EUI.
name	String	Device name.
applicationID	String	The ID of the selected application.
appName	String	The name of the selected application.
description	String	The description of this device.
profileID	String	The profile ID of this device.
profileName	String	The name of the selected profile.
fCntUp	Integer	Uplink Frame-counter
fCntDown	Integer	Downlink Frame-counter
skipFCntCheck	Boolean	Whether to enable frame-counter validation
appKey	String	Application Key
devAddr	String	Device Address.
appSKey	String	Application Session Key.
nwkSKey	String	Network Session Key.
supportsJoin	Boolean	Whether to support OTAA join mode.
active	Boolean	Whether this device is activated
lastSeenAt	String	Show the time of last packet transmitted.
lastSeenAtTime	String	Show the specific time of last packet transmitted.
mbMode	String	Modbus RTU data transmission mode
mbFramePort	String	Modbus RTU fPort
mbTCPPort	String	TCP Port
fPort	integer	Downlink command port
payloadCodecID	String	Payload Codec ID
payloadName	String	Payload Codec name
appTotalCount	String	Total application number
appResult	Array[]	The result of searched application
appName	String	
applicationID	String	
pfTotalCount	String	
profileResult	Array[]	Supported profile
profileID	String	
profileName	String	
SupportsJoin	Boolean	Whether to support OTAA mode
channelPlan	String	
disable	Boolean	True or false

### Request Example

```
https://192.168.23.150:8080/api/urdevices?limit=10&offset=0&organizationID=1&applicationID=0
```

## Response Example

```
{
  "devTotalCount": "1",
  "deviceResult": [
    {
      "devEUI": "24E124809E080562",
      "name": "TS201",
      "applicationID": "1",
      "appName": "cloud1",
      "description": "test",
      "profileID": "04c34d5a-11eb-4504-8180-fc197c9a6752",
      "profileName": "ClassA-OTAA",
      "fCntUp": 4017,
      "fCntDown": 452,
      "skipFCntCheck": false,
      "appKey": "5572404c696e6b4c6f52613230313823",
      "devAddr": "07f3e032",
      "appSKey": "552d8cbaf3c9ada601d3ccc506985eb2",
      "nwkSKey": "868dcdec5bfdaf9a2da0cb05029b2d52",
      "supportsJoin": true,
      "active": true,
      "lastSeenAt": "1 minute ago",
      "mbMode": "",
      "mbFramePort": "0",
      "mbTcpPort": "0",
      "lastSeenAtTime": "2024-05-21 21:15:13.686038 +0800 CST",
      "fPort": 1,
      "payloadCodecID": "95",
      "payloadName": "TS201"
    }
  ],
  "appTotalCount": "2",
  "appResult": [
    {
      "appName": "cloud1",
      "applicationID": "1"
    }
  ],
  "pfTotalCount": "8",
  "profileResult": [
    {
      "profileID": "eba41845-c435-4912-8d65-e167ca0e92e4",
      "profileName": "ClassA-ABP",
      "supportsJoin": false
    }
  ]
}
```



```
    },  
    {  
      "profileID": "04c34d5a-11eb-4504-8180-fc197c9a6752",  
      "profileName": "ClassA-OTAA",  
      "supportsJoin": true  
    },  
    {  
      "profileID": "798bd818-e283-411b-aa5d-de5871b0ada4",  
      "profileName": "ClassB-ABP",  
      "supportsJoin": false  
    },  
    {  
      "profileID": "9ea4556c-7a7f-4432-bda2-b944c1ce3aa6",  
      "profileName": "ClassB-OTAA",  
      "supportsJoin": true  
    },  
    {  
      "profileID": "aa3e75a5-46cf-4b8f-90f1-766fea61b291",  
      "profileName": "ClassC-ABP",  
      "supportsJoin": false  
    },  
    {  
      "profileID": "6ea5f177-de37-4e79-9e0c-63545946f4c4",  
      "profileName": "ClassC-OTAA",  
      "supportsJoin": true  
    },  
    {  
      "profileID": "956189ca-cde4-46c8-904f-439a499c1f5e",  
      "profileName": "ClassCB-ABP",  
      "supportsJoin": false  
    },  
    {  
      "profileID": "2affb22c-0187-4e2b-9f6d-2b3a04239d10",  
      "profileName": "ClassCB-OTAA",  
      "supportsJoin": true  
    }  
  ],  
  "channelPlan": "EU868",  
  "disable": false  
}
```

## 7.2 Create Device

**Request Method:** POST

**Request Address:**

`https://{gatewayIP}:8080/api/urdevices`

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

Body

Parameter	Type	Required	Description
devEUI	String	Yes	Device EUI.
name	String64	No	A unique device name.
applicationID	String	Yes	The ID of the selected application.
description	String	No	The description of this device.
profileID	String	No	The profile ID of this device.
fCntUp	Integer	No	Uplink Frame-counter
fCntDown	Integer	No	Downlink Frame-counter
skipFCntCheck	Boolean	No	Whether to enable frame-counter validation
appKey	String	No	32bit application Key. Necessary to type when join type is OTAA.
devAddr	String	Yes	Device Address. Necessary to type when join type is ABP.
appSKey	String	No	32 bit application Session Key. Necessary to type when join type is ABP.
nwkSKey	String	No	32 bit network Session Key. Necessary to type when join type is ABP.
mbMode	String	No	Option: tcp, rtu tcp: Modbus RTU to TCP rtu: Modbus RTU over TCP
mbFramePort	String	No	Modbus RTU fPort, only works when mbMode is typed, range: 2-84, 86-223
mbTCPPort	String	No	TCP Port, only works when mbMode is typed Range: 1-65535
fPort	integer	No	Downlink command port, range: 1-233
payloadCodecID	String	No	Payload Codec ID, this can be searched via payload codec API. ID=0 means None.

### Response Parameters

Parameter	Type	Description
error	String	Error content

code	Integer	Error code, return 200 when success
------	---------	-------------------------------------

### Request Example

```
{
  "devEUI": "24e124136e091081",
  "name": "em300-test",
  "applicationID": "1",
  "description": "test",
  "profileID": "04c34d5a-11eb-4504-8180-fc197c9a6752",
  "skipFCntCheck": false,
  "appKey": "5572404C696E6B4C6F52613230313823",
  "fPort": 85,
  "payloadCodecID": "0"
}
```

### Response Example

```
{
  "code": 200,
  "error": ""
}
```

## 7.3 Update Device

**Request Method:** PUT

**Request Address:**

https://{gatewayIP}:8080/api/urdevices/{devEUI}

Params

Parameter	Type	Required	Description
devEUI	String	Yes	The devEUI need to update

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

Body

Parameter	Type	Required	Description
name	String64	Yes	A unique device name.
description	String	Yes	The description of this device.
profileID	String	Yes	The profile ID of this device.
fCntUp	Integer	No	Uplink Frame-counter
fCntDown	Integer	No	Downlink Frame-counter

skipFCntCheck	Boolean	Yes	Whether to enable frame-counter validation
appKey	String	No	32bit application Key. Necessary to type when join type is OTAA.
devAddr	String	No	Device Address. Necessary to type when join type is ABP.
appSKey	String	No	32 bit application Session Key. Necessary to type when join type is ABP.
nwkSKey	String	No	32 bit network Session Key. Necessary to type when join type is ABP.
mbMode	String	No	Option: tcp, rtu tcp: Modbus RTU to TCP rtu: Modbus RTU over TCP
mbFramePort	String	No	Modbus RTU fPort, only works when mbMode is typed, range: 2-84, 86-223
mbTCPPort	String	No	TCP Port, only works when mbMode is typed Range: 1-65535
fPort	integer	Yes	Downlink command port, range: 1-233
payloadCodecID	String	Yes	Payload Codec ID, this can be searched via payload codec API. ID=0 means None.

**Note: when updating the parameter of a device, it is necessary to include all required settings into request.**

### Response Parameters

Parameter	Type	Description
error	String	Error content
code	Integer	Error code, return 200 when success

### Request Example

<https://192.168.23.150:8080/api/urdevices/24e124136e091081>

Body

```
{
  "name": "em300",
  "applicationID": "1",
  "description": "test",
  "profileID": "04c34d5a-11eb-4504-8180-fc197c9a6752",
  "skipFCntCheck": false,
  "fPort": 85,
  "payloadCodecID": "0"
}
```

### Response Example

```
{
  "code": 200,
```

```
"error": ""  
}
```

## 7.4 Delete Device

**Request Method:** DELETE

**Request Address:**

```
https://{gatewayIP}:8080/api/urdevices/{devEUI}
```

Params

Parameter	Type	Required	Description
devEUI	String	Yes	The devEUI need to delete

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

### Response Parameters

**Success:**

```
{}
```

**Failure:**

Parameter	Type	Description
error	String	Error content
code	Integer	Error code

### Request Example

```
https://192.168.23.150:8080/api/urdevices/24e124136e091081
```

### Response Example

```
{}
```

## 8. Multicast Group

### 8.1 Get Multicast Group

**Request Method:** GET

**Request Address:**

https://{gatewayIP}:8080/api/multicast-groups

Params

Parameter	Type	Required	Description
limit	String	Yes	Max number of profiles to return in the result-test.
offset	String	Yes	Offset in the result-set (for pagination).
organizationID	String	Yes	1
devEUI	String	No	Search the group which include the device EUI
search	String	No	multicast group name

#### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

#### Response Parameters

Parameter	Type	Description
totalCount	String	Total number of multicast groups
Result	Array[]	
id	String	Multicast group ID
name	String	Multicast group name
mcAddr	String	Multicast device Address
mcNwkSKey	String	Multicast Network Session Key
mcAppSKey	String	Multicast application Session Key
fCnt	Integer	Frame-counter
groupType	String	Class type
dr	Integer	datarate
frequency	Integer	Downlink frequency, unit: Hz
pingSlotPeriod	Integer	Ping Slot periodicity when class B type
createAt	String	Create time of this group
updateAt	String	Update time of this group
devices	Object[]	Device information in this group.
devEUI	String	Device EUI.

name	String	Device name.
applicationID	String	The ID of the selected application.
appName	String	The name of the selected application.
description	String	The description of this device.
deviceprofileID	String	The profile ID of this device.
deviceprofileName	String	The name of the profile.
deviceStatusBattery	Integer	0: external power device 1-254: battery level 255: the device was not able to measure the battery level 256: the device status is not available
deviceStatusMargin	Integer	-32 - 32: the demodulation SNR ration in DB 256: the device status is not available
lastSeenAt	String	Last time to receive the message from this device

**Request Example:**

```
https://192.168.23.150:8080/api/multicast-groups?limit=10&offset=0&organizationID=1
```

**Response Example:**

```
{
  "totalCount": "1",
  "result": [
    {
      "id": "66987800-44e5-47b5-bcf1-eb1ff5d9bfd",
      "name": "Device Control",
      "mcAddr": "11111111",
      "mcNwkSKey": "5572404c696e6b4c6f52613230313823",
      "mcAppSKey": "5572404c696e6b4c6f52613230313823",
      "fCnt": 0,
      "groupType": "CLASS_C",
      "dr": 0,
      "frequency": 869525000,
      "pingSlotPeriod": 128,
      "createdAt": "2024-05-23 16:19:35",
      "updatedAt": "2024-05-23 16:19:35",
      "devices": [
        {
          "devEUI": "24E124809E080562",
          "name": "WS558",
          "applicationID": "1",
          "description": "test",
          "deviceProfileID": "6ea5f177-de37-4e79-9e0c-63545946f4c4",
          "deviceProfileName": "ClassC-OTAA",
          "deviceStatusBattery": 0,

```

```

        "deviceStatusMargin": 0,
        "lastSeenAt": ""
    }
}
]
}
]
}

```

## 8.2 Get Multicast Group by ID

**Request Method:** GET

**Request Address:**

<https://{gatewayIP}:8080/api/multicast-groups/{id}>

Params

Parameter	Type	Required	Description
id	String	Yes	Multicast group ID

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

### Response Parameters

Parameter	Type	Description
multicastGroup	Array[]	
id	String	Multicast group ID
name	String	Multicast group name
mcAddr	String	Multicast device Address
mcNwkSKey	String	Multicast Network Session Key
mcAppSKey	String	Multicast application Session Key
fCnt	Integer	Frame-counter
groupType	String	Class type
dr	Integer	datarate
frequency	Integer	Downlink frequency, unit: Hz
pingSlotPeriod	Integer	Ping Slot periodicity when class B type
createAt	String	Create time of this group
updateAt	String	Update time of this group

**Request Example:**

<https://192.168.23.150:8080/api/multicast-groups/66987800-44e5-47b5-bcf1-eb1ff5d9bfdf>



**Response Example:**

```
{
  "multicastGroup": {
    "id": "66987800-44e5-47b5-bcf1-eb1ff5d9bdfd",
    "name": "Device Control",
    "mcAddr": "11111111",
    "mcNwkSKey": "5572404c696e6b4c6f52613230313823",
    "mcAppSKey": "5572404c696e6b4c6f52613230313823",
    "fCnt": 0,
    "groupType": "CLASS_C",
    "dr": 0,
    "frequency": 869525000,
    "pingSlotPeriod": 128
  },
  "createdAt": "2024-05-23 16:19:35",
  "updatedAt": "2024-05-23 16:19:35"
}
```

## 8.3 Create Multicast Group

**Request Method:** POST**Request Address:**`https://{gatewayIP}:8080/api/multicast-groups`**Request Parameters**

Header

Parameter	Required	Description
Content-Type	Yes	application/json

Body

Parameter	Type	Required	Description
multicastGroup	Object[]	No	
name	String	Yes	A unique multicast group name
mcAddr	String	No	8-bit multicast device Address
mcNwkSKey	String	No	32-bit multicast Network Session Key
mcAppSKey	String	No	32-bit multicast application Session Key
fCnt	Integer	No	Frame-counter
groupType	String	No	Option: CLASS_C, CLASS_B
dr	Integer	No	Datarate, the same as pingslot datarate, refer to <a href="#">Default RX2/PingSlot Settings</a>
frequency	Integer	No	Downlink frequency, unit: Hz the same as pingslot frequency, refer to

			<u>Default RX2/PingSlot Settings</u>
pingSlotPeriod	Integer	No	Ping Slot periodicity when class B type Option: 32: every second 64: every 2 second 128: every 4 second 256: every 8 second 512: every 16 second 1024: every 32 second 2048: every 64 second 4096: every 128 second

### Response Parameters

Parameter	Type	Description
id	String	Return this ID when creating a multicast group with success
error	String	Error content
code	Integer	Error code, return 200 when success

### Request Example:

```
{
  "multicastGroup": {
    "name": "Device Control",
    "mcAddr": "11111111",
    "mcNwkSKey": "5572404c696e6b4c6f52613230313823",
    "mcAppSKey": "5572404c696e6b4c6f52613230313823",
    "fCnt": 0,
    "groupType": "CLASS_C",
    "dr": 0,
    "frequency": 869525000,
    "pingSlotPeriod": 128
  }
}
```

### Response Example:

```
{
  "id": "b06971fb-1aa0-493b-9319-a897078f5400"
}
```

## 8.4 Update Multicast Group

**Request Method:** PUT

**Request Address:**

https://{gatewayIP}:8080/api/multicast-groups/{id}

Params

Parameter	Type	Required	Description
id	String	Yes	Multicast group ID

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

Body

Parameter	Type	Required	Description
multicastGroup	Object[]	Yes	
name	String	Yes	A unique multicast group name
mcAddr	String	Yes	8-bit multicast device Address
mcNwkSKey	String	Yes	32-bit multicast Network Session Key
mcAppSKey	String	Yes	32-bit multicast application Session Key
fCnt	Integer	Yes	Frame-counter
groupType	String	Yes	Option: CLASS_C, CLASS_B
dr	Integer	Yes	datarate, the same as pingslot datarate, refer to <a href="#">Default RX2/PingSlot Settings</a>
frequency	Integer	Yes	Downlink frequency, unit: Hz the same as pingslot frequency, refer to <a href="#">Default RX2/PingSlot Settings</a>
pingSlotPeriod	Integer	Yes	Ping Slot periodicity when class B type Option: 32: every second 64: every 2 second 128: every 4 second 256: every 8 second 512: every 16 second 1024: every 32 second 2048: every 64 second 4096: every 128 second

**Note:** when updating the parameter of a device, it is necessary to include all settings into request.

### Response Parameters

**Success:**

{}

**Failure:**

Parameter	Type	Description
error	String	Error content
code	Integer	Error code

**Request Example:**

```
{
  "multicastGroup": {
    "name": "Device Control3",
    "mcAddr": "11111111",
    "mcNwkSKey": "5572404c696e6b4c6f52613230313823",
    "mcAppSKey": "5572404c696e6b4c6f52613230313823",
    "fCnt": 0,
    "groupType": "CLASS_C",
    "dr": 0,
    "frequency": 869525000,
    "pingSlotPeriod": 128
  }
}
```

**Response Example:**

```
{}
```

## 8.5 Delete Multicast Group

**Request Method:** DELETE**Request Address:**

```
https://{gatewayIP}:8080/api/multicast-groups/{id}
```

Params

Parameter	Type	Required	Description
id	String	Yes	The multicast group ID need to delete

**Request Parameters**

Header

Parameter	Required	Description
Content-Type	Yes	application/json

**Response Parameters**

**Success:**

{}

**Failure:**

Parameter	Type	Description
error	String	Error content
code	Integer	Error code

### Request Example

```
https://192.168.23.150:8080/api/multicast-groups/b06971fb-1aa0-493b-9319-a897078f5400
```

### Response Example

{}

## 8.6 Get Devices in Multicast Group

**Request Method:** GET

**Request Address:**

```
https://{gatewayIP}:8080/api/multicast-groups/{id}/devices
```

Params

Parameter	Type	Required	Description
limit	String	Yes	Max number of devices to return in the result-test.
offset	String	Yes	Offset in the result-set (for pagination).
id	String	Yes	Multicast group ID

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

### Response Parameters

Parameter	Type	Description
totalCount	String	Total number of multicast groups
Result	Array[]	
devEUI	String	Device EUI.
name	String	Device name.
applicationID	String	The ID of the selected application.
appName	String	The name of the selected application.
description	String	The description of this device.

deviceprofileID	String	The profile ID of this device.
deviceprofileName	String	The name of the profile.
deviceStatusBattery	Integer	0: external power device 1-254: battery level 255: the device was not able to measure the battery level 256: the device status is not available
deviceStatusMargin	Integer	-32 - 32: the demodulation SNR ration in DB 256: the device status is not available
lastSeenAt	String	Last time to receive the message from this device

**Request Example:**

```
https://192.168.23.150:8080/api/multicast-groups/66987800-44e5-47b5-bcf1-eb1ff5d9bdfd/devices?limit=10&offset=0
```

**Response Example:**

```
{
  "totalCount": "1",
  "result": [
    {
      "devEUI": "24E124809E080562",
      "name": "TS201",
      "applicationID": "1",
      "description": "test",
      "deviceProfileID": "04c34d5a-11eb-4504-8180-fc197c9a6752",
      "deviceProfileName": "ClassA-OTAA",
      "deviceStatusBattery": 0,
      "deviceStatusMargin": 0,
      "lastSeenAt": "2024-05-23 17:27:12"
    }
  ]
}
```

## 8.7 Add a Device to Multicast Group

**Request Method:** POST**Request Address:**

```
https://{gatewayIP}:8080/api/multicast-groups/{id}/device
```

Params

Parameter	Type	Required	Description
id	String	Yes	Multicast group ID

**Request Parameters**

## Header

Parameter	Required	Description
Content-Type	Yes	application/json

## Body

Parameter	Type	Required	Description
devEUI	String	Yes	Device EUI need to add to this group

**Response Parameters**

## Success:

```
{}
```

## Failure:

Parameter	Type	Description
error	String	Error content
code	Integer	Error code

**Request Example:**

```
https://192.168.23.150:8080/api/multicast-groups/b06971fb-1aa0-493b-9319-a897078f5400/device
```

## Body

```
{
  "devEUI": "24E124809E080562"
}
```

**Response Example:**

```
{}
```

## 8.8 Add Multiple Devices to Multicast Group

**Request Method:** POST**Request Address:**

```
https://{gatewayIP}:8080/api/multicast-groups/{id}/devices
```

## Params

Parameter	Type	Required	Description
id	String	Yes	Multicast group ID

**Request Parameters**

## Header

Parameter	Required	Description
Content-Type	Yes	application/json

## Body

Parameter	Type	Required	Description
-----------	------	----------	-------------

devEUIs	Array[]	Yes	Device EUIs need to add to this group
---------	---------	-----	---------------------------------------

**Response Parameters****Success:**

{}

**Failure:**

Parameter	Type	Description
error	String	Error content
code	Integer	Error code

**Request Example:**

```
https://192.168.23.150:8080/api/multicast-groups/b06971fb-1aa0-493b-9319-a897078f5400/devices
```

Body

```
{
  "devEUIs":
    ["24E124809E080562","6136D41480612305"]
}
```

**Response Example:**

{}

## 8.9 Remove Devices from Multicast Group

**Request Method:** POST**Request Address:**

```
https://{gatewayIP}:8080/api/multicast-groups/{id}/deletedevices
```

Params

Parameter	Type	Required	Description
id	String	Yes	Multicast group ID

**Request Parameters**

Header

Parameter	Required	Description
Content-Type	Yes	application/json

Body

Parameter	Type	Required	Description
devEUIs	Array[]	Yes	Device EUIs need to remove from this group

**Response Parameters****Success:**



```
{}
```

**Failure:**

Parameter	Type	Description
error	String	Error content
code	Integer	Error code

**Request Example:**

```
https://192.168.23.150:8080/api/multicast-groups/b06971fb-1aa0-493b-9319-a897078f5400/deletedevices
```

**Body**

```
{
  "devEUIs":
    ["24E124809E080562","6136D41480612305"]
}
```

**Response Example:**

```
{}
```

## 8.10 Remove a Device from Multicast Group by Device EUI

**Request Method:** DELETE**Request Address:**

```
https://{gatewayIP}:8080/api/multicast-groups/{id}/device/[devEUI]
```

**Params**

Parameter	Type	Required	Description
id	String	Yes	Multicast group ID
devEUI	String	Yes	Device EUI in this group

**Request Parameters****Header**

Parameter	Required	Description
Content-Type	Yes	application/json

**Response Parameters****Success:**

```
{}
```

**Failure:**

Parameter	Type	Description
error	String	Error content
code	Integer	Error code

**Request Example:**

```
https://192.168.23.150:8080/api/multicast-groups/b06971fb-1aa0-493b-9319-a897078f540
```

0/device/24E124809E080562

**Response Example:**

```
{}
```

## 9. Gateway Fleet

### 9.1 Get Gateway List

**Request Method:** GET**Request Address:**

https://{gatewayIP}:8080/api/gateways

Params

Parameter	Type	Required	Description
limit	Integer	Yes	Max number of profiles to return in the result-test.
offset	Integer	Yes	Offset in the result-set (for pagination).
organizationID	String	Yes	1
search	String	No	Search by name or gateway ID

**Request Parameters**

Header

Parameter	Required	Description
Content-Type	Yes	application/json

**Response Parameters**

Parameter	Type	Description
totalCount	String	Total number of gateways
localNS	Boolean	Whether NS mode of main gateway is enabled
Result	Array[]	
mac	String	Gateway ID
name	String	Gateway name
description	String	
createAt	String	Ttime to add this gateway
updateAt	String	Time to update this gateway
organizationID	String	1
networkServerID	String	1
firstSeenAt	String	First time to receive data from this gateway
lastSeenAt	String	Last time to receive data from this gateway
latitude	Number	latitude of gateway

longitude	Number	longitude of gateway
altitude	Number	Altitude of gateway
connected	Boolean	Whether connection status is connected

**Request Example:**

```
https://192.168.23.150:8080/api/gateways?limit=10&offset=0&organizationID=1
```

**Response Example:**

```
{
  "totalCount": 1,
  "localNS": true,
  "result": [
    {
      "mac": "24E124FFFEF1272B",
      "name": "Local Gateway",
      "description": "Local LoRa Gateway",
      "createdAt": "2024-05-10 20:01:52",
      "updatedAt": "2024-05-10 20:01:52",
      "organizationID": "1",
      "networkServerID": "1",
      "firstSeenAt": "2024-05-10 20:02:33",
      "lastSeenAt": "2024-05-23 19:30:26",
      "latitude": 0,
      "longitude": 0,
      "altitude": 0,
      "connected": true
    }
  ]
}
```

## 9.2 Get Gateway by Gateway ID

**Request Method:** GET**Request Address:**

```
https://{gatewayIP}:8080/api/gateways/{mac}
```

Params

Parameter	Type	Required	Description
mac	String	Yes	Gateway ID

**Request Parameters**

Header

Parameter	Required	Description
Content-Type	Yes	application/json

### Response Parameters

Parameter	Type	Description
mac	String	Gateway ID
name	String	Gateway name
description	String	
createAt	String	Ttime to add this gateway
updateAt	String	Time to update this gateway
organizationID	String	1
networkServerID	String	1
firstSeenAt	String	First time to receive data from this gateway
lastSeenAt	String	Last time to receive data from this gateway
latitude	Number	latitude of gateway
longitude	Number	longitude of gateway
altitude	Number	Altitude of gateway
connected	Boolean	Whether connection status is connected
ping	Boolean	Whether to send a periodic ping from gateway
gatewayProfileID	String	

### Request Example:

<https://192.168.23.150:8080/api/gateways/24E124FFFEF1272B>

### Response Example:

```
{
  "mac": "24E124FFFEF1272B",
  "name": "Local Gateway",
  "description": "Local LoRa Gateway",
  "latitude": 0,
  "longitude": 0,
  "altitude": 0,
  "createdAt": "2024-05-10 20:01:52",
  "updatedAt": "2024-05-10 20:01:52",
  "firstSeenAt": "2024-05-10 20:02:33",
  "lastSeenAt": "2024-05-23 19:47:26",
  "organizationID": "1",
  "ping": false,
  "networkServerID": "1",
  "gatewayProfileID": "",
  "connected": true
}
```

## 9.3 Create Gateway

**Request Method:** POST

**Request Address:**

https://{gatewayIP}:8080/api/gateways

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

Body

Parameter	Type	Required	Description
mac	String	Yes	Gateway ID
name	String	Yes	Gateway name, only letters, digits, "-" and "_" are allowed, the first character must be letter
description	String	No	
organizationID	String	Yes	1
networkServerID	String	Yes	1
gatewayProfileID	String	No	
latitude	Number	No	latitude of gateway
longitude	Number	No	longitude of gateway
altitude	Number	No	Altitude of gateway
ping	Boolean	No	Whether to send a periodic ping from gateway

### Response Parameters

**Success:**

{}

**Failure:**

Parameter	Type	Description
error	String	Error content
code	Integer	Error code

### Request Example:

```
{
  "mac": "24E124FFFEF54092",
  "name": "Local-Gateway1",
  "organizationID": "1",
  "networkServerID": "1"
}
```

**Response Example:**

```
{}
```

## 9.4 Update Gateway

**Request Method:** PUT**Request Address:**

```
https://{gatewayIP}:8080/api/gateways/{mac}
```

Params

Parameter	Type	Required	Description
mac	String	Yes	Gateway ID need to update

**Request Parameters**

Header

Parameter	Required	Description
Content-Type	Yes	application/json

Body

Parameter	Type	Required	Description
name	String	Yes	Gateway name, only letters, digits, "-" and "_" are allowed, the first character must be letter
description	String	Yes	
organizationID	String	Yes	1
latitude	Number	Yes	latitude of gateway
longitude	Number	Yes	longitude of gateway
altitude	Number	Yes	Altitude of gateway
ping	Boolean	Yes	Whether to send a periodic ping from gateway

**Note: when updating the parameter of a device, it is necessary to include all settings into request.**

**Response Parameters****Success:**

```
{}
```

**Failure:**

Parameter	Type	Description
error	String	Error content
code	Integer	Error code

**Request Example:**

```
https://192.168.23.150:8080/api/gateways/24E124FFFEF1272B
```

Body

```
{
```

```
{
  "name": "Local-Gateway2",
  "description": "Local LoRa Gateway",
  "latitude": 0,
  "longitude": 0,
  "altitude": 0,
  "organizationID": "1",
  "ping": false
}
```

**Response Example:**

```
{}
```

## 9.5 Delete Gateway

**Request Method:** DELETE**Request Address:**

```
https://{gatewayIP}:8080/api/gateways/{mac}
```

**Params**

Parameter	Type	Required	Description
mac	String	Yes	Gateway ID need to update

**Request Parameters****Header**

Parameter	Required	Description
Content-Type	Yes	application/json

**Response Parameters****Success:**

```
{}
```

**Failure:**

Parameter	Type	Description
error	String	Error content
code	Integer	Error code

**Request Example:**

```
https://192.168.23.150:8080/api/gateways/24E124FFFEF1272B
```

**Response Example:**

```
{}
```

## 10. Packets

### 10.1 Get Packets

**Request Method:** GET

**Request Address:**

<https://{gatewayIP}:8080/api/urpackets>

Params

Parameter	Type	Required	Description
OrganizationID	String	Yes	1

**Response Parameters**

Parameter	Type	Description
totalCount	String	Total packet number displayed on the web GUI.
packets	Array[]	
frequency	Integer	Uplink or downlink frequency. Unit: Hz
power	String	Tx power of downlink packet. Unit: dBm
immediately	String	Whether the downlink packet must be sent immediately.
dataRate	String	The name of the selected application.
modulation	String	LoRa or FSK.
bandwidth	Integer	Unit: Hz
spreadFactor	Integer	
bitRate	Integer	Bit rate for FSK modulation.
codeRate	String	
gatewayMac	String	Gateway ID to send this packet.
timeSinceGPSEpoch	String	Time when the packet was received since GPS epoch.
timestamp	Integer	Gateway timestamp.
rsi	String	RSSI value for uplink packet.
loraSNR	String	SNR value for uplink packet.
devEUI	String	Device EUI.
time	String	Time when the uplink was received or the downlink was created.
type	String	Packet type.
fCnt	Integer	Frame counter.
devAddr	String	Device address or multicast address.
adr	String	Whether ADR is enabled.
adrAckReq	String	Whether need to send confirmed packet to device.
ack	String	Whether this packet is ACK packet.
mic	String	
appEUI	String	Device App EUI.



fPort	String	Uplink or downlink port
size	String	Payload size
payloadBase64	String	Base64 format payload content
payloadHex	String	Hex format payload content
classType	String	Device class type.
enqueue	Boolean	Whether this packet is within the enqueue
payloadJson	String	Payload content after decoded

### Request Example

```
https://192.168.23.150:8080/api/urpackets?&organizationID=1
```

### Response Example

```
{
  "packets":[
    {
      "frequency":868100000,
      "power":"-",
      "immediately":"-",
      "dataRate":"SF7BW125",
      "modulation":"LORA",
      "bandwidth":125,
      "spreadFactor":7,
      "bitRate":0,
      "codeRate":"4/5",
      "gatewayMac":"24E124FFFEF1272B",
      "timeSinceGPSEPOCH":"389028h53m30.034s",
      "timestamp":2377477814,
      "rssi":"-53",
      "loraSNR":"13.8",
      "devEUI":"24E124809E080562",
      "time":"2024-05-23T20:53:12+08:00",
      "type":"UpUnc",
      "fCnt":5496,
      "devAddr":"07F3E032",
      "adr":"true",
      "adrAckReq":"false",
      "ack":"false",
      "mic":"5954644f",
      "appEUI":"24E124C0002A0001",
      "fPort":85,
      "size":7,
      "payloadBase64":"AXVjA2cEAQ==",
      "payloadHex":"01756303670401",
      "enqueue":false,
    }
  ]
}
```

```

    "classType": "Class A",
    "payloadJson": "{ \"battery\": 99, \"temperature\": 26 }"
  },
  .....
],
"totalCount": "1000"
}

```

## 10.2 Delete All Stored Packets

**Request Method:** DELETE

**Request Address:**

https://{gatewayIP}:8080/api/urpackets

Params

Parameter	Type	Required	Description
organizationID	String	Yes	1

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

### Response Parameters

**Success:**

{}

**Failure:**

Parameter	Type	Description
error	String	Error content
code	Integer	Error code

### Request Example

https://192.168.23.150:8080/api/urpackets?organizationID=1

### Response Example

{}

# 11. Send Downlink

## 11.1 Send Downlink Command to Device

**Request Method:** POST

**Request Address:**

`https://{gatewayIP}:8080/api/devices/{devEUI}/queue`

Params

Parameter	Type	Required	Description
devEUI	String	Yes	Device EUI which need to send downlink

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

Body

Parameter	Type	Required	Description
confirmed	Boolean	No	
fPort	Integer	No	Downlink port
data	String	No	Base64 format downlink command
jsonObject	String	No	Decoded object (when application coded has been configured),when providing the 'jsonObject', you can omit 'data'

### Response Parameters

**Success:**

`{}`

**Failure:**

Parameter	Type	Description
error	String	Error content
code	Integer	Error code

### Request Example:

`https://192.168.23.150:8080/api/devices/24e124809e080562/queue`

Body

```
{
  "confirmed": true,
  "fPort": 85,
  "data": "/xD/"
}
```

OR

```
{
  "confirmed": true,
  "fPort": 85,
  "jsonObject": "{\"gpio_output_1\":\"1\"}"
}
```

**Response Example:**

```
{}
```

## 11.2 Get Device Downlink Command in Queue

**Request Method:** GET**Request Address:**

```
https://{gatewayIP}:8080/api/devices/{devEUI}/queue
```

Params

Parameter	Type	Required	Description
devEUI	String	Yes	Device EUI

**Request Parameters**

Header

Parameter	Required	Description
Content-Type	Yes	application/json

**Response Parameters**

Parameter	Type	Description
items	Array[]	
devEUI	String	Device EUI which send downlink command to
reference	String	Random reference (used on ack notification)
confirmed	Boolean	Whether to need confirmed
fPort	Integer	Downlink port
data	String	Base64 format downlink command
fCnt	Integer	Downlink frame count.

**Request Example:**

```
https://192.168.23.150:8080/api/devices/24e124809e080562/queue
```

**Response Example:**

```
{
  "items": [
    {
      "devEUI": "24e124809e080562",
```

```
{
  "reference": "",
  "confirmed": true,
  "fPort": 85,
  "data": "/44ABQA=",
  "fCnt": 26
}
```

## 11.3 Delete Device Downlink Command from Queue

**Request Method:** DELETE

**Request Address:**

`https://{gatewayIP}:8080/api/devices/{devEUI}/queue`

Params

Parameter	Type	Required	Description
devEUI	String	Yes	Device EUI which need to send delete downlink message in queue

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

### Response Parameters

**Success:**

`{}`

**Failure:**

Parameter	Type	Description
error	String	Error content
code	Integer	Error code

**Request Example:**

`https://192.168.23.150:8080/api/devices/24e124809e080562/queue`

**Response Example:**

`{}`

## 11.4 Send Downlink Command to Multicast Group

**Request Method:** POST

**Request Address:**

`https://{gatewayIP}:8080/api/multicast-groups/{id}/queue`

Params

Parameter	Type	Required	Description
id	String	Yes	Multicast group ID

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

Body

Parameter	Type	Required	Description
multicastQueueItem	Object[]	Yes	
fCnt	Integer	No	
fPort	Integer	Yes	Downlink port
data	String	Yes	Base64 format downlink command

### Response Parameters

Parameter	Type	Description
fCnt	Integer	Return fcnt of downlink command when success
error	String	Error content
code	Integer	Error code

### Request Example:

`https://192.168.23.150:8080/api/multicast-groups/b06971fb-1aa0-493b-9319-a897078f5400/device`

Body

```
{
  "multicastQueueItem": {
    "fPort": 85,
    "data": "/xD/"
  }
}
```

### Response Example:

```
{
  "fCnt": 1
}
```

## 11.5 Get Multicast Downlink Command in Queue

**Request Method:** GET

**Request Address:**

https://{gatewayIP}:8080/api/multicast-groups/{id}/queue

Params

Parameter	Type	Required	Description
id	String	Yes	Multicast group ID

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

### Response Parameters

Parameter	Type	Description
multicastQueueItems	Object[]	
multicastGroupID	String	
fCnt	Integer	
fPort	Integer	Downlink port
data	String	Base64 format downlink command

### Request Example:

https://192.168.23.150:8080/api/multicast-groups/b06971fb-1aa0-493b-9319-a897078f5400/device

### Response Example:

```
{
  "multicastQueueItems": [
    {
      "multicastGroupID": "b06971fb-1aa0-493b-9319-a897078f5400",
      "fPort": 85,
      "data": "/xD/"
    }
  ]
}
```

## 11.6 Delete Multicast Downlink Command from Queue

**Request Method:** DELETE

**Request Address:**

https://{gatewayIP}:8080/api/multicast-groups/{id}/queue

**Params**

Parameter	Type	Required	Description
id	String	Yes	Multicast group ID

**Request Parameters****Header**

Parameter	Required	Description
Content-Type	Yes	application/json

**Response Parameters****Success:**

{}

**Failure:**

Parameter	Type	Description
error	String	Error content
code	Integer	Error code

**Request Example:**

https://192.168.23.150:8080/api/multicast-groups/b06971fb-1aa0-493b-9319-a897078f5400/device

**Response Example:**

{}

## 12. FUOTA

### 12.1 Add FUOTA Task

**Request Method:** POST

**Request Address:**

https://{gatewayIP}:8080/api/fuota/task

**Request Parameters****Header**

Parameter	Required	Description
Content-Type	Yes	application/json

**Body**



Parameter	Type	Required	Description
fuotaTask	Object[]	Yes	FUOTA task information
name	String	Yes	Task name, length limit: 127
startedAt	String	Yes	The time to start this task
description	String	No	Task description
deveui	String[]	Yes	Device EUI list
firmwareInfo	Object[]	Yes	Firmware information
firmwareName	String	Yes	Firmware name
description	String	No	
fileContent	String	Yes	Firmware content (base64)
officialFirmwareUrl	String	No	Official firmware URL
fragmentInfo	Object[]	Yes	Firmware fragment info
fragmentSize	Integer	yes	Fragment size, range: 1-250, unit: byte
fragmentInterval	Integer	yes	Fragment interval, range: 1000-10000, unit: ms
fragmentRedundancy	Integer	yes	Fragment redundancy, range: 0-100, unit:%
tmpMcInfo	Object[]	Yes	Temperary multicast info
groupType	Integer	yes	0: class C
dr	Integer	yes	Datarate, refer to <a href="#">Default RX2/PingSlot Settings</a>
frequency	Integer	yes	Downlink frequency, unit: Hz refer to <a href="#">Default RX2/PingSlot Settings</a>
isOfficialFirmware	bool	No	Whether is official firmware
officialFirmwarInfo	Object[]	No	Get these info from API 12.10
id	String	Yes	Firmware ID
model	String	Yes	Product model
name	String	Yes	Firmware name
firmwareVersion	String	Yes	Firmware version
hardwareVersion	String	Yes	Hardware version
supportMinHardwareVersion	String	Yes	Supported min hardware version
supportMinFirmwareVersion	String	Yes	Supported min firmware version
url	String	Yes	Download url
md5	String	Yes	
crc32	String	Yes	
size	String	Yes	Firmware size
frequencyBands	String[]	Yes	Supported frequency
firmwareNotesCn	String	Yes	English note
firmwareNotesEn	String	Yes	Chinese note

## Response Parameters

Parameter	Type	Description
id	Integer	Return the task ID when creating a task with success
error	String	Error content
code	Integer	Error code, return 200 when success

**Request Example:**

Add a local Firmware Task:

```
{
  "fuotaTask": {
    "name": "Test",
    "startedAt": "2025-04-15T00:50:00.000Z",
    "description": "test",
    "deveui": [
      "24e1247151234568",
      "24e124715123456b"
    ],
    "firmwareInfo": {
      "fileContent": "aaaaaaaaaaaaaaaaaaaaaaaaaaaa",
      "firmwareName": "WT101.0000.0100.0101.bin"
    },
    "isOfficialFirmware": false,
    "fragmentInfo": {
      "fragmentSize": 88,
      "fragmentInterval": 5000,
      "fragmentRedundancy": 30
    },
    "tmpMcInfo": {
      "dr": 3,
      "frequency": 505300000
    }
  }
}
```

Add an Official Firmware Task:

```
{
  "fuotaTask": {
    "name": "test",
    "startedAt": "2025-04-15T00:26:00.000Z",
    "description": "test",
    "deveui": [
      "24e1247151234569",
      "24e124715123456a"
    ],
    "officialFirmwareInfo": {
```

```
    "id": "1888828394920153089",
    "model": "CT303",
    "name": "CT303 v1.4",
    "firmwareVersion": "v1.4",
    "hardwareVersion": "v1.0",
    "supportMinHardwareVersion": "v1.0",
    "supportMinFirmwareVersion": "v1.3",
    "url":
"https://milesight-msc-cn-test-new.s3.cn-north-1.amazonaws.com.cn/firmware/system/280
137a5-27d1-4c45-981e-96cf932ed5b1-1739166276.bin",
    "md5": "8f5c6beaf3ac56d829b178203fa9d610",
    "crc32": "d67cf42e",
    "size": "182581",
    "frequencyBands": [
        "ALL"
    ],
    "firmwareNotesCn": "",
    "firmwareNotesEn": ""
},
"isOfficialFirmware": true,
"fragmentInfo": {
    "fragmentSize": 88,
    "fragmentInterval": 5000,
    "fragmentRedundancy": 30
},
"tmpMcInfo": {
    "dr": 3,
    "frequency": 505300000
}
}
```

### Response Example:

```
{
  "id": 1
}
```

## 12.2 Get FUOTA Task List

**Request Method:** GET

**Request Address:**

```
https://{gatewayIP}:8080/api/fuota/task
```

## Request Parameters

### Header

Parameter	Required	Description
Content-Type	Yes	application/json

### Params

Parameter	Type	Required	Description
limit	Integer	No	Max number of every page.
offset	Integer	No	Offset in the result-set (for pagination).
search	String	No	keyword

## Response Parameters

Parameter	Type	Description
total	Integer	Total number of tasks
tasks	Array[]	Every task info
id	Integer	Task ID
name	String	Task name
description	String	Task description
status	Integer	0: pending, 1: waiting, 2: executing, 3: upgrade success, 4: upgrade failed
startedAt	String	The time to start this task
stoppedAt	String	The time to stop this task
createdAt	String	The time to create this task
updatedAt	String	The time to complete this task
firmwareName	String	Upgraded firmware name
doneDevices	Integer	The device number to complete the upgrade
totalDevices	Integer	Total device number to upgrade the firmware
deveui	String[]	Device EUI list
firmwareInfo	Object	Firmware information
firmwareName	String	Firmware name
description	String	Firmware description
fileContent	String	Firmware content (base64)
productModel	String	Product model
firmwareVersion	String	Firmware version
supportHardwareVersion	String	Supported hardware version
supportFirmwareVersion	String	Supported firmware version
officialFirmwareUrl	String	Official firmware URL
fragmentInfo	Object	Firmware fragment info
fragmentSize	Integer	Fragment size, unit: byte
fragmentInterval	Integer	Fragment interval, unit: ms
fragmentRedundancy	Integer	Fragment redundancy, unit: %
tmpMcInfo	Object	Temporary multicast info

sessionTimeDelay	Integer	
sessionTimeOut	Integer	
status	Integer	
groupType	Integer	0: class C
dr	Integer	Datarate
frequency	Integer	Downlink frequency, unit: Hz
isOfficialFirmware	Bool	Whether is official firmware
officialFirmwareInfo	Object	Official firmware info
id	String	Firmware ID
model	String	Product model
name	String	Firmware name
firmwareVersion	String	Firmware version
hardwareVersion	String	Hardware version
supportMinHardwareVersion	String	Supported min hardware version
supportMinFirmwareVersion	String	Supported min firmware version
url	String	Download url
md5	String	
crc32	String	
size	String	Firmware size
frequencyBands	String[]	Supported frequency
firmwareNotesCn	String	English note
firmwareNotesEn	String	Chinese note

**Request Example:**

```
https://192.168.23.150:8080/api/fuota/task?limit=10&offset=0
```

**Response Example:**

```
{
  "total": 1,
  "tasks": [
    {
      "id": 184,
      "name": "test",
      "description": "test",
      "status": 0,
      "startedAt": "2025-04-15T01:13:00+01:00",
      "stoppedAt": "",
      "createdAt": "2025-04-15T07:17:40+01:00",
      "updatedAt": "2025-04-15T07:17:40+01:00",
    }
  ]
}
```

```
"firmwareName": "CT305 v1.4",
"doneDevices": 0,
"totalDevices": 2,
"device": [
  "24e1247151234568",
  "24e1247151234569"
],
"firmwareInfo": {
  "firmwareName": "CT305 v1.4",
  "description": "",
  "fileContent": "",
  "productModel": "CT305",
  "firmwareVersion": "v1.4",
  "supportHardwareVersion": "v1.0",
  "supportFirmwareVersion": "v1.3",
  "officialFirmwareUrl": ""
},
"fragmentInfo": {
  "fragmentSize": 88,
  "fragmentInterval": 5000,
  "fragmentRedundancy": 30
},
"tmpMcInfo": {
  "sessionTimeDelay": 0,
  "sessionTimeOut": 0,
  "status": 0,
  "groupType": 0,
  "dr": 3,
  "frequency": 505300000
},
"officialFirmwareInfo": {
  "id": "",
  "model": "CT305",
  "name": "CT305 v1.4",
  "firmwareVersion": "v1.4",
  "hardwareVersion": "v1.0",
  "supportMinHardwareVersion": "v1.0",
  "supportMinFirmwareVersion": "v1.3",
  "url":
"https://milesight-msc-cn.com.cn/firmware/system/280137a5-27d1-4c45-981e-96cf932ed5b1-1739166276.bin",
  "md5": "",
  "crc32": "d67cf42e",
  "size": "0",
```

```

        "frequencyBands": [],
        "firmwareNotesCn": "",
        "firmwareNotesEn": ""
    },
    "isOfficialFirmware": true
}
]
}

```

## 12.3 Update FUOTA Task

**Request Method:** PUT

**Request Address:**

<https://{gatewayIP}:8080/api/fuota/task>

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

Body

Parameter	Type	Required	Description
fuotaTask	Object[]	Yes	FUOTA task information
id	Integer	Yes	Task ID, this can be searched by API 12.2
name	String	Yes	Task name, length limit: 127
startedAt	String	Yes	The time to start this task
description	String	No	Task description
deveui	String[]	Yes	Device EUI list
firmwareInfo	Object[]	Yes	Firmware information
firmwareName	String	Yes	Firmware name
description	String	No	
fileContent	String	Yes	Firmware content (base64)
officialFirmwareUrl	String	No	Official firmware URL
fragmentInfo	Object[]	Yes	Firmware fragment info
fragmentSize	Integer	yes	Fragment size, range: 1-250, unit: byte
fragmentInterval	Integer	yes	Fragment interval, range: 1000-10000, unit: ms
fragmentRedundancy	Integer	yes	Fragment redundancy, range: 0-100, unit:%
tmpMcInfo	Object[]	Yes	Temporary multicast info

groupType	Integer	yes	0: class C
dr	Integer	yes	Datarate, refer to <a href="#">Default RX2/PingSlot Settings</a>
frequency	Integer	yes	Downlink frequency, unit: Hz refer to <a href="#">Default RX2/PingSlot Settings</a>
isOfficialFirmware	bool	No	Whether is official firmware
officialFirmwarInfo	Object[]	No	Get these info from API 12.10
id	String	Yes	Firmware ID
model	String	Yes	Product model
name	String	Yes	Firmware name
firmwareVersion	String	Yes	Firmware version
hardwareVersion	String	Yes	Hardware version
supportMinHardwareVersion	String	Yes	Supported min hardware version
supportMinFirmwareVersion	String	Yes	Supported min firmware version
url	String	Yes	Download url
md5	String	Yes	
crc32	String	Yes	
size	String	Yes	Firmware size
frequencyBands	String[]	Yes	Supported frequency
firmwareNotesCn	String	Yes	English note
firmwareNotesEn	String	Yes	Chinese note

### Response Parameters

Parameter	Type	Description
code	Integer	Error code, return 200 when success
error	String	Error content

### Request Example:

Update a local Firmware Task:

```
{
  "fuotaTask": {
    "id": 187,
    "name": "Test",
    "startedAt": "2025-04-15T00:50:00.000Z",
    "description": "test",
    "deveui": [
      "24e1247151234568",
      "24e124715123456b"
    ],
    "firmwareInfo": {
      "fileContent": "aaaaaaaaaaaaaaaaaaaaaaaaaaaaa",

```



```

        "firmwareName": "WT101.0000.0100.0101.bin"
    },
    "isOfficialFirmware": false,
    "fragmentInfo": {
        "fragmentSize": 88,
        "fragmentInterval": 5000,
        "fragmentRedundancy": 30
    },
    "tmpMcInfo": {
        "dr": 3,
        "frequency": 505300000
    }
}

```

#### Update an Official Firmware Task:

```

{
    "fuotaTask": {
        "id": 187,
        "name": "test",
        "startedAt": "2025-04-15T00:26:00.000Z",
        "description": "test",
        "deveui": [
            "24e1247151234569",
            "24e124715123456a"
        ],
        "officialFirmwareInfo": {
            "id": "1888828394920153089",
            "model": "CT303",
            "name": "CT303 v1.4",
            "firmwareVersion": "v1.4",
            "hardwareVersion": "v1.0",
            "supportMinHardwareVersion": "v1.0",
            "supportMinFirmwareVersion": "v1.3",
            "url":
                "https://milesight-msc-cn-test-new.s3.cn-north-1.amazonaws.com.cn/firmware/system/280
                137a5-27d1-4c45-981e-96cf932ed5b1-1739166276.bin",
            "md5": "8f5c6beaf3ac56d829b178203fa9d610",
            "crc32": "d67cf42e",
            "size": "182581",
            "frequencyBands": [
                "ALL"
            ],
            "firmwareNotesCn": "",
            "firmwareNotesEn": ""
        }
    }
}

```

```
    },
    "isOfficialFirmware": true,
    "fragmentInfo": {
      "fragmentSize": 88,
      "fragmentInterval": 5000,
      "fragmentRedundancy": 30
    },
    "tmpMcInfo": {
      "dr": 3,
      "frequency": 505300000
    }
  }
}
```

**Response Example:**

```
{
  "code": 200,
  "error": ""
}
```

## 12.4 Delete FUOTA Task

**Request Method:** POST**Request Address:**

```
https://{gatewayIP}:8080/api/fuota/task/delete
```

**Request Parameters**

Header

Parameter	Required	Description
Content-Type	Yes	application/json

Body

Parameter	Type	Required	Description
ids	Integer[]	Yes	Task IDs to delete

**Response Parameters**

Parameter	Type	Description
code	Integer	Error code, return 200 when success
error	String	Error content

**Request Example:**

```
{
  "ids": [1, 2, 3]
}
```

**Response Example:**

```
{
  "code": 200,
  "error": ""
}
```

## 12.5 Retry FUOTA Task

**Request Method:** POST**Request Address:**

```
https://{gatewayIP}:8080/api/fuota/task/retry
```

**Request Parameters**

Header

Parameter	Required	Description
Content-Type	Yes	application/json

Body

Parameter	Type	Required	Description
ids	Integer[]	Yes	Task IDs to retry

**Response Parameters**

Parameter	Type	Description
code	Integer	Error code, return 200 when success
error	String	Error content

**Request Example:**

```
{
  "ids": [1, 2, 3]
}
```

**Response Example:**

```
{
  "code": 200,
  "error": ""
}
```

## 12.6 Get Payload Size of Datarate

**Request Method:** GET

**Request Address:**

`https://{gatewayIP}:8080/api/fuota/task/payloadsize`

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

### Response Parameters

Parameter	Type	Description
maxPayloadSizeMap	Array	
dr	UInteger	datarate
maxPayloadSize	UInteger	Max payload size

**Request Example:**

`https://192.168.23.150:8080/api/fuota/task/payloadsize`

**Response Example:**

```
{
  "maxPayloadSizeMap": [
    {
      "dr": 0,
      "maxPayloadSize": 46
    },
    {
      "dr": 1,
      "maxPayloadSize": 46
    },
    {
      "dr": 2,
      "maxPayloadSize": 46
    },
    {
      "dr": 3,
      "maxPayloadSize": 110
    },
    {
      "dr": 4,
      "maxPayloadSize": 237
    }
  ]
}
```

```
}
{
  "dr": 5,
  "maxPayloadSize": 237
}
]
```

## 12.7 Get Device List from FUOTA Task

**Request Method:** GET

**Request Address:**

https://{gatewayIP}:8080/api/fuota/devices

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

Params

Parameter	Type	Required	Description
taskId	Integer	Yes	The task ID to search devices
limit	Integer	No	Max number of every page.
offset	Integer	No	Offset in the result-set (for pagination).
search	String	No	keyword

### Response Parameters

Parameter	Type	Description
total	Integer	Total number of devices in this task
devices	Array[]	Every device info in this task
devEui	String	Device EUI
name	String	Device name
profileId	String	Profile ID
profileName	String	Profile name
hardwareVersion	String	Supported hardware version
firmwareVersion	String	Supported firmware version
fragmentSessionStatus	Integer	See <a href="#">Fragment Session Status</a>
multicastSessionStatus	Integer	See <a href="#">Multicast Session Status</a>
status	Integer	Device status
productModel	String	Product model of this device
taskId	Integer	Task ID which the device belongs to

**Request Example:**

```
https://192.168.23.150:8080/api/fuota/devices?taskId=1&limit=10&offset=0
```

**Response Example:**

```
{
  "total": 2,
  "devices": [
    {
      "devEui": "24e1247151234568",
      "name": "WT201-1",
      "profileId": "a98f2ea4-b4d8-42a4-bf0c-5dbb6e4617ff",
      "profileName": "ClassC-OTAA",
      "hardwareVersion": "",
      "firmwareVersion": "",
      "fragmentSessionStatus": 0,
      "multicastSessionStatus": 0,
      "status": 0,
      "productModel": "WT201",
      "taskId": 183
    },
    {
      "devEui": "24e1247151234569",
      "name": "WT201-2",
      "profileId": "a98f2ea4-b4d8-42a4-bf0c-5dbb6e4617ff",
      "profileName": "ClassC-OTAA",
      "hardwareVersion": "",
      "firmwareVersion": "",
      "fragmentSessionStatus": 0,
      "multicastSessionStatus": 0,
      "status": 0,
      "productModel": "WT201",
      "taskId": 184
    }
  ]
}
```

## 12.8 Get Device List from Firmware Info

**Request Method:** POST**Request Address:**

```
https://{gatewayIP}:8080/api/fuota/devices/firmware
```

**Request Parameters**

## Header

Parameter	Required	Description
Content-Type	Yes	application/json

## Params

Parameter	Type	Required	Description
firmwareName	String	Yes	Firmware name
fileContent	String	Yes	Firmware file content (base64)
limit	Integer	No	Max number of every page.
offset	Integer	No	Offset in the result-set (for pagination).
officialFirmwareInfo	Object	No	Official firmware info
taskId	Integer	No	Task ID
search	String	No	keyword

## Response Parameters

Parameter	Type	Description
total	Integer	Total number of devices in this task
devices	Array[]	Every device info in this task
devEui	String	Device EUI
name	String	Device name
profileId	String	Profile ID
profileName	String	Profile name
hardwareVersion	String	Supported hardware version
firmwareVersion	String	Supported firmware version
fragmentSessionStatus	Integer	See <a href="#">Fragment Session Status</a>
multicastSessionStatus	Integer	See <a href="#">Multicast Session Status</a>
status	Integer	Device status
productModel	String	Product model of this device
taskId	Integer	Task ID which the device belongs to

## Request Example:

Get device list from a local Firmware info:

```
{
  "firmwareName": "test_firmware",
  "fileContent":
"d2e1z6LXgt3N7UqWlIpKMNlqBfH4VQgcL74+5cu0AKfdp5r9BHrWBCMyOSpl0Ekn3aea/QR6
1gQjMjkq.....",
  "limit": 10,
  "offset": 0,
  "taskId": 0,
  "search": ""
}
```

Get device list from an official Firmware info:

```
{
  "limit": 10,
  "offset": 0,
  "taskId": 0,
  "search": "",
  "officialFirmwareInfo": {
    "0": true,
    "id": "1888828394920153089",
    "model": "CT303",
    "name": "CT303 v1.4",
    "firmwareVersion": "v1.4",
    "hardwareVersion": "v1.0",
    "supportMinHardwareVersion": "v1.0",
    "supportMinFirmwareVersion": "v1.3",
    "url":
      "https://milesight-msc-cn-test-new.s3.cn-north-1.amazonaws.com.cn/firmware/system/280
      137a5-27d1-4c45-981e-96cf932ed5b1-1739166276.bin",
    "md5": "8f5c6beaf3ac56d829b178203fa9d610",
    "crc32": "d67cf42e",
    "size": "182581",
    "frequencyBands": [
      "ALL"
    ],
    "firmwareNotesCn": "",
    "firmwareNotesEn": ""
  }
}
```

### Response Example:

```
{
  "total": 1,
  "devices": [
    {
      "devEui": "24E124136A456465",
      "name": "test_device",
      "profileId": "profile1",
      "profileName": "ClassA",
      "hardwareVersion": "1.0.0",
      "firmwareVersion": "1.0.0",
      "fragmentSessionStatus": 0,
      "multicastSessionStatus": 0,
      "status": 0,
    }
  ]
}
```



```
{
  "productModel": "model1",
  "taskId": 1
}
```

## 12.9 Get Official Firmware Model List

**Request Method:** GET

**Request Address:**

`https://{gatewayIP}:8080/api/fuota/official/models`

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

Params

Parameter	Type	Required	Description
dateCenter	string	Yes	eu, us, cn, sg

### Response Parameters

Parameter	Type	Description
models	String[]	Model list

### Request Example:

`https://192.168.23.150:8080/api/fuota/official/models?dateCenter=sg`

### Response Example:

```
{
  "models": ["WT201", "WT101"]
}
```

## 12.10 Get Official Firmware Info

**Request Method:** GET

**Request Address:**

`https://{gatewayIP}:8080/api/fuota/official/firmware`

### Request Parameters

## Header

Parameter	Required	Description
Content-Type	Yes	application/json

## Params

Parameter	Type	Required	Description
model	string	Yes	Product model
limit	Integer	No	Max number of every page.
offset	Integer	No	Offset in the result-set (for pagination).
search	String	No	keyword

## Response Parameters

Parameter	Type	Description
total	Integer	Total firmware amount
data	Array	Firmware list
id	String	Firmware ID
model	String	Product model
name	String	Firmware name
firmwareVersion	String	Firmware version
hardwareVersion	String	Hardware version
supportMinHardwareVersion	String	Supported min hardware version
supportMinFirmwareVersion	String	Supported min firmware version
url	String	Download url
md5	String	
crc32	String	
size	Integer	Firmware size
frequencyBands	String[]	Supported frequency
firmwareNotesCn	String	English note
firmwareNotesEn	String	Chinese note

## Request Example:

```
https://192.168.23.150:8080/api/fuota/official/firmware?model=model1&limit=10&offset=0
```

## Response Example:

```
{
  "total": 1,
  "data": [
    {
      "id": "1",
      "model": "model1",
      "name": "test_firmware",
      "firmwareVersion": "1.0.0",
      "hardwareVersion": "1.0.0",
      "supportMinHardwareVersion": "1.0.0",
```

```
"supportMinFirmwareVersion": "1.0.0",  
"url": "http://example.com/firmware",  
"md5": "md5_value",  
"crc32": "crc32_value",  
"size": 1024,  
"frequencyBands": ["CN470"],  
"firmwareNotesCn": "中文说明",  
"firmwareNotesEn": "English notes"  
}  
]  
}
```

## 12.11 Return Code List

### Error Code

Code	Description
10310001	Not able to delete devices since it belongs to a FUOTA task
10310002	Not able to delete devices since they belong to FUOTA tasks
10310003	Task name is blank or too long
10310004	Task start time format invalid
10310005	Miss fragment info
10310006	Miss tmp_mc_info
10310007	Miss firmware info
10310008	Miss official firmware info
10310009	Miss custom firmware info
10310010	Fragment size invalid
10310011	Fragment interval invalid
10310012	Redundancy percent invalid
10310013	Over the maximum task amount
10310014	Not able to execute due to current task status
10310015	Failed to get official firmware information from server

### Multicast Session Status

Code	Description
0	Multicast session has not started yet
1	Trying to set up temp. Multicast group
2	Temp. Multicast group is confirmed
3	Multicast session is ready
4	Multicast session is processing
5	Multicast session completes
6	Multicast group set up timeout: the device does not respond to

	the setting command from gateway
7	Multicast ID does not match
8	Multicast session request timeout: the device does not respond to the gateway
9	Multicast ID is not defined
10	Frequency error: The device does not support this frequency
11	Datarate error: The device does not support this datarate
12	Multicast ID does not match
13	Unknown error

## Fragment Session Status

Code	Description
0	Fragment has not started yet
1	Trying to set up fragment session
2	Fragment session is ready
3	Fragment session is processing
4	Fragment session set up timeout: the device does not respond to the setting command from gateway
5	Fragment descriptor mismatch
6	Not supporting fragment session search
7	Low device memory
8	The device does not support fragment coding mode
9	Lack of fragments
10	Upgrade failed
11	Fragment session error
12	Fragment session complete
13	Upgrade complete
14	Upgrade timeout
15	Status report timeout: the device does not report status within the timeout
16	Unknown error

# 13. BACnet Server

## 13.1 Get BACnet Server Configuration

**Request Method:** GET

**Request Address:**

<https://{gatewayIP}:8080/api/bacnet/server>

## Request Parameters

### Header

Parameter	Required	Description
Content-Type	Yes	application/json

## Response Parameters

Parameter	Type	Description
id	Integer	
enable	Integer	Whether to enable BACnet server, 0: Disable, 1: Enable
udp_port	Integer	UDP Port
device_id	Integer	BACnet device id
device_name	String	BACnet device name
bbmd_enable	Integer	Whether to enable BBMD, 0: Disable, 1: Enable
bbmd_ip	String	BBMD IP address
bbmd_port	Integer	BBMD Port
bbmd_register_interval	Integer	BBMD Time TO Live (s)
global_object_enable	Boolean	Whether to enable global object
global_objects	Array	
object_type	String	
auto_add_objects	Boolean	Whether to add BACnet objects when adding devices

## Response Example

```
{
  "data": {
    "id": 1,
    "enable": 1,
    "udp_port": 47808,
    "device_id": 1234,
    "device_name": "UG56-6041D4818086",
    "bbmd_enable": 0,
    "bbmd_ip": "",
    "bbmd_port": 47808,
    "bbmd_register_interval": 60000,
    "global_object_enable": false,
    "global_objects": ["status"],
    "auto_add_objects": false
  }
}
```

## 13.2 Set BACnet Server Configuration

**Request Method:** POST

**Request Address:**

https://{gatewayIP}:8080/api/bacnet/server

**Request Parameters**

## Header

Parameter	Required	Description
Content-Type	Yes	application/json

## Body

Parameter	Type	Required	Description
id	Integer	Yes	1
enable	Integer	Yes	Whether to enable BACnet server, 0: Disable, 1: Enable
udp_port	Integer	Yes	UDP Port
device_id	Integer	Yes	BACnet device id
device_name	String	Yes	BACnet device name
bbmd_enable	Integer	Yes	Whether to enable BBMD, 0: Disable, 1: Enable
bbmd_ip	String	Yes	BBMD IP address
bbmd_port	Integer	Yes	BBMD Port
bbmd_register_interval	Integer	Yes	BBMD Time TO Live (s)
global_object_enable	Boolean	Yes	Whether to enable global object
global_objects	Array	Yes	
object_type	String	Yes	
auto_add_objects	Boolean	Yes	Whether to add BACnet objects when adding devices

**Response Parameters**

Parameter	Type	Description
Error	string	
Code	number	

**Request Example**

```
{
  "data": {
    "id": 1,
    "enable": 1,
    "udp_port": 47808,
    "device_id": 1234,
    "device_name": "UG56-6041D4818086",
    "bbmd_enable": 0,
    "bbmd_ip": "",
    "bbmd_port": 47808,

```

```
{
  "bbmd_register_interval": 60000,
  "global_object_enable": true,
  "global_objects": [
    "status"
  ],
  "auto_add_objects": false
}
```

### Response Example

```
{
  "error": "",
  "code": 0
}
```

## 13.3 Get Available BACnet Object List

**Request Method:** POST

**Request Address:**

<https://{gatewayIP}:8080/api/bacnet/getAll>

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

Body

Parameter	Type	Required	Description
search	String	No	keyword
limit	Integer	Yes	Max number of applications to return in the result-test.
offset	Integer	No	Offset in the result-set (for pagination).

### Response Parameters

Parameter	Type	Description
total	Integer	The total number of the BACnet object which can be selected.
noc_data	Array	Notification class data
data	Array	
id	String	Device ID
dev_eui	String	Device EUI
name	String	Device name
type	String	Device type

ids	Array	Already added BACnet object IDs, dev EUI+Payload Codec object ID
objects	Array	BACnet object info
id	String	dev EUI+Payload Codec object ID
dev_eui	String	Device EUI
name	String	BACnet object name
payload_codec_object_id	String	
lora_name	Array	LoRa object name from payload codec
lora_unit_type_id	Integer	LoRa object unit ID from payload codec
type	String	BACnet object type
description	String	BACnet object description
unit_type_id	Integer	Bacnet unit ID
cov_enable	Integer	Analog type only, 0: Disable, 1: Enable
cov_increment	String	Analog type only
polarity	Integer	Binary type only, 0: Disable or Normal, 1: Reverse
relinquish_default	String	Analog output only
inactive_text	String	Output of inactive binary input/output status
active_text	String	Output of active binary input/output status
notification_class	Integer	Related notification class, -1=None
high_limit	String	High limit to report the event
low_limit	String	Low limit to report the event
deadband	String	Analog type only, the delay value to change back to normal status
limit_enable	Integer	0: Disable, 1: Enable
event_enable	Integer	0: Disable, 1: Enable
notify_type	Integer	0: Alarms, 1: Events
alarm_value	Integer	
alarm_value_array	Array	MultiState input or MultiState value only, the value to define offnormal or normal event
fault_value_array	Array	MultiState input or MultiState value only, the value to report To Fault event
feedback_value	Integer	MultiState output or binary output only, the value to define offnormal or normal event
time_delay	Integer	The delay time to report the event
unit	String	Unit name
instance_id	Integer	BACnet object instance ID
number_of_states	Integer	multiState type only, the number of the states
state_text	Array	multiState type only, names of states
values	Array	
name	String	The meaning of the value
value	Integer	
reference	Array	Reference list



count	Integer	Related unit count
units	Array	Related unit list
type_alias	String	BACnet object type alias
value	Integer	Current value
updated_times	Integer	Update times
updated_at	String	Last update time

### Request Example

```
{
  "search": "",
  "order": "asc",
  "offset": 0,
  "limit": 10
}
```

### Response Example

```
{
  "total": 981,
  "noc_data": [],
  "data": [
    {
      "id": "24e1247151234511",
      "dev_eui": "24e1247151234511",
      "name": "WT201",
      "type": "",
      "ids": [
        "24e1247151234511-0",
        "24e1247151234511-1510",
        "24e1247151234511-1620",
        "24e1247151234511-1573",
        ...
      ],
      "objects": [
        {
          "id": "24e1247151234511-1554",
          "dev_eui": "24e1247151234511",
          "name": "WT201.card_config.action_type",
          "payload_codec_object_id": "1554",
          "lora_name": "card_config.action_type",
          "lora_unit_type_id": 95,
          "type": "multistate_value_object",
          "description": "",
          "unit_type_id": 95,
          "cov_enable": 0,

```

```
"cov_increment": "1.0",
"polarity": 0,
"relinquish_default": "0",
"inactive_text": "",
"active_text": "",
"notification_class": -1,
"high_limit": "0",
"low_limit": "0",
"deadband": "0",
"limit_enable": 0,
"event_enable": 0,
"notify_type": 0,
"alarm_value": 0,
"alarm_value_array": [],
"fault_value_array": [],
"feedback_value": 0,
"time_delay": 0,
"unit": "no-units",
"instance_id": 0,
"number_of_states": 2,
"state_text": [
    "power",
    "plan"
],
"values": [
    {
        "name": "power",
        "value": 0
    },
    {
        "name": "plan",
        "value": 1
    }
],
"reference": [
    "card_config.enable",
    "card_config.in_plan_type",
    "card_config.out_plan_type",
    "card_config.invert"
],
"count": 0,
"units": [],
"type_alias": "MultiState-Value",
"value": ""
```

```

        "updated_times": 0,
        "updated_at": ""
    },
    ...
]
}
}
}

```

## 13.4 Get Existed BACnet Object List

**Request Method:** POST

**Request Address:**

<https://{gatewayIP}:8080/api/bacnet/get>

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

Body

Parameter	Type	Required	Description
search	String	No	keyword
order	String	No	asc or desc
limit	Integer	Yes	Max number of applications to return in the result-test.
offset	Integer	No	Offset in the result-set (for pagination).

### Response Parameters

Parameter	Type	Description
total	Integer	The total number of the BACnet object which can be selected.
noc_data	Array	Notification class data
id	String	
object_name	String	BACnet object name
to_offnormal_priority	Integer	The priority number used by recipients to sort the event notifications. Range: 0-255 (0 being most important, 255 least important)
to_normal_priority	Integer	
to_fault_priority	Integer	
description	String	BACnet object description
ack_required	Integer	0: disable, 1: to offnormal, 2: to fault, 4: to normal, 7: enable all
instance_id	Integer	BACnet object instance ID
type	String	Notification class object

recipient_list	Array	The list to receive event notification
device_identifier	Integer	Target recipient device ID
valid_day	Integer	Valid days to send notification
from_time	String	Valid time to send notification
to_time	String	
process_identifier	Integer	The identifier to indicate what process the alarm is intended for
transitions	Integer	0: disable, 1: to offnormal, 2: to fault, 4: to normal, 7: enable all
confirmed_notify	Integer	0: Unconfirmed, 1: Confirmed
data	Array	
id	String	Device ID
dev_eui	String	Device EUI
name	String	Device name
type	String	Device type
ids	Array	Already added BACnet object IDs, dev EUI+Payload Codec object ID
objects	Array	BACnet object info
id	String	dev EUI+Payload Codec object ID
dev_eui	String	Device EUI
name	String	BACnet object name
payload_codec_object_id	String	
lora_name	Array	LoRa object name from payload codec
lora_unit_type_id	Integer	LoRa object unit ID from payload codec
type	String	BACnet object type
description	String	BACnet object description
unit_type_id	Integer	Bacnet unit ID
cov_enable	Integer	Analog type only, 0: Disable, 1: Enable
cov_increment	String	Analog type only
polarity	Integer	Binary type only, 0: Disable or Normal, 1: Reverse
relinquish_default	String	Analog output only
inactive_text	String	Output of inactive binary input/output status
active_text	String	Output of active binary input/output status
notification_class	Integer	Related notification class, -1=None
high_limit	String	High limit to report the event
low_limit	String	Low limit to report the event
deadband	String	Analog type only, the delay value to change back to normal status
limit_enable	Integer	0: Disable, 1: Enable
event_enable	Integer	0: Disable, 1: Enable
notify_type	Integer	0: Alarms, 1: Events
alarm_value	Integer	

alarm_value_array	Array	MultiState input or MultiState value only, the value to define offnormal or normal event
fault_value_array	Array	MultiState input or MultiState value only, the value to report To Fault event
feedback_value	Integer	MultiState output or binary output only, the value to define offnormal or normal event
time_delay	Integer	The delay time to report the event
unit	String	Unit name
instance_id	Integer	BACnet object instance ID
number_of_states	Integer	multiState type only, the number of the states
state_text	Array	multiState type only, names of states
values	Array	
name	String	The meaning of the value
value	Integer	
reference	Array	Reference list
count	Integer	Related unit count
units	Array	Related unit list
type_alias	String	BACnet object type alias
value	Integer	Current value
updated_times	Integer	Update times
updated_at	String	Last update time

### Request Example

```
{
  "search": "",
  "order": "asc",
  "offset": 0,
  "limit": 10
}
```

### Response Example

```
{
  "total": 980,
  "noc_data": [
    {
      "id": "1",
      "object_name": "test",
      "to_offnormal_priority": 100,
      "to_normal_priority": 125,
      "description": "test",
      "ack_required": 7,
      "instance_id": 0,
      "to_fault_priority": 125,
      "type": "notification-class-object",

```

```
"recipient_list": [
  {
    "device_identifier": 12314,
    "valid_day": 127,
    "from_time": "0:0",
    "to_time": "23:59",
    "process_identifier": 1234,
    "transitions": 7,
    "confirmed_notify": 1
  }
],
"data": [
  {
    "id": "24e1247151234568",
    "dev_eui": "24e1247151234568",
    "name": "WT201-1",
    "type": "device",
    "ids": [],
    "objects": [
      {
        "id": "22240",
        "dev_eui": "24e1247151234568",
        "name": "WT201-1.fan_mode",
        "payload_codec_object_id": "1516",
        "lora_name": "fan_mode",
        "lora_unit_type_id": 95,
        "type": "multistate_value_object",
        "description": "",
        "unit_type_id": 95,
        "cov_enable": 0,
        "cov_increment": "1.0",
        "polarity": 0,
        "relinquish_default": "0",
        "inactive_text": "",
        "active_text": "",
        "notification_class": -1,
        "high_limit": "0",
        "low_limit": "0",
        "deadband": "0",
        "limit_enable": 0,
        "event_enable": 0,
        "notify_type": 0,
```

```

        "alarm_value": 0,
        "alarm_value_array": [],
        "fault_value_array": [],
        "feedback_value": 0,
        "time_delay": 0,
        "unit": "no-units",
        "instance_id": 8,
        "number_of_states": 4,
        "state_text": [
            "auto",
            "on",
            "circulate",
            "disable"
        ],
        "values": [],
        "reference": [],
        "count": 0,
        "units": [],
        "type_alias": "MultiState-Value",
        "value": "-",
        "updated_times": 0,
        "updated_at": "-"
    }
}
]
}
}

```

## 13.5 Add BACnet Object

**Note: Please enquire the available BACnet object list via API 13.3 before adding.**

**Request Method:** POST

**Request Address:**

<https://{gatewayIP}:8080/api/bacnet/add>

### Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

Body

Parameter	Type	Required	Description
data	Array	Yes	

id	String	No	Device ID
dev_eui	String	Yes	Device EUI
name	String	Yes	Device name
objects	Array	Yes	BACnet object info
id	String	Yes	dev EUI+Payload Codec object ID
dev_eui	String	Yes	Device EUI
name	String	Yes	BACnet object name
payload_codec_object_id	String	Yes	
lora_name	Array	Yes	LoRa object name from payload codec
lora_unit_type_id	Integer	Yes	LoRa object unit ID from payload codec
type	String	Yes	BACnet object type
description	String	Yes	BACnet object description
unit_type_id	Integer	Yes	Bacnet unit ID
cov_enable	Integer	Yes	Analog type only, 0: Disable, 1: Enable
cov_increment	String	Yes	Analog type only
polarity	Integer	Yes	Binary type only, 0: Disable or Normal, 1: Reverse
relinquish_default	String	Yes	Analog output only
inactive_text	String	Yes	Output of inactive binary input/output status
active_text	String	Yes	Output of active binary input/output status
notification_class	Integer	Yes	Related notification class, -1=None
high_limit	String	Yes	High limit to report the event
low_limit	String	Yes	Low limit to report the event
deadband	String	Yes	Analog type only, the delay value to change back to normal status
limit_enable	Integer	Yes	0: Disable, 1: Enable
event_enable	Integer	Yes	0: Disable, 1: Enable 0: disable, 1: to normal, 2: to fault, 4: to offnormal, 7: enable all
notify_type	Integer	Yes	0: Alarms, 1: Events
alarm_value	Integer	Yes	
alarm_value_array	Array	Yes	MultiState input or MultiState value only, the value to define offnormal or normal event
fault_value_array	Array	Yes	MultiState input or MultiState value only, the value to report To Fault event
feedback_value	Integer	Yes	MultiState output or binary output only, the value to define offnormal or normal event
time_delay	Integer	Yes	The delay time to report the event
unit	String	Yes	Unit name



instance_id	Integer	Yes	BACnet object instance ID
number_of_states	Integer	Yes	multiState type only, the number of the states
state_text	Array	Yes	multiState type only, names of states
values	Array	Yes	
name	String	Yes	The meaning of the value
value	Integer	Yes	
reference	Array	Yes	Reference list
count	Integer	Yes	Related unit count
units	Array	Yes	Related unit list
type_alias	String	Yes	BACnet object type alias
value	Integer	Yes	Current value
updated_times	Integer	Yes	Update times
updated_at	String	Yes	Last update time

### Response Parameters

Parameter	Type	Description
error	String	Error content
code	Integer	Error code

### Request Example

```
{
  "data": [
    {
      "id": "24e124710e338704-32577",
      "dev_eui": "24e124710e338704",
      "name": "am319.co2",
      "objects": [
        {
          "id": "24e124710e338704-32577",
          "dev_eui": "24e124710e338704",
          "name": "am319.co2",
          "payload_codec_object_id": "32577",
          "lora_name": "co2",
          "lora_unit_type_id": 96,
          "type": "analog_input_object",
          "description": "",
          "unit_type_id": 96,
          "cov_enable": 0,
          "cov_increment": "1.0",
          "polarity": 0,
          "relinquish_default": "0",
          "inactive_text": "",
          "active_text": ""
        }
      ]
    }
  ]
}
```

```
        "notification_class": -1,
        "high_limit": "0",
        "low_limit": "0",
        "deadband": "0",
        "limit_enable": 0,
        "event_enable": 0,
        "notify_type": 0,
        "alarm_value": 0,
        "alarm_value_array": [],
        "fault_value_array": [],
        "feedback_value": 0,
        "time_delay": 0,
        "unit": "ppm",
        "instance_id": 1,
        "number_of_states": 0,
        "state_text": [],
        "values": [],
        "reference": [],
        "count": 2,
        "units": [
            {
                "unit": "ppm",
                "unit_type_id": 0
            },
            {
                "unit": "ppb",
                "unit_type_id": 0
            }
        ],
        "type_alias": "Analog-Input",
        "value": "",
        "updated_times": 0,
        "updated_at": ""
    }
}
]
```

### Response Example

```
{
  "error": "",
  "code": 0
}
```

# Appendix

## Return Code List

Code	Description
200	Success
2	Unknown error
3	Parameter exceptions
5	Data not found
6	Already exists
9	Failed precondition
20	Parse json failed
21	Type or url not supported
22	Method not supported
23	Internal server error
24	Request timeout
25	Gateway EUI not matched

## Default RX2/Ping Slot Settings

Channel Plan	RX2 Frequency/Hz	RX2 Datarate	Ping Slot Frequency/Hz	Ping Slot Datarate
CN470	Default: 505300000 Range: 470400000~509700000	Default:0 Range: 0: DR0(SF12,125kHz) 1: DR1(SF11,125kHz) 2: DR2(SF10,125kHz) 3: DR3(SF9,125kHz) 4: DR4(SF8,125kHz) 5: DR5(SF7,125kHz) 6: DR6(SF7,500KHz)	Default: 508300000 Range: 470400000~509700000	Default:2 Range: 0: DR0(SF12,125kHz) 1: DR1(SF11,125kHz) 2: DR2(SF10,125kHz) 3: DR3(SF9,125kHz) 4: DR4(SF8,125kHz) 5: DR5(SF7,125kHz) 6: DR6(SF7,500KHz)
EU868	Default: 869525000 Range: 863000000~870000000	Default: 0 Range: 0: DR0(SF12,125kHz) 1: DR1(SF11,125kHz) 2: DR2(SF10,125kHz) 3: DR3(SF9,125kHz) 4: DR4(SF8,125kHz) 5: DR5(SF7,125kHz) 6: DR6(SF7,250KHz)	Default: 869525000 Range: 863000000~870000000	Default: 3 Range: 0: DR0(SF12,125kHz) 1: DR1(SF11,125kHz) 2: DR2(SF10,125kHz) 3: DR3(SF9,125kHz) 4: DR4(SF8,125kHz) 5: DR5(SF7,125kHz) 6: DR6(SF7,250KHz)
IN865	Default: 866550000	Default: 2 Range:	Default: 866550000	Default: 4 Range:

	Range: 863000000~870000000	0: DR0(SF12,125kHz) 1: DR1(SF11,125kHz) 2: DR2(SF10,125kHz) 3: DR3(SF9,125kHz) 4: DR4(SF8,125kHz) 5: DR5(SF7,125kHz)	Range: 863000000~870000000	0: DR0(SF12,125kHz) 1: DR1(SF11,125kHz) 2: DR2(SF10,125kHz) 3: DR3(SF9,125kHz) 4: DR4(SF8,125kHz) 5: DR5(SF7,125kHz)
RU864	Default: 869100000 Range: 864000000~870000000	Default: 0 Range: 0: DR0(SF12,125kHz) 1: DR1(SF11,125kHz) 2: DR2(SF10,125kHz) 3: DR3(SF9,125kHz) 4: DR4(SF8,125kHz) 5: DR5(SF7,125kHz) 6: DR6(SF7,250KHz)	Default: 868900000 Range: 864000000~870000000	Default: 3 Range: 0: DR0(SF12,125kHz) 1: DR1(SF11,125kHz) 2: DR2(SF10,125kHz) 3: DR3(SF9,125kHz) 4: DR4(SF8,125kHz) 5: DR5(SF7,125kHz) 6: DR6(SF7,250KHz)
KR920	Default: 921900000 Range: 920900000~923300000	Default: 0 Range: 0: DR0(SF12,125kHz) 1: DR1(SF11,125kHz) 2: DR2(SF10,125kHz) 3: DR3(SF9,125kHz) 4: DR4(SF8,125kHz) 5: DR5(SF7,125kHz)	Default: 923100000 Range: 920900000~923300000	Default: 3 Range: 0: DR0(SF12,125kHz) 1: DR1(SF11,125kHz) 2: DR2(SF10,125kHz) 3: DR3(SF9,125kHz) 4: DR4(SF8,125kHz) 5: DR5(SF7,125kHz)
US915	Default: 923300000 Range: 923300000~927500000	Default: 8 Range: 0: DR0(SF10,125kHz) 1: DR1(SF9,125kHz) 2: DR2(SF8,125kHz) 3: DR3(SF7,125kHz) 4: DR4(SF8,500kHz) 8: DR8(SF12,500kHz) 9: DR9(SF11,500kHz) 10: DR10(SF10,500kHz) 11: DR11(SF9,500kHz) 12: DR12(SF8,500kHz) 13: DR13(SF7,500kHz)	Default: 923300000 Range: 923300000~927500000	Default: 8 Range: 0: DR0(SF10,125kHz) 1: DR1(SF9,125kHz) 2: DR2(SF8,125kHz) 3: DR3(SF7,125kHz) 4: DR4(SF8,500kHz) 8: DR8(SF12,500kHz) 9: DR9(SF11,500kHz) 10: DR10(SF10,500kHz) 11: DR11(SF9,500kHz) 12: DR12(SF8,500kHz) 13: DR13(SF7,500kHz)
AU915	Default: 923300000 Range: 923300000~927500000	Default: 8 Range: 0: DR0(SF10,125kHz) 1: DR1(SF9,125kHz) 2: DR2(SF8,125kHz) 3: DR3(SF7,125kHz) 4: DR4(SF8,500kHz)	Default: 923300000 Range: 923300000~927500000	Default: 8 Range: 0: DR0(SF10,125kHz) 1: DR1(SF9,125kHz) 2: DR2(SF8,125kHz) 3: DR3(SF7,125kHz) 4: DR4(SF8,500kHz)

		8: DR8(SF12,500kHz) 9: DR9(SF11,500kHz) 10: DR10(SF10,500kHz) 11: DR11(SF9,500kHz) 12: DR12(SF8,500kHz) 13: DR13(SF7,500kHz)		8: DR8(SF12,500kHz) 9: DR9(SF11,500kHz) 10: DR10(SF10,500kHz) 11: DR11(SF9,500kHz) 12: DR12(SF8,500kHz) 13: DR13(SF7,500kHz)
AS923-1	Default: 923200000 Range: 915000000~928000000	Default: 2 Range: 0: DR0(SF12,125kHz) 1: DR1(SF11,125kHz) 2: DR2(SF10,125kHz) 3: DR3(SF9,125kHz) 4: DR4(SF8,125kHz) 5: DR5(SF7,125kHz) 6: DR6(SF7,250KHz)	Default: 923400000 Range: 915000000~928000000	Default: 3 Range: 0: DR0(SF12,125kHz) 1: DR1(SF11,125kHz) 2: DR2(SF10,125kHz) 3: DR3(SF9,125kHz) 4: DR4(SF8,125kHz) 5: DR5(SF7,125kHz) 6: DR6(SF7,250KHz)
AS923-2	Default: 921400000 Range: 915000000~928000000	Default: 2 Range: 0: DR0(SF12,125kHz) 1: DR1(SF11,125kHz) 2: DR2(SF10,125kHz) 3: DR3(SF9,125kHz) 4: DR4(SF8,125kHz) 5: DR5(SF7,125kHz) 6: DR6(SF7,250KHz)	Default: 921600000 Range: 915000000~928000000	Default: 2 Range: 0: DR0(SF12,125kHz) 1: DR1(SF11,125kHz) 2: DR2(SF10,125kHz) 3: DR3(SF9,125kHz) 4: DR4(SF8,125kHz) 5: DR5(SF7,125kHz) 6: DR6(SF7,250KHz)
AS923-3	Default: 916600000 Range: 915000000~928000000	Default: 2 Range: 0: DR0(SF12,125kHz) 1: DR1(SF11,125kHz) 2: DR2(SF10,125kHz) 3: DR3(SF9,125kHz) 4: DR4(SF8,125kHz) 5: DR5(SF7,125kHz) 6: DR6(SF7,250KHz)	Default: 916800000 Range: 915000000~928000000	Default: 2 Range: 0: DR0(SF12,125kHz) 1: DR1(SF11,125kHz) 2: DR2(SF10,125kHz) 3: DR3(SF9,125kHz) 4: DR4(SF8,125kHz) 5: DR5(SF7,125kHz) 6: DR6(SF7,250KHz)
AS923-4	Default: 917300000 Range: 915000000~928000000	Default: 2 Range: 0: DR0(SF12,125kHz) 1: DR1(SF11,125kHz) 2: DR2(SF10,125kHz) 3: DR3(SF9,125kHz) 4: DR4(SF8,125kHz) 5: DR5(SF7,125kHz) 6: DR6(SF7,250KHz)	Default: 917500000 Range: 915000000~928000000	Default: 2 Range: 0: DR0(SF12,125kHz) 1: DR1(SF11,125kHz) 2: DR2(SF10,125kHz) 3: DR3(SF9,125kHz) 4: DR4(SF8,125kHz) 5: DR5(SF7,125kHz) 6: DR6(SF7,250KHz)

**-End-**