Milesight Gateway

HTTP API Specification

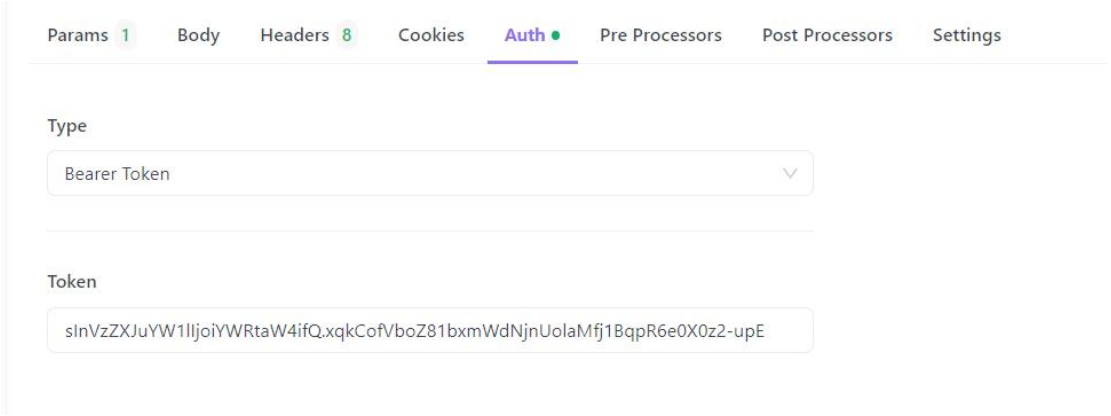
Date	Applicable Firmware Version	Description
---	---	Initial version
Aug. 11, 2023	60.0.0.42/56.0.0.3	Add packet forwarder, NS-general, etc.
May 28, 2024	60.0.0.42-r5/56.0.0.4	Update login account information

Contents

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.



1. API Authentication

1.1 Login

Request Method: POST

Request Address:

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

Request Parameters

Header

Parameter	Required	Description
Content-Type	Yes	application/json

Body

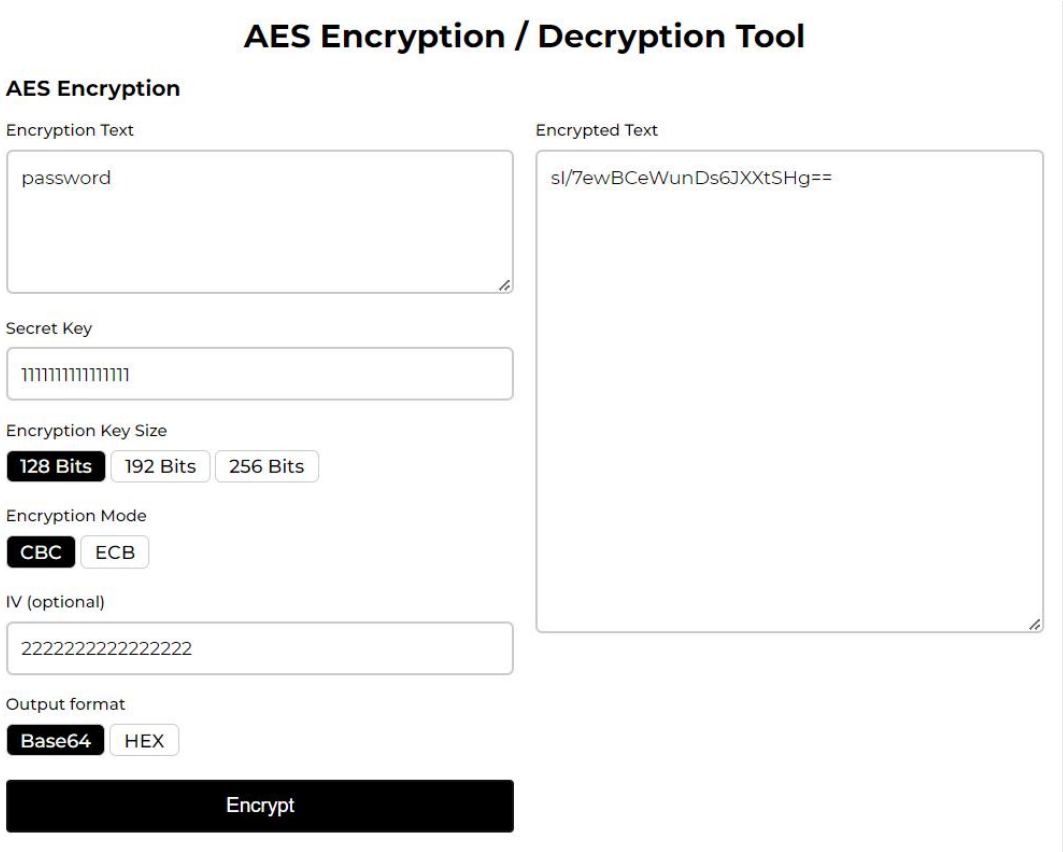
Parameter	Type	Required	Description
username	string	Yes	60.0.0.42-r4 and before: apiuser 60.0.0.42-r5 and later: admin
password	string	Yes	Web login password

Note: for firmware 60.0.0.42-r5/56.0.0.4 and later, the password needs AES encryption before typing in. AES encryption settings:

Parameter	Option
Encryption cipher mode	CBC

Encryption Key Size	128 bits
IV (initial Vector)	2222222222222222
Secret key	1111111111111111
Output format	Base64

Example: https://anycript.com/crypto



Response Parameters

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

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": "sI/7ewBCeWunDs6JXXtSHg=="
```

```
}
```

Response Example

```
{
"jwt":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiJsb3JhLWFwcC1zZXJ2ZXIiLCJleHAiOjE2ODY1NDE1NDAsImlzcyI6ImxvcmEtYXBwLXNlcnZlciIsIm5iZiI6MTY4NjQ1NTE0MCwic3ViIjoidXNlciIsInVzZXJuYW1lIjoiYXBpdXNlciJ9.-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 Loriot: 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/RU864/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	**Examples:** 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 **Note:** 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\nADA5MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQW1hem9uMRkwFwYDVQQDEx
BBbWF6\nb24gUm9vdCBDQSAxMB4XDTE1MDUyNjAwMDAwMFoXDTM4MDExNzAwMDAw
MFowOTEL\nMAkGA1UEBhMCVVMxDzANBgNVBAoTBkFtYXpvbjEZMBcGA1UEAxMQQW1he
m9uIFJv\nb3QgQ0EgMTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALJ4gHHKeN
Xj\nca9HgFB0fW7Y14h29Jlo91ghYPl0hAEvrAIthtOgQ3pOsqTQNroBvo3bSMgHFzZM\n9O6II8
c 6zf1tRn4SWiw3te5djgdYZ6k/oI2peVKVuRF4fn9tBb6dNqcmzU5L/qw\nIFAGbHrQgLKm
a/sRxmPUDgH3KKHOVj4utWp
UhnMJbulHheb4mjUcAwhmahRWa6\nVOujw5H5SNz/0egwLX0tdHA114gk957EWW67c4cX8j
JGKLhD
```

rcdqsq08p8kDi1L\n93FcXmn/6pUCyziKrlA4b9v7LWIbxcceVOF34GfID5yHI9Y/QCB/IIDEgEw
OyQm\njgSubJrIqg0CAwEAAaNCMEAwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBA
MC\nAYYwHQYDVR0OBBYEFIQYzIU07LwMlJQuCFmcx7IQTgoIMA0GCSqGSIb3DQEBCwUA\n
A4IBAQCY8jdaQZChGsV2USggNiMOruYou6r4lK5IpDB/G/wkjUu0yKGX9rbxenDI\nU5PMCCjjm
CXPI6T53iHTfIUJrU6adTrCC2qJeHZERxhlbI1Bjjt/msv0tadQ1wUs\nN
gDS63pYaACbvXy8MWy7Vu33PqUXHeeE6V/Uq2V8viTO96LXFvKWlJbYK8U90vv\no/ufQJVtM
VT8QtPHRh8jrdkPSHCa2XV4cdFyQzR1bldZwgJcJmApzyMZFo6IQ6XU\n5MsI yMRQ
hDKXJioaldXgjUkK642M4UwtBV8ob2xJNDd2ZhwLnoQdeXeGADbkpy\nrqXRfboQnoZsG4q5W
TP468SQvvG5\n-----END CERTIFICATE-----",
"CAName": "AmazonRootCA1 (4).pem",
"TLSCert": "-----BEGIN
CERTIFICATE-----\nMIIDWjCCAkKgAwIBAgIVAPty/swo3PehJhh6CfCx12SsxShoMA0GCSqGSIb3
DQEB\nCwUAME0xSzBJBgNVBAsMQkFtYXpvbiBXZWIgU2VydmljZXMgTz1BbWF6b24uY29t\nI
EluYy4gTD1TZWF0dGxlIFNUPVdhc2hpbmd0b24gQz1VUzAeFw0yMzA1MjkwNjMw\nNTdaFw
00OTEyMzEyMzU5NTlaMB4xHDAaBgNVBAMME0FXUyBJb1QgQ2VydGlmaWNh\ndGUwggEi
MA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC5yymzegf2HqAjPZ6F\nOO7G95cCMbsN
qT2HhtF3wSKxV2Z3Uj4zCEReaFmHwg7ppFciyutQefrHbe
GSGk8\nfl98qQ3q1z8cDy1222kt7aB3r5/MeHU1d9hTk2rTDO4IridXsDgOxbwuVgIp37eH\nX/N
evqVescjO4EjlqKvjc1p 2f9SsZiGVA9iIfosrcmPyYX0laPftz338CXoaGGr\nrS gb8 c
bF9vPyZrGxDPSy37f3wDmuVNAS2dE 4SJ9cJdbWqTevacyYYIwF3smG\na67brqzrRiAd7sUlsv9
/a2Q/boT0rmiBKXphi9ZVvbjIITxqsr4VDePsI77Uf1p\nw
nTAgMBAAGjYDBeMB8GA1UdIwQYMBaAFGyF3xnSijZiJpDAvKAEE4OQKnhlMB0G\nA1UdDg
QWBBSNMZUVY726CxNCkyw7d8b5zKjrSDAMBgNVHRMBAf8EAjAAMA4GA1Ud\nDwEB/wQE
AwIHgDANBgkqhkiG9w0BAQsFAAOCAQEAY8YLjxbtZEtm5E
M2W35AjCV\nyMBaM3av8xUTc1USUol2WRwuN8z7jBv/EE7Z7SbSY8nuocdevi7z33n6SgSbXX
K/\nOM3oNdC5bKKbELSgUIxUzSItHzEF5tVySRMXw5Bou4uZZwejHYai5waSB1kEzoVH\nCFtYC
TCr/eR4/DDjSrBbLJf72rdrvCCRHLxlc0/gV1PaIb8ipgdy8Oo jagop3CN\ngh6Y
B8eUVdHjoBWG7lt46YB2XRnkOAoHmou3x
2zgL4XtHVlqSeoofVIl/Ui9mH\nQZaLnkipn0e/WTkPDe6yNt44uprWiMjKI1xAEVfRptnNoeK
RzTlMoeYvTILHg: =\n-----END CERTIFICATE-----\n",
"certName":
"c4f6e6184319b7b67ed55ace8e4ab11c47ccc1bf4e764142b328b190ab5d572d-certificate.pe
m.crt",
"TLSKey": "-----BEGIN RSA PRIVATE
KEY-----\nMIIEpgIBAAKCAQEAucsps3oH9h6gIz2ehTjuxveXAjG7Dak9h4bRd8EisVdmd1I
\nMwhEXmhZh8IO6aRXIsrrUHn6x23vhkhpPH5ffKkN6tc/HA8tdttpLe2gd6
fzHh1\nNXfYU5Nq0wzuCK4nV7A4DsW8LlYCKd
3h1/zXr6lXrHIzuBI5air43Naftn/UrGY\nhlQPYiH6LK3Jj8mF9JWj37c99/Al6Ghhq60voG/PnPmxf
bz8maxsQz0st
398A5r\nlTQEtnRPuEifXCXW1qk3r2nMmGCMBdR5hmuu266s60YgHe7FJbL/fv2tkP26E9K5\n
ogSl6YYvWVb24yCE8arK FQ3j7CO
1H9acPp0wIDAQABAoIBAQCNjqKGMYsRVcIz\nN4apAHjFNe1jaZqYTQMpWGoOcSPKtMnITex
AG6NjvuPAEPYlw
X0B2GE3ntZKYTe\n7hroUWzVYUTOUhOugE3mikO4D548sp3xPzIme9phyycQCIDwMiidohEEC

```
FpZf4SE\nfQSra0rVPDJpVNGn9bD08i My93
WTPfjiACDNENH8JlskKHQQYq8q18gmSjK/Z1\n1I4vtn0B7sVC/LGyrC1koxeVAZAomnC/B9fkHl
X9fMKCwZ0HFyaYKJFi9bAZqZop\nqHc
jzHkGuvTRHHTkyrZDBgIiwqE58UQz0vEpH80b2zq7yXZesV2SAJk1H5cGPxo\neUMInlp5AoGBA
NpAnD KZQXGawmnKWz
RtamYKIkbifHg4QoSkqk2dklil17Bi3e\ngrNWvw32jdYoIa9gAAxlHh5hbjTeB
LKWteLQMgcvvUhsT6BRuhfXmPiPXOSsjcm\neiIAYRYHOVNua85n3pZA53m81L5FrmVzYLjioh
Mf3yMdNFpryoSkDgZFAoGBANnt\nZACIaS9H4tk3tAJzZcWvNp6Xo5SO
JOnEpxda4fHI5uumM0JcRJSphqOxk QC9UH\nITqWWZ1ojvoZdh
xpTF9ezPXmQU1ePbJDJU71YFNYq8/Z0kHWARVsommBB/54bXQ\n6qKZCqezpP3qShdczq
S1U1sPB0/PDbZmpj26903AoGBAMvsJpGwC6gTtps0Vx6B\nj0qr9gpJo6sDXbyzsIaB9RJRYzDo6
Mbzl8mpCi1v9zjbmXgeusZ58RXIJPBCPk8\nPRi8AfF3 mwr6OIQBGLGCPmmv
gmnuGsz9zJIe1lHoj32T7oCi 1wRIb/2Mw1Dvy\nS Mzzl
0Hdc/SFVb9cNtTbVJAoGBAIO3fGVhPYoJPfAiPde5XcrlAvNKHeG9pDrC\nIXqG8GZJk5kWJ17OF
RyI/cJRh0UxgCVKwOfDVfjQ6JH7b6zqVzkZaazE48UFM86d\nZ9CPMJWNa1cW0FuQZ79
IE/6YBWLPkSg66xJ4KDDvc1c4C5ZGc0LV41TAQ1YzSX\nw1wu6hJzAoGBAMnZsAUtbcCVgK5X
OBLTCJ1hIwMwB X2WktvMWpsSIy55RcQG3uQ\n67CYu7DKRktmcfFp5sJgr4 EV4pSb8
adTs4jSdR7YczkBg5VqgMffzXh10WIICM\noV5nqvg0uVpYNeHKpg9JfQpo5fbyNvdC5BsuA4h5
AbCypj7qSJdZaCx2\n-----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/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
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

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/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
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, bacnetip Note: when deleting a HTTPS integration, select http

Request Example

https://192.168.23.150:8080/api/urapplications/6/integrations/bacnetip

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        }\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 = 1;\n            i += 1;\n        }\n        // LORAWAN
CLASS TYPE\n        else if (channel_id === 0xff && channel_type === 0x0f)
{\n            decoded.lorawan_class = 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        // 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            // ℃\n            decoded.temperature = readInt16LE(bytes.slice(i, i + 2)) /
```

```
10;\n                i += 2;\n            }\n            // HUMIDITY\n            else if (channel_id === 0x04
&& channel_type === 0x68) {\n                decoded.humidity = bytes[i] / 2;\n                i +=
1;\n            }\n            // HISTORY\n            else if (channel_id === 0x20 && channel_type ===
0xce) {\n                var data = {};\n                data.timestamp = readUInt32LE(bytes.slice(i, i
+ 4));\n                data.temperature = readInt16LE(bytes.slice(i + 4, i + 6)) /
10;\n                data.humidity = bytes[i + 6] / 2;\n                i +=
7;\n\n                decoded.history = decoded.history ||
[];\n                decoded.history.push(data);\n            } else
{\n                break;\n            }\n        }\n\n        return decoded;\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
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 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",
            "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 *\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        }\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 = 1;\n            i += 1;\n        }\n        // LORAWAN
CLASS TYPE\n        else if (channel_id === 0xff && channel_type === 0x0f)
{\n            decoded.lorawan_class = 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        // 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            // ℃\n            decoded.temperature = readInt16LE(bytes.slice(i, i + 2)) /
10;\n            i += 2;\n        }\n        // HUMIDITY\n        else if (channel_id === 0x04
&& channel_type === 0x68) {\n            decoded.humidity = bytes[i] / 2;\n            i +=
1;\n        }\n        // HISTORY\n        else if (channel_id === 0x20 && channel_type ===
0xce) {\n            var data = {};\n            data.timestamp = readUInt32LE(bytes.slice(i, i
+ 4));\n            data.temperature = readInt16LE(bytes.slice(i + 4, i + 6)) /
10;\n            data.humidity = bytes[i + 6] / 2;\n            i +=
```

```
7;\n\n            decoded.history = decoded.history ||
[];\n             decoded.history.push(data);\n         } else
{\n             break;\n         }\n     }\n\n     return decoded;\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
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 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",
    "testEnabled": false,
    "fPort": 0
}
```


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\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\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\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


```
{
    "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           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\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\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\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\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\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"
}
```

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	
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 Default RX2/PingSlot Settings
pingSlotFreq	Integer	No	PingSlot Frequency for Class B, unit: Hz refer to Default RX2/PingSlot Settings
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 Default RX2/PingSlot Settings
rxFreq2	Integer	No	RX2 channel frequency, unit: Hz refer to Default RX2/PingSlot Settings
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 **Note:** 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 Default RX2/PingSlot Settings
pingSlotFreq	Integer	No	PingSlot Frequency for Class B, unit: Hz refer to Default RX2/PingSlot Settings
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 Default RX2/PingSlot Settings
rxFreq2	Integer	No	RX2 channel frequency, unit: Hz refer to Default RX2/PingSlot Settings
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 **Note:** 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, rtru 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, rtru 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-eb1ff5d9bfdf",
            "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-eb1ff5d9bfdf",
        "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 Default RX2/PingSlot Settings
frequency	Integer	No	Downlink frequency, unit: Hz the same as pingslot frequency, refer to Default RX2/PingSlot Settings
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 Default RX2/PingSlot Settings
frequency	Integer	Yes	Downlink frequency, unit: Hz the same as pingslot frequency, refer to Default RX2/PingSlot Settings
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-eb1ff5d9bfdf/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-a897078f5400/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.
rssi	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/"
}
```

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:

{}

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 Range: 863000000~870000000	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)	Default: 866550000 Range: 863000000~870000000	Default: 4 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)
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) 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)
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)	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)		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-