

# WisDM API Developer Manual

## Contents

|                                              |    |
|----------------------------------------------|----|
| Overview .....                               | 2  |
| WisDM API Key .....                          | 3  |
| Create WisDM API Key .....                   | 3  |
| Manage API Keys .....                        | 7  |
| Edit an API Key .....                        | 7  |
| Regenerate an API Key .....                  | 8  |
| Delete an API Key .....                      | 9  |
| Accessing the WisDM API .....                | 10 |
| Authentication Header .....                  | 10 |
| Test Your Connection .....                   | 10 |
| API Reference .....                          | 11 |
| Get API Version .....                        | 11 |
| Get Organization Information .....           | 12 |
| List Locations .....                         | 13 |
| Create Location (Basic Station) .....        | 13 |
| List of Gateways by Location .....           | 17 |
| Register Gateway .....                       | 18 |
| Get Gateway Status by Location .....         | 20 |
| Get Gateway Battery Status by Location ..... | 22 |
| Get Single Gateway Status .....              | 24 |
| Get Gateway Performance Metrics .....        | 25 |
| Get Gateway Battery Information .....        | 27 |
| Enable Solar Battery Monitoring .....        | 28 |
| Send Asynchronous Command .....              | 29 |
| Query Asynchronous Command Result .....      | 30 |
| Reboot Gateway .....                         | 31 |
| Reset Gateway Password .....                 | 32 |
| Upgrade Gateway Firmware .....               | 33 |
| Get Network Priority .....                   | 33 |
| Set Network Priority .....                   | 34 |
| Get LAN Configuration .....                  | 35 |

|                                           |    |
|-------------------------------------------|----|
| Configure LAN Interface.....              | 36 |
| Get WAN Configuration .....               | 37 |
| Configure WAN Interfaces .....            | 39 |
| Get Gateway Syslog Configuration .....    | 41 |
| Update Gateway Syslog Configuration ..... | 42 |
| Get Gateway Syslog Download URLs .....    | 43 |
| Support .....                             | 44 |

## Overview

The **WisDM API** enables secure, programmatic access to WisDM data through **API Key authentication**. It allows third-party systems to integrate, automate, and analyze operations using WisDM's **RESTful interface**.

**Availability:** Enterprise subscription tier only

**Scope:** Bound to an Organization and managed by its Administrator

**Key Control:** Each API Key has an independent permission scope and can be activated, revoked, or regenerated at any time.

To begin integration, generate an **API Key**, which will serve as the authentication credential for your external system.

# WisDM API Key

## Create WisDM API Key

1. **Creating an API Key**
2. Log in to **WisDM**.
3. Navigate to **Settings** → **API Key** from the left-hand menu.

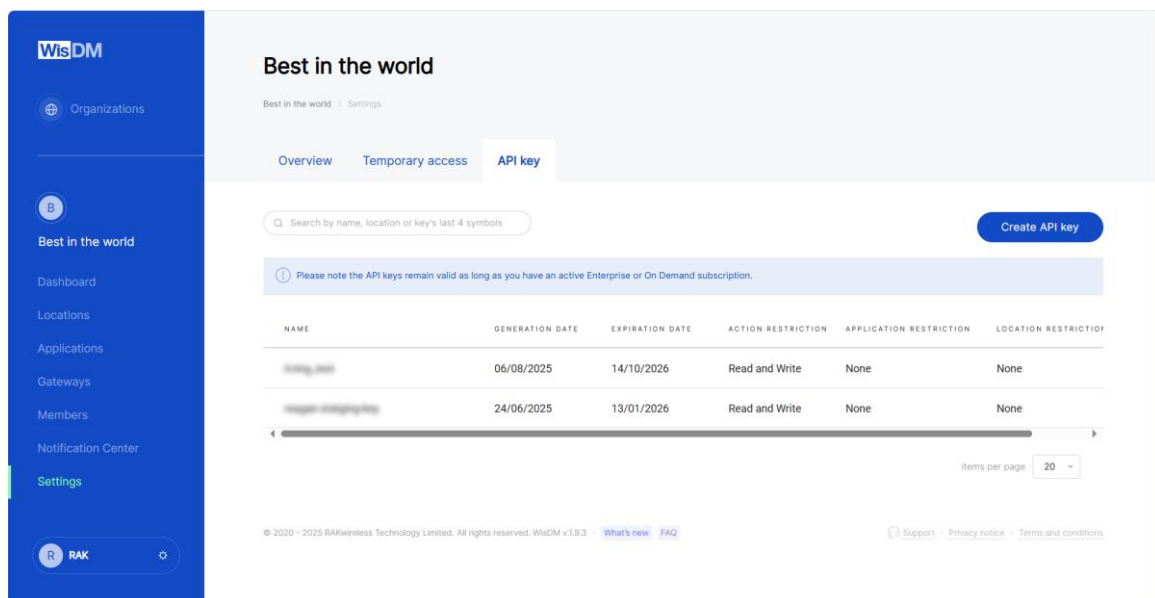


Figure 1 WisDM API Management

4. Click **Create API Key** (top right corner).
5. Enter a **name** for the key.
6. Configure the following settings:
  - **Action restrictions**
    - **Read only:** Grants access to GET methods (data retrieval only).
    - **Read and Write:** Grants full access, including POST and DELETE methods.
  - **Application restrictions**
    - **None:** No restriction, the key can be used from any source.
    - **HTTP referrers (websites):** Restrict API use to specific web origins.
    - **IP addresses (web servers, cron jobs, etc):** Allow API use only from specified IP addresses.
  - **Location restrictions**
    - Optionally limit API use to one or several locations.
  - **Key Lifespan**

- Choose a validity period. The key will automatically expire after this period.

[Overview](#)
[Temporary access](#)
[API key](#)

[< Back to list](#)

### Create API key

API key name

Name

Enter API name

0/32

API key settings

#### Action restrictions

An action restriction control which methods you can use with your API key. Read only allows to interact with GET methods, while Read and write enables access to POST and DELETE as well.

☒ Read only
 ☐ Read and write

#### Application restrictions

An application restriction controls which websites, IP addresses, or applications can use your API key. You can set one application restriction per key.

☒ None
 ☐ HTTP referrers (web sites)
 ☐ IP addresses (web servers, cron jobs, etc.)

#### Location restrictions

You can restrict action with this key to one or several locations.

##### Select allowed locations

Search a location
 

▼

#### Key lifespan

90 days

▼

The key will be deactivated automatically after the selected time period. You will need to regenerate it.

**Figure 2 New API Key Configuration**

- Click **Generate Key**.
- Enter your **WisDM account password** and the **6-digit verification code** (if two-factor authentication is enabled).
- Click **Generate Key** again to confirm.



## Create API key?

Please complete the following authentication steps to  
proceed with API key generation

Confirm your password

and enter 6-digit code from your authenticator app

Generate key

Cancel

*Figure 3 API Key Creation Confirmation*

Once generated, a pop-up will display your new API Key.



Here is your API key

f167b493e03dd01073c1846b0d976970 

Please note the key is only shown once. Copy it and store in a secure space before you leave the page.

I have copied the key

Figure 4 WisDM API Key Created

**⚠ Important:**

The API Key is displayed **only once**.

Copy and store it securely before leaving the page.

Lost keys **cannot be retrieved** and must be regenerated.

## Manage API Keys

After creation, all API Keys can be managed from **Settings** → **API Key**. Each key can be **edited, regenerated, or deleted** as required.

## Edit an API Key

1. Select the API Key you wish to modify.
2. Update its name or permission settings.
3. Click **Save Changes** to apply updates.

Overview

Temporary access

API key

< Back to list

Edit API key

API key name

API key settings

Regenerate

Delete

Name

my-key

6/32

API key

..... 6970

Created by

RAK 17.10.2025 16:27

Action restrictions

An action restriction control which methods you can use with your API key. Read only allows to interact with GET methods, while Read and write enables access to POST and DELETE as well.

☐

Read only

☒

Read and write

☒ Reboot gateway

☒ Reset password

☒ Edit network interface settings

☒ Add gateway to location

Application restrictions

An application restriction controls which websites, IP addresses, or applications can use your API key. You can set one application restriction per key.

☒ None

☐ HTTP referrers (web sites)

☐ IP addresses (web servers, cron jobs, etc.)

Location restrictions

You can restrict action with this key to one or several locations.

Select allowed locations

Q Search a location

Key lifespan

90 days

The key will be deactivated automatically after the selected time period. You will need to regenerate it.

Figure 5 Edit API Key settings

### Regenerate an API Key

Regenerate your API Key when it **expires** or if you **suspect it has been compromised**.  
 The new key will retain all previous configurations.

1. Click the **Regenerate** icon (top right corner).
2. Enter your credentials.
3. Click **Regenerate API Key** to confirm.



## Regenerate API key?

Are you sure you want to regenerate this API key? Your previous key will become invalid. You will need to change the current key to the newly generated in your code in order to continue using RAK API.

Confirm your password

and enter 6-digit code from your authenticator app

Regenerate API key

Cancel

*Figure 6 Regenerate API key*

### Delete an API Key

Delete unused or replaced keys.

Once deleted, all systems using this key will **immediately lose API access**.

1. Click the **Delete** icon (top right corner).
2. Confirm your action.
3. Click **Delete API Key** to permanently remove it.



### Delete API key?

Are you sure you want to delete this API key? You will need to generate new API key and add it to your code in order to continue using RAK API.

Confirm your password

and enter 6-digit code from your authenticator app

Delete API key

Cancel

*Figure 7 Delete API Key Confirmation*

### Important:

Deletion **cannot be undone**.

Any integration using this key will instantly lose access to the WisDM API.

## Accessing the WisDM API

You can access the WisDM API using any REST-compatible client, such as your backend service, automation script, or integration framework. It uses standard HTTP methods (GET, POST, etc.) and exchanges data in JSON format for interoperability with most systems.

All WisDM API requests must be sent to the following domain:

```
https://api.wisdm.network
```

Each API endpoint is appended to this base URL. For example:

```
GET https://api.wisdm.network/v4/organization
```

### Authentication Header

Every request requires a valid **API Key** for authentication.

```
X-API-KEY: <Your_API_Key>
Accept-Encoding: gzip
```

- X-API-KEY: Identifies your organization and permission scope.
- Accept-Encoding: gzip: Enables compressed responses for better efficiency.

### Test Your Connection

Before integrating, verify that your API Key and environment are correctly configured.

#### Request Example

```
GET https://api.wisdm.network/v4/organization
X-API-KEY: <Your_API_Key>
Accept-Encoding: gzip
```

#### Response Example

```
{
  "id": "7e0abc48-63b4-475c-a632-ea5bcde5c66d",
  "name": "Best in the world"
}
```

If your response matches the example above, it confirms that:

- The WisDM API is reachable.
- Your API Key is valid and active.
- You can proceed with further API operations.

#### NOTE

- If the API returns “403 Action is not allowed”, verify that your key has the required permission level.
- Each key belongs to one Organization and cannot access others.
- Generate separate keys for each system for better traceability and audit control.

## API Reference

This section lists all available WisDM API endpoints and their usage details. All endpoints are versioned under /v4/ and require an authentication header containing a valid API Key.

### Get API Version

Retrieves the current WisDM API version for connection testing and environment verification. Requires a valid API Key with **Read** permission.

#### Endpoint

```
GET /v4/version
```

#### Request Example

```
GET https://api.wisdm.network/v4/version
X-API-KEY: <Your_API_Key>
Accept-Encoding: gzip
```

#### Response Example

```
{
  "version": "v4.0.0",
  "commitId": "a12b3c4"
}
```

#### Response Fields

| Field    | Type   | Description                      |
|----------|--------|----------------------------------|
| version  | string | Current WisDM API version number |
| commitId | string | Internal system build identifier |

#### Error Codes

| HTTP Code | Meaning               | Description                       |
|-----------|-----------------------|-----------------------------------|
| 400       | Bad Request           | Invalid or missing parameters     |
| 403       | Forbidden             | API key lacks required permission |
| 500       | Internal Server Error | System error                      |

## Get Organization Information

Retrieves information about the Organization associated with your API Key. Requires a valid API Key with **Read** permission.

### Endpoint

```
GET /v4/organization
```

### Request Example

```
GET https://api.wisdm.network/v4/organization
X-API-KEY: <Your_API_Key>
Accept-Encoding: gzip
```

### Response Example

```
{
  "id": "f53d5b7d-2413-44f3-b206-8295e16a19cd",
  "name": "yolo"
}
```

### Response Fields

| Field | Type   | Description            |
|-------|--------|------------------------|
| id    | string | Organization unique ID |
| name  | string | Organization name      |

### Error Codes

| HTTP Code | Meaning               | Description                       |
|-----------|-----------------------|-----------------------------------|
| 400       | Bad Request           | Invalid or missing parameters     |
| 403       | Forbidden             | API key lacks required permission |
| 404       | Resource Not Found    | Requested resource does not exist |
| 500       | Internal Server Error | System error                      |

## List Locations

Lists all Locations under an Organization, returning their IDs and names. Requires a valid API Key with **Read** permission.

### Endpoint

```
GET /v4/organization/{organizationId}/locations
```

### Request Example

```
GET https://api.wisdmanetwork.com/v4/organization/{organizationId}/locations
X-API-KEY: <Your_API_Key>
Accept-Encoding: gzip
```

### Response Example

```
{
  "total": 2,
  "locations": [
    { "id": "4fed110b-461b-4815-8bfc-4866b3b74c27", "name": "Factory A" },
    { "id": "58bbb735-dbd4-4787-9c8e-b9cacf098d80", "name": "Factory B" }
  ]
}
```

### Response Fields

| Field | Type    | Description               |
|-------|---------|---------------------------|
| total | integer | Total number of locations |
| id    | string  | Location ID               |
| name  | string  | Location name             |

## Create Location (Basic Station)

Creates a new Location under an Organization and configures it for **Basic Station** work mode in a single call. Requires a valid API Key with **WRITE\_ADD\_LOCATION\_TO\_ORGANIZATION** permission.

### Endpoint

```
POST /v4/organization/{organizationId}/locations
```

## Request Example

```
POST https://api.wisdm.network/v4/organization/{organizationId}/locations
X-API-KEY: <Your_API_Key>
Accept-Encoding: gzip
Content-Type: application/json

{
  "name": "Factory A",
  "address": {
    "country": "Bulgaria",
    "addressLine1": "1 Industrial St",
    "city": "Sofia",
    "state": "Sofia",
    "zip": "1000",
    "latitude": 42.6977,
    "longitude": 23.3219
  },
  "workMode": "Basics station",
  "enableCriticalFwAutoUpdate": true,
  "basicStation": {
    "serverType": "CUPS",
    "serverURL": "https://cups.example.com",
    "serverPort": 443,
    "authenticationMode": "TLS Server Authentication",
    "trustCACertificate": "-----BEGIN CERTIFICATE-----...-----END CERTIFICATE-----",
    "useIndividualKeys": true
  },
  "advancedFrequencySettings": {
    "region": "EU868",
    "regionVariation": "default",
    "isLoRaWANPublic": true,
    "frequencySubBand": "G1"
  },
  "channelsNumber": 8
}
```

## Response Example

```
{
  "locationId": "58bbb735-dbdc-4787-9c8e-b9cacf098d80",
  "organizationId": "f53d5b7d-2413-44f3-b206-8295e16a19cd",
  "name": "Factory A",
  "workMode": "Basics station"
}
```

## Request Fields

| Field                      | Type    | Description                                                                   |
|----------------------------|---------|-------------------------------------------------------------------------------|
| name                       | string  | Human-friendly location name displayed in WisDM UI (min 4, max 50 characters) |
| address                    | object  | Optional. If omitted, the organization's registered address is used           |
| workMode                   | string  | Work mode of the location. Only <b>Basics station</b> is supported            |
| enableCriticalFwAutoUpdate | boolean | Enables automatic installation of critical firmware updates                   |
| basicStation               | object  | Basic Station server configuration                                            |
| advancedFrequencySettings  | object  | Frequency plan configuration                                                  |
| channelsNumber             | integer | Number of channels supported (8 or 16). Default: 8                            |

## Address Object (address)

| Field        | Type   | Description          |
|--------------|--------|----------------------|
| country      | string | Country name         |
| addressLine1 | string | Street address       |
| city         | string | City                 |
| state        | string | State / Region       |
| zip          | string | ZIP / Postal code    |
| latitude     | number | Latitude coordinate  |
| longitude    | number | Longitude coordinate |

## Basics Station Object (basicStation)

| Field      | Type    | Description                |
|------------|---------|----------------------------|
| serverType | string  | CUPS-BOOT / CUPS / LNS     |
| serverURL  | string  | Fully qualified server URL |
| serverPort | integer | TCP port (1–65535)         |

| Field              | Type    | Description                                                                                                                         |
|--------------------|---------|-------------------------------------------------------------------------------------------------------------------------------------|
| authenticationMode | string  | None / TLS Server Authentication / TLS Server and Client Authentication / TLS Server Authentication and Client Token Authentication |
| trustCACertificate | string  | PEM-encoded CA certificate                                                                                                          |
| useIndividualKeys  | boolean | Enables per-gateway credentials                                                                                                     |
| clientCertificate  | string  | PEM client certificate (required for shared TLS client auth)                                                                        |
| clientKey          | string  | PEM client private key (required for shared TLS client auth)                                                                        |
| token              | string  | Token used for client authentication                                                                                                |

### Response Fields

| Field          | Type   | Description                               |
|----------------|--------|-------------------------------------------|
| locationId     | string | Unique identifier of the created location |
| organizationId | string | Organization ID that owns the location    |
| name           | string | Location name                             |
| workMode       | string | Work mode applied to the location         |

### Error Codes

| HTTP Code | Meaning               | Description                                          |
|-----------|-----------------------|------------------------------------------------------|
| 400       | Bad Request           | Invalid or missing parameters                        |
| 403       | Forbidden             | API key lacks required permission                    |
| 404       | Not Found             | Organization not found                               |
| 409       | Conflict              | Location already exists or conflicting configuration |
| 500       | Internal Server Error | System error                                         |

## List of Gateways by Location

Lists all gateways in a specific Location, including model and status. Requires a valid API Key with **Read** permission.

### Endpoint

```
GET /v4/location/{locationId}/gateways
```

### Request Example

```
GET https://api.wisdm.network/v4/location/{locationId}/gateways
X-API-KEY: <Your_API_Key>
Accept-Encoding: gzip
```

### Response Example

```
{
  "total": 1,
  "gateways": [
    {
      "sn": "918100M46020602001",
      "eui": "ac1f09fffe0707a1",
      "name": "RAK7268V2H-918100M46020602001",
      "model": "RAK7268V2H",
      "mac": "AC:1F:09:07:07:A1",
      "status": "ONLINE",
      "lastDyingGasp": "None"
    }
  ]
}
```

### Response Fields

| Field         | Type   | Description                     |
|---------------|--------|---------------------------------|
| sn            | string | Gateway serial number           |
| eui           | string | Gateway EUI                     |
| name          | string | Gateway name                    |
| model         | string | Device model                    |
| mac           | string | MAC address                     |
| status        | string | ONLINE / OFFLINE / PROVISIONING |
| lastDyingGasp | string | Last event timestamp            |

## Register Gateway

Registers an **unassigned gateway** under a specific Location. This API is essential for automated factory provisioning or **field deployment**. Requires a valid API Key with **WRITE\_ADD\_GATEWAY\_TO\_LOCATION** permission.

### NOTE

Before registration, ensure the following conditions are met:

- The gateway is already **whitelisted under the Organization**.
- The target Location operates in **Basics Station mode**.
- The gateway was **pre-provisioned during factory setup**.
- If factory provisioning is incomplete, registration will fail.

### Endpoint

POST /v4/location/{locationId}/gateways

### Request Example

```
POST https://api.wisdm.network/v4/location/4fed110b-461b-4815-8bfc-4866b3b74c27/gateways
X-API-KEY: <Your_API_Key>
Accept-Encoding: gzip
Content-Type: application/json
{
  "sn": "GW123456",
  "eui": "ABCDEF1234567890",
  "name": "Gateway-1",
  "bsHasLocalAuthSuite": true,
  "bsClientCertificate": "-----BEGIN CERTIFICATE-----...",
  "bsClientKey": "-----BEGIN PRIVATE KEY-----...",
  "bsToken": "eyJhbGciOiJIUzI1NiIs..."}
}
```

### Request Parameters

| Field               | Type    | Description                                                                                                                                                                                                                             |
|---------------------|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| sn                  | string  | Gateway Serial Number                                                                                                                                                                                                                   |
| eui                 | string  | Gateway EUI (unique hardware identifier)                                                                                                                                                                                                |
| name                | string  | Gateway name                                                                                                                                                                                                                            |
| bsHasLocalAuthSuite | boolean | Set to false for gateways without pre-configured Basic Station certificates. Set to true if the gateway has factory-preconfigured Basic Station certificates. Only gateways with version <b>v2.2.15 or newer</b> supports this feature. |
| bsClientCertificate | string  | Client certificate for TLS authentication. Required when bsHasLocalAuthSuite is false and location uses 'TLS Server and Client Authentication' mode.                                                                                    |

| Field       | Type   | Description                                                                                                                                                                                              |
|-------------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| bsClientKey | string | Client private key for TLS authentication. Required when bsHasLocalAuthSuite is false and location uses 'TLS Server and Client Authentication' mode.                                                     |
| bsToken     | string | Authentication token for token-based authentication. Required when bsHasLocalAuthSuite is false and location uses 'TLS Server Authentication and Client Token Authentication' mode authentication mode . |

#### NOTE

WisDM verifies the request in several steps:

1. Checks API Key ownership and write permission scope.
2. Confirms that the gateway belongs to the same Organization.
3. Validates that the target Location supports Basics Station mode.
4. If TLS or token-based authentication is required, validates the credentials.
5. Upon success, WisDM establishes a binding between the gateway and the Location, returning a *201 Created* response.

#### Response Example

```
{
  "locationId": "4fed110b-461b-4815-8bfc-4866b3b74c27",
  "sn": "GW123456",
  "eui": "ABCDEF1234567890"
}
```

#### Response Fields

| Field      | Type   | Description                                      |
|------------|--------|--------------------------------------------------|
| locationId | string | ID of the Location the gateway was registered to |
| sn         | string | Gateway serial number                            |
| eui        | string | Gateway EUI                                      |

#### Error Codes

| HTTP Code | Meaning                       | Description                                                                       |
|-----------|-------------------------------|-----------------------------------------------------------------------------------|
| 400       | Missing or invalid parameters | Required fields (sn, eui, name) not provided, or auth mode mismatch with Location |
| 403       | Insufficient permission       | API Key lacks write privileges or the gateway doesn't belong to the Organization  |
| 404       | Not Found                     | Resource Not Found                                                                |
| 406       | Invalid mode                  | Location is not in Basics Station mode, or gateway is not factory-registered      |

| HTTP Code | Meaning      | Description                         |
|-----------|--------------|-------------------------------------|
| 500       | System error | Registration or backend sync failed |

## Get Gateway Status by Location

Retrieves the **operational status** of all gateways under a specific **Location**, with **pagination support** for large deployments. Requires a valid API Key with **Read** permission.

### Endpoint

```
GET /v4/location/{locationId}/gateways/status
```

### Request Example

```
GET https://api.wisdm.network/v4/location/4fed110b-461b-4815-8bfc-4866b3b74c27/gateways/status?pageSize=100
X-API-KEY: <Your_API_Key>
Accept-Encoding: gzip
```

### Response Example

```
{
  "pageSize": 100,
  "totalPages": 3,
  "nextToken": "eyJwYWdlIjoyfQ==",
  "gateways": [
    {
      "sn": "918100RC6050800000",
      "eui": "ac1f09fffe990000",
      "name": "RAK7289V2",
      "model": "RAK7289CV2",
      "mac": "00:3D:C6:FD:58:C5",
      "status": "ONLINE",
      "lastDyingGasp": "None",
      "system": {
        "uptime": 10372,
        "cpu": 33,
        "memory": 51
      },
      "gps": {
        "longitude": 174.966721,
        "latitude": -38.0526032
      },
      "network": {
        "lan": {
          "subnet": "192.168.230.1/24",
          "ethernet": {
            "enabled": false
          },
          "wifi": {
            "enabled": false,
            "ssid": "ltDhgPVEZUqClu4=",
            "encryption": "NONE",
            "key": "xZ5+9j2p8+4=",
            "channel": "Auto",
            "hidden": false
          }
        }
      }
    }
  ]
}
```

```

    },
    "wan": {
      "active": "CELLULAR",
      "ethernet": {
        "enabled": true,
        "priority": 1,
        "online": false,
        "uptime": 0
      },
      "wifi": {
        "enabled": false,
        "priority": 2
      },
      "cellular": {
        "enabled": true,
        "priority": 3,
        "online": true,
        "ipv4AddrMask": "100.94.77.194/30",
        "uptime": 10299,
        "sim_present": true,
        "imsi": "234106459829513",
        "iccid": "89883070000054069701",
        "imei": "862869033261012",
        "signalLevel": 1,
        "rsrp": -102,
        "rsrq": -6,
        "sinr": 18,
        "rssi": -76,
        "rat": "LTE",
        "networkType": "FDD LTE",
        "autoSelect": true,
        "plmn": "53001"
      }
    },
    "lora": {
      "activeNodes": 0,
      "txPackets": 2,
      "rxPackets": 119
    },
    "storage": {
      "healthy": true,
      "total": 15185,
      "free": 15048,
      "used_percent": 0
    }
  },
  .....
]
}

```

## Response Fields

| Field      | Type    | Description                                                                                     |
|------------|---------|-------------------------------------------------------------------------------------------------|
| pageSize   | integer | Number of results per page                                                                      |
| totalPages | integer | Total available pages                                                                           |
| nextToken  | string  | Pagination token for next page This field is omitted if there are no additional pages to fetch. |
| gateways   | array   | List of gateway status summaries                                                                |

## Gateway Object Fields

| Field         | Type   | Description                                      |
|---------------|--------|--------------------------------------------------|
| sn            | string | Gateway serial number                            |
| eui           | string | Gateway EUI                                      |
| name          | string | Gateway name                                     |
| model         | string | Device model                                     |
| mac           | string | MAC address                                      |
| status        | string | Current status (ONLINE / OFFLINE / PROVISIONING) |
| lastDyingGasp | string | last dying gasp event date time                  |
| system        | object | SystemStatistics                                 |
| gps           | object | GPS status of the gateway                        |
| network       | object | Network status of the gateway                    |
| lora          | object | LoRa packet statistics of the gateway            |
| storage       | object | Storage status of the gateway                    |

## Get Gateway Battery Status by Location

Retrieves the solar battery health and power metrics for all gateways under a specific Location. Requires a valid API Key with **Read** permission.

### Endpoint

```
GET /v4/location/{locationId}/gateways/battery
```

### Request Example

```
GET https://api.wisdm.network/v4/location/f53d5b7d-2413-44f3-b206-8295e16a19cd/gateways/battery?pageSize=100
X-API-KEY: <Your_API_Key>
Accept-Encoding: gzip
```

## Response Example

```
{
  "pageSize": 100,
  "totalPages": 2,
  "nextToken": "eyJwYWdlIjoyfQ==",
  "gateways": [
    {
      "sn": "9181016S6041224001",
      "battery": {
        "temperature": 28,
        "voltage": 22.03,
        "soc": 48,
        "current": 60,
        "soh": 100,
        "cycles": 3,
        "remainCapacity": 24410,
        "fullChargeCapacity": 50000,
        "mode": "CHARGING"
      }
    }
  ]
}
```

## Response Fields

| Field      | Type    | Description                                                                                     |
|------------|---------|-------------------------------------------------------------------------------------------------|
| pageSize   | integer | Number of results returned per page                                                             |
| totalPages | integer | Total number of pages available                                                                 |
| nextToken  | string  | Pagination token for next page This field is omitted if there are no additional pages to fetch. |
| gateways   | array   | List of gateways with associated battery data                                                   |

## Battery Object Fields

| Field              | Type    | Description                                                   |
|--------------------|---------|---------------------------------------------------------------|
| temperature        | integer | Battery temperature (°C)                                      |
| voltage            | integer | Battery voltage (V)                                           |
| soc                | integer | State of Charge (percentage)                                  |
| current            | integer | current of the solar battery in mA                            |
| soh                | integer | State of Health (percentage)                                  |
| cycles             | integer | Cycle times of the solar battery                              |
| remainCapacity     | integer | Remaining capacity of the solar battery in mAh                |
| fullChargeCapacity | integer | Full charge capacity of the solar battery in mAh              |
| mode               | string  | Battery working mode (CHARGING / DISCHARGING / FULLY_CHARGED) |

## Get Single Gateway Status

Retrieves the **real-time operational status** of a specific gateway, including its **system performance**, **network interfaces**, **GPS state**, **LoRa activity**, and **storage health**. Requires a valid API Key with **Read** permission.

This API is ideal for targeted monitoring or troubleshooting at the device level.

- When invoked, WisDM returns the most recent runtime snapshot for the gateway identified by its **Serial Number (SN)**.
- If the gateway is offline, the API returns the **latest cached state** reported before disconnection.

### Endpoint

```
GET /v4/gateway/{sn}/status
```

### Request Example

```
GET https://api.wisdm.network/v4/gateway/918100RN6040814001/status
X-API-KEY: <Your_API_Key>
Accept-Encoding: gzip
```

### Response Example

```
{
  "sn": "918100RN6040814001",
  "eui": "ac1f09fffe16882b",
  "name": "RAK7268CV2",
  "model": "RAK7268CV2",
  "mac": "AC:1F:09:16:88:2B",
  "status": "ONLINE",
  "lastDyingGasp": "None",
  "system": {"uptime": 3142, "cpu": 35, "memory": 51},
  "gps": {
    "state": "GPS_NO_OUTPUT",
    "satellites_number": 0,
    "longitude": 0.0,
    "latitude": 0.0
  },
  "network": {
    "lan": {
      "subnet": "192.168.230.1/24",
      "ethernet": {"enabled": false},
      "wifi": {
        "enabled": true,
        "ssid": "RAK-865",
        "encryption": "WPA2-PSK",
        "key": "test-key",
        "channel": "Auto",
        "hidden": false
      }
    },
    "wan": {
      "active": "ETH",
      "ethernet": {
```

```

        "enabled": true,
        "priority": 3,
        "online": true,
        "ipv4AddrMask": "10.0.10.74/24",
        "uptime": 2544
      },
      "wifi": {"enabled": false, "priority": 1},
      "cellular": {"enabled": true, "priority": 2, "online": false, "uptime": 0}
    },
    "lora": {"activeNodes": 3, "txPackets": 0, "rxPackets": 497},
    "storage": {"healthy": true, "total": 15176, "free": 15139, "used_percent": 0}
  }
}

```

## Response Fields

| Field         | Type   | Description                                      |
|---------------|--------|--------------------------------------------------|
| sn            | string | Gateway serial number                            |
| eui           | string | Gateway EUI                                      |
| name          | string | Gateway name                                     |
| model         | string | Device model                                     |
| mac           | string | MAC address                                      |
| status        | string | Current status (ONLINE / OFFLINE / PROVISIONING) |
| lastDyingGasp | string | last dying gasp event date time                  |
| system        | object | System statistics                                |
| gps           | object | GPS status of the gateway                        |
| network       | object | Network status of the gateway                    |
| lora          | object | LoRa packet statistics of the gateway            |
| storage       | object | Storage status of the gateway                    |

## Get Gateway Performance Metrics

Retrieves historical performance metrics for a specific gateway - including **RSSI**, **SNR**, **uplink**, and **downlink** statistics. Requires a valid API Key with **Read** permission.

### Endpoint

```
GET /v4/gateway/{sn}/perf/{metric}
```

### Request Example

```

GET https://api.wisdm.network/v4/gateway/GW123456/perf/rssi?interval=12h&rssi=2
X-API-KEY: <Your_API_Key>
Accept-Encoding: gzip

```

## Request Parameters

| Parameter | Type   | Description                                  |
|-----------|--------|----------------------------------------------|
| sn        | string | Gateway serial number                        |
| metric    | string | Metric type (rssi / snr / uplink / downlink) |
| interval  | string | Time range                                   |

## Query Parameters

| Parameter | Type   | Description                                                   |
|-----------|--------|---------------------------------------------------------------|
| interval  | string | Time range                                                    |
| rssi      | number | RSSI level bucket to receive (Required only if metric = rssi) |
| snr       | snr    | SNR level bucket to receive (Required only if metric = snr)   |

### Note

When querying RSSI or SNR metrics, the corresponding level parameter (rssi or snr) must be provided, the request is invalid.

## RSSI Level Definitions

Used when metric = rssi

| Value | RSSI Range (dBm)         |
|-------|--------------------------|
| 0     | $\leq -140$              |
| 1     | $> -140$ and $\leq -120$ |
| 2     | $> -120$ and $\leq -100$ |
| 3     | $> -100$ and $\leq -80$  |
| 4     | $> -80$ and $\leq -60$   |
| 5     | $> -60$ and $\leq -40$   |
| 6     | $> -40$                  |

## SNR Level Definitions

Used when metric = snr

| Value | SNR Range (dB)         |
|-------|------------------------|
| 0     | $\leq -15$             |
| 1     | $> -15$ and $\leq -10$ |
| 2     | $> -10$ and $\leq -5$  |
| 3     | $> -5$ and $\leq 0$    |
| 4     | $> 0$ and $< 5$        |
| 5     | $> 5$ and $\leq 10$    |

| Value | SNR Range (dB) |
|-------|----------------|
| 6     | > 10 and ≤ 15  |
| 7     | > 15           |

### Response Example

```
{
  "result": [
    { "value": 120, "timestamp": "2025-10-09T10:00:00Z" },
    { "value": 98, "timestamp": "2025-10-09T10:10:00Z" }
  ],
  "nextToken": "abc123",
  "clientToken": "xyz456"
}
```

### Response Fields

| Field       | Type   | Description                                 |
|-------------|--------|---------------------------------------------|
| result      | array  | List of metric values and timestamps        |
| value       | number | Metric value at a specific time             |
| timestamp   | string | UTC timestamp of recorded data              |
| nextToken   | string | Token for the next page (if data continues) |
| clientToken | string | Identifier for tracking multi-page queries  |

## Get Gateway Battery Information

Retrieves the **solar battery information** for a specific gateway, including **temperature**, **voltage**, **state of charge (SOC)**, **state of health (SOH)**, and **charging mode**. Requires a valid API Key with **Read** permission.

### Endpoint

```
GET /v4/gateway/{sn}/battery
```

### Request Example

```
GET https://api.wisdm.network/v4/gateway/GW123456/battery
X-API-KEY: <Your_API_Key>
Accept-Encoding: gzip
```

### Response Example

```
{
  "sn": "918100QL6041021001",
  "battery": {
    "temperature": 26,
    "voltage": 13.6,
    "soc": 51,
    "current": -180,
    "soh": 100,
    "cycles": 5,
    "remainCapacity": 25520,
    "fullChargeCapacity": 50000,
    "mode": "DISCHARGING"
  }
}
```

### Response Fields

| Field   | Type   | Description           |
|---------|--------|-----------------------|
| sn      | string | Gateway serial number |
| battery | object | Battery information   |

### Battery Object Fields

| Field              | Type    | Description                                                   |
|--------------------|---------|---------------------------------------------------------------|
| temperature        | integer | Battery temperature (°C)                                      |
| voltage            | integer | Battery voltage (V)                                           |
| soc                | integer | State of Charge (percentage)                                  |
| current            | integer | current of the solar battery in mA                            |
| soh                | integer | State of Health (percentage)                                  |
| cycles             | integer | Cycle times of the solar battery                              |
| remainCapacity     | integer | Remaining capacity of the solar battery in mAh                |
| fullChargeCapacity | integer | Full charge capacity of the solar battery in mAh              |
| mode               | string  | Battery working mode (CHARGING / DISCHARGING / FULLY_CHARGED) |

### Enable Solar Battery Monitoring

Enables or disables the solar battery monitoring service for the gateway.

Requires a valid API Key with **WRITE\_ENABLE\_BATTERY\_MONITORING** permission.

### Endpoint

POST /v4/gateway/{sn}/battery

### Request Example

```
POST https://api.wisdmanetwork/v4/gateway/{sn}/battery
X-API-KEY: <Your_API_Key>
Accept-Encoding: gzip
Content-Type: application/json

{
  "enabled": true
}
```

### Response Example

```
204 No Content
```

### Request Fields

| Field   | Type    | Description                                  |
|---------|---------|----------------------------------------------|
| enabled | boolean | Enables or disables solar battery monitoring |

### Error Codes

| HTTP Code | Meaning               | Description                       |
|-----------|-----------------------|-----------------------------------|
| 400       | Bad Request           | Invalid parameters                |
| 403       | Forbidden             | API key lacks required permission |
| 404       | Not Found             | Gateway not found                 |
| 500       | Internal Server Error | System error                      |

## Send Asynchronous Command

Submits a remote command to a specific gateway for diagnostic or maintenance purposes. The request immediately returns a **commandId**, which can later be used to check the execution result through `GET /v4/gateway/{sn}/async-cmd/{commandId}`.

Currently, this API supports limited commands, such as netstat, used to inspect the network status of a gateway. Requires **Read and Write** permission for supported diagnostic commands.

### Endpoint

```
POST /v4/gateway/{sn}/async-cmd
```

## Request Example

```
POST https://api.wisdm.network/v4/gateway/GW123456/async-cmd
X-API-KEY: <Your_API_Key>
Accept-Encoding: gzip
Content-Type: application/json
{
  "command": "netstat",
  "args": "-apn",
  "pipes": [
    { "command": "grep", "args": "LISTEN" }
  ],
  "timeout": 60
}
```

## Request Parameters

| Field   | Type    | Description                                     |
|---------|---------|-------------------------------------------------|
| command | string  | Command to be executed (e.g., netstat)          |
| args    | string  | Command-line arguments                          |
| pipes   | array   | Optional chained commands                       |
| timeout | integer | Maximum execution time in seconds (default: 60) |

## Response Example

```
{ "commandId": 1760413840 }
```

## Error Codes

| HTTP Code | Meaning            | Description                              |
|-----------|--------------------|------------------------------------------|
| 400       | Invalid parameters | Command not recognized or invalid syntax |
| 404       | Resource Not Found | Gateway not found                        |
| 403       | Access denied      | API Key lacks permission                 |
| 500       | System error       | Gateway offline or execution error       |

## Query Asynchronous Command Result

This endpoint allows you to confirm whether the command has completed successfully and to view its console output.

When a gateway executes a queued command, the result is uploaded to WisDM. This endpoint queries the stored result and reports one of the following states:

- **PENDING:** Command has been queued or is waiting for execution.
- **SUCCESS:** Command executed successfully; result is available.
- **FAILED:** Execution failed or timed out.

Requires a valid API Key with **Read** permission.

## Endpoint

```
GET /v4/gateway/{sn}/async-cmd/{commandId}
```

## Request Example

```
GET https://api.wisdm.network/v4/gateway/GW123456/async-cmd/cmd-1760413840
X-API-KEY: <Your_API_Key>
Accept-Encoding: gzip
```

## Response Example

```
{
  "sn": "GW123456",
  "commandId": "1760413840",
  "result": "tcp 0 0 0.0.0.0:80 0.0.0.0:* LISTEN ..."
}
```

## Response Fields

| Field     | Type   | Description                         |
|-----------|--------|-------------------------------------|
| sn        | string | Gateway serial number               |
| commandId | string | Unique command identifier           |
| result    | string | Command output or execution message |

## Error Codes

| HTTP Code                 | Meaning            | Description                             |
|---------------------------|--------------------|-----------------------------------------|
| 400 Bad Request           | Invalid parameters | Incorrect or missing commandId          |
| 403 Forbidden             | Access denied      | API Key lacks read permission           |
| 404 Not Found             | Resource not found | CommandId does not exist or has expired |
| 500 Internal Server Error | System error       | Gateway offline or internal fault       |

## Reboot Gateway

Sends a **remote reboot command** to a specific gateway. Requires API Key with **WRITE\_REBOOT\_GATEWAY** permission.

## Endpoint

```
POST /v4/gateway/{sn}/reboot
```

## Request Example

```
POST https://api.wisdm.network/v4/gateway/GW123456/reboot
X-API-KEY: <Your_API_Key>
Accept-Encoding: gzip
```

## Response Example

204 No Content

## Error Codes

| HTTP Code | Meaning                 | Description                                      |
|-----------|-------------------------|--------------------------------------------------|
| 400       | Invalid parameters      | Missing or incorrect Serial Number               |
| 403       | Insufficient permission | API Key lacks write privileges                   |
| 404       | Resource Not Found      | Gateway not found                                |
| 500       | System failure          | Gateway is offline or unable to receive commands |

## Reset Gateway Password

Resets the **local root password** of a specific gateway. Requires API Key with **WRITE\_RESET\_PASSWORD** permission.

## Endpoint

POST /v4/gateway/{sn}/password

## Request Example

```
POST https://api.wisdm.network/v4/gateway/GW123456/password
X-API-KEY: <Your_API_Key>
Accept-Encoding: gzip
Content-Type: application/json
{
  "password": "StrongPass!2025"
}
```

## Request Parameters

| Field    | Type   | Description                             |
|----------|--------|-----------------------------------------|
| password | string | The new password to set for the gateway |

## Response Example

204 No Content

## Error Codes

| HTTP Code                 | Meaning                       | Description                                      |
|---------------------------|-------------------------------|--------------------------------------------------|
| 400 Bad Request           | Invalid or missing parameters | The password field was omitted or invalid        |
| 403 Forbidden             | Insufficient permission       | API Key lacks the WRITE_RESET_PASSWORD privilege |
| 500 Internal Server Error | System error                  | Gateway offline or internal configuration issue  |

## Upgrade Gateway Firmware

Upgrades the gateway firmware to the specified version.

Requires a valid API Key with **WRITE\_UPGRADE\_FIRMWARE** permission.

### Endpoint

```
POST /v4/gateway/{sn}/upgrade
```

### Request Example

```
POST https://api.wisdm.network/v4/gateway/{sn}/upgrade
X-API-KEY: <Your_API_Key>
Accept-Encoding: gzip
Content-Type: application/json

{
  "version": "2.2.13"
}
```

### Response Example

```
204 No Content
```

### Request Fields

| Field   | Type   | Description                                    |
|---------|--------|------------------------------------------------|
| version | string | Target firmware version (upgrade or downgrade) |

### Error Codes

| HTTP Code | Meaning               | Description                               |
|-----------|-----------------------|-------------------------------------------|
| 400       | Bad Request           | Invalid serial number or firmware version |
| 404       | Not Found             | Gateway not found                         |
| 403       | Forbidden             | API key lacks required permission         |
| 500       | Internal Server Error | System error                              |

## Get Network Priority

Retrieves the current **WAN interface priority configuration** for a specific gateway. Requires a valid API Key with **Read** permission.

### Endpoint

```
GET /v4/gateway/{sn}/network/priority
```

## Request Example

```
GET https://api.wisdm.network/v4/gateway/GW123456/network/priority
X-API-KEY: <Your_API_Key>
Accept-Encoding: gzip
```

## Response Example

```
{
  "sn": "GW123456",
  "priority": "ETH_CELLULAR_WIFI"
}
```

## Response Fields

| Field    | Type   | Description                                                                                |
|----------|--------|--------------------------------------------------------------------------------------------|
| sn       | string | Gateway serial number                                                                      |
| priority | string | Current interface order (e.g., <i>ETH_CELLULAR_WIFI</i> , <i>CELLULAR_WIFI_ETH</i> , etc.) |

## Error Codes

| HTTP Code | Meaning            | Description                                |
|-----------|--------------------|--------------------------------------------|
| 400       | Invalid parameter  | Incorrect or missing Serial Number         |
| 403       | Access denied      | API Key lacks read permissions             |
| 404       | Resource Not Found | Gateway not found                          |
| 500       | Internal failure   | Gateway not reachable or config file error |

## Set Network Priority

Sets the **WAN interface priority** order for a specific gateway. Requires API Key with **WRITE\_EDIT\_NETWORK\_INTERFACE** permission.

## Endpoint

```
POST /v4/gateway/{sn}/network/priority
```

## Request Example

```
POST https://api.wisdm.network/v4/gateway/GW123456/network/priority
X-API-KEY: <Your_API_Key>
Accept-Encoding: gzip
Content-Type: application/json
{
  "priority": "ETH_CELLULAR_WIFI"
}
```

## Request Parameters

| Field    | Type   | Description                                                                                                                              |
|----------|--------|------------------------------------------------------------------------------------------------------------------------------------------|
| priority | string | Defines interface order; supports combinations such as: ETH_CELLULAR_WIFI, ETH_WIFI_CELLULAR, CELLULAR_WIFI_ETH, WIFI_ETH_CELLULAR, etc. |

## Response Example

```
204 No Content
```

## Error Codes

| HTTP Code | Meaning                      | Description                                  |
|-----------|------------------------------|----------------------------------------------|
| 400       | Invalid or unsupported value | Priority field missing or incorrect syntax   |
| 404       | Resource Not Found           | Gateway not found                            |
| 403       | Insufficient permission      | API Key lacks modification rights            |
| 500       | Internal failure             | Gateway unreachable or config file corrupted |

## Get LAN Configuration

Retrieves the **current LAN configuration** of a specific gateway, including both **Ethernet** and **Wi-Fi LAN** interfaces.

Requires a valid API Key with **Read** permission.

## Endpoint

```
GET /v4/gateway/{sn}/network/lan
```

## Request Example

```
GET https://api.wisdm.network/v4/gateway/GW123456/network/lan
X-API-KEY: <Your_API_Key>
Accept-Encoding: gzip
```

## Response Example

```
{
  "sn": "918100M46020602001",
  "ethernet": {
    "enabled": false
  },
  "wifi": {
    "enabled": true,
    "ssid": "GW-LAN",
    "encryption": "WPA2-PSK",
    "key": "StrongKey123",
    "channel": "Auto",
    "hidden": false
  }
}
```

## Response Fields

| Field            | Type    | Description                                                      |
|------------------|---------|------------------------------------------------------------------|
| sn               | string  | Gateway serial number                                            |
| ethernet.enabled | boolean | Whether the Ethernet LAN interface is active                     |
| wifi.enabled     | boolean | Whether the Wi-Fi LAN interface is active                        |
| wifi.ssid        | string  | Wi-Fi network name (SSID)                                        |
| wifi.encryption  | string  | Encryption type (NONE / WPA-PSK / WPA2-PSK / WPA-WPA2-PSK-MIXED) |
| wifi.hidden      | boolean | Whether the SSID broadcast is hidden                             |
| wifi.channel     | string  | Current Wi-Fi channel (e.g., "Auto" or "2412MHz")                |

## Error Codes

| HTTP Code | Meaning            | Description                                 |
|-----------|--------------------|---------------------------------------------|
| 400       | Invalid parameter  | Missing or malformed Serial Number          |
| 403       | Access denied      | API Key lacks permission                    |
| 404       | Resource Not Found | Gateway not found                           |
| 500       | Internal failure   | Gateway unreachable or data retrieval error |

## Configure LAN Interface

Configures the **LAN interface parameters** of a specific gateway. Requires API Key with **WRITE\_EDIT\_NETWORK\_INTERFACE** permission.

### NOTE

Ethernet LAN configuration changes are restricted to prevent accidental disconnection from WisDM.

## Endpoint

```
POST /v4/gateway/{sn}/network/lan
```

## Request Example

```
POST https://api.wisdm.network/v4/gateway/GW123456/network/lan
X-API-KEY: <Your_API_Key>
Accept-Encoding: gzip
Content-Type: application/json
{
  "wifi": {
    "enabled": true,
    "ssid": "GW-LAN",
    "encryption": "WPA2-PSK",
```

```
"key": "StrongKey123",
"channel": "Auto",
"hidden": false  }
```

### Request Parameters

| Field           | Type    | Description                                                      |
|-----------------|---------|------------------------------------------------------------------|
| wifi.enabled    | boolean | Enables or disables Wi-Fi LAN interface                          |
| wifi.ssid       | string  | Wi-Fi network name (max 32 characters)                           |
| wifi.encryption | string  | Encryption mode (NONE / WPA-PSK / WPA2-PSK / WPA-WPA2-PSK-MIXED) |
| wifi.key        | string  | Wi-Fi password (8-63 characters, alphanumeric + special symbols) |
| wifi.channel    | string  | Wi-Fi channel selection (default: "Auto")                        |
| wifi.hidden     | boolean | Hides the SSID from broadcast if true                            |

### Response Example

204 No Content

### Error Codes

| HTTP Code | Meaning                 | Description                                      |
|-----------|-------------------------|--------------------------------------------------|
| 400       | Invalid request         | JSON syntax error or missing required field      |
| 403       | Insufficient permission | API Key lacks edit privileges                    |
| 404       | Resource Not Found      | Gateway not found                                |
| 500       | Internal error          | Gateway unreachable or failed configuration push |

### Get WAN Configuration

Retrieves the WAN interface settings of the gateway.  
Requires a valid API Key with Read permission.

### Endpoint

GET /v4/gateway/{sn}/network/wan

### Request Example

GET https://api.wisdm.network/v4/gateway/{sn}/network/wan

X-API-KEY: <Your\_API\_Key>

Accept-Encoding: gzip

### Response Example

```
{
  "ethernet": {
    "enabled": true,
    "protocol": "dhcp",
    "dhcpClient": { "useDNS": true },
    "tracking": {
      "trackMethod": "ping",
      "trackTargets": ["8.8.8.8"],
      "reliability": 1,
      "pingCount": 1,
      "pingInterval": 5,
      "pingTimeout": 2,
      "interfaceDown": 3,
      "interfaceUp": 3
    }
  },
  "wifi": {
    "enabled": false
  },
  "cellular": {
    "enabled": true,
    "apn": "internet",
    "tracking": {
      "trackMethod": "ping",
      "trackTargets": ["1.1.1.1"],
      "reliability": 1,
      "pingCount": 1,
      "pingInterval": 5,
      "pingTimeout": 2,
      "interfaceDown": 3,
      "interfaceUp": 3
    }
  }
}
```

## Response Fields

| Field    | Type   | Description                |
|----------|--------|----------------------------|
| ethernet | object | WAN Ethernet configuration |
| wifi     | object | WAN Wi-Fi configuration    |
| cellular | object | WAN Cellular configuration |

## Configure WAN Interfaces

Sets the WAN interface settings of the gateway (Ethernet / Wi-Fi / Cellular).  
Requires a valid API Key with **WRITE\_EDIT\_NETWORK\_INTERFACE** permission.

## Endpoint

POST /v4/gateway/{sn}/network/wan

## Request Example

POST https://api.wisdm.network/v4/gateway/{sn}/network/wan

X-API-KEY: <Your\_API\_Key>

Accept-Encoding: gzip

Content-Type: application/json

```
{
  "ethernet": {
    "enabled": true,
    "protocol": "dhcp",
    "dhcpClient": { "useDNS": true },
    "tracking": {
      "trackMethod": "ping",
      "trackTargets": ["8.8.8.8"],
      "reliability": 1,
      "pingCount": 1,
      "pingInterval": 5,
      "pingTimeout": 2,
      "interfaceDown": 3,
      "interfaceUp": 3
    }
  },
  "wifi": {
    "enabled": false
  },
  "cellular": {
    "enabled": false
  }
}
```

## Response Example

204 No Content

## Ethernet Configuration

| Field         | Type    | Description                      |
|---------------|---------|----------------------------------|
| enabled       | boolean | Enable or disable WAN Ethernet   |
| protocol      | string  | dhcp / static / ppoe             |
| staticAddress | object  | Required when protocol is static |
| dhcpClient    | object  | Required when protocol is dhcp   |
| ppoe          | object  | Required when protocol is ppoe   |
| tracking      | object  | Connectivity tracking settings   |

## Wi-Fi Configuration

| Field         | Type    | Description                      |
|---------------|---------|----------------------------------|
| enabled       | boolean | Enable or disable WAN Wi-Fi      |
| ssid          | string  | Wi-Fi SSID                       |
| encryption    | string  | Encryption type                  |
| key           | string  | Wi-Fi password                   |
| protocol      | string  | dhcp / static                    |
| staticAddress | object  | Required when protocol is static |
| dhcpClient    | object  | Required when protocol is dhcp   |
| tracking      | object  | Connectivity tracking settings   |

## Cellular Configuration

| Field    | Type    | Description                    |
|----------|---------|--------------------------------|
| enabled  | boolean | Enable or disable WAN Cellular |
| apn      | string  | APN name                       |
| username | string  | Optional APN username          |
| password | string  | Optional APN password          |
| pinCode  | string  | Optional SIM PIN               |
| tracking | object  | Connectivity tracking settings |

## Error Codes

| HTTP Code | Meaning     | Description                       |
|-----------|-------------|-----------------------------------|
| 400       | Bad Request | Invalid parameters                |
| 403       | Forbidden   | API key lacks required permission |

| HTTP Code | Meaning               | Description       |
|-----------|-----------------------|-------------------|
| 404       | Resource Not Found    | Gateway not found |
| 500       | Internal Server Error | System error      |

### Get Gateway Syslog Configuration

Returns the latest syslog configuration persisted for the specified gateway.

Requires a valid API Key with Read permission.

#### Endpoint

```
GET /v4/gateway/{sn}/syslog
```

#### Request Example

```
GET https://api.wisdm.network/v4/gateway/{sn}/syslog
X-API-KEY: <Your_API_Key>
Accept-Encoding: gzip
```

#### Response Example

```
{
  "sn": "918100RN6040814001",
  "shareLogs": "externalAndWisdm",
  "logsRotationCycle": 7,
  "wisdmLogsRetrievalCycle": 10,
  "wisdmLogsRotationCycle": 14,
  "externalLogServer": {
    "ip": "192.168.1.10",
    "port": 514
  }
}
```

#### Response Fields

| Field                   | Type    | Description                                 |
|-------------------------|---------|---------------------------------------------|
| sn                      | string  | Gateway serial number                       |
| shareLogs               | string  | none / external / wisdm / externalAndWisdm  |
| logsRotationCycle       | integer | Log rotation window (days)                  |
| wisdmLogsRetrievalCycle | integer | Interval in minutes for WisDM log retrieval |
| wisdmLogsRotationCycle  | integer | Number of days WisDM retains logs           |

| Field             | Type   | Description                          |
|-------------------|--------|--------------------------------------|
| externalLogServer | object | External syslog server configuration |

### External Log Server

| Field | Type    | Description              |
|-------|---------|--------------------------|
| ip    | string  | Syslog server IP address |
| port  | integer | Syslog server port       |

### Update Gateway Syslog Configuration

Updates syslog configuration for the specified gateway.

Requires a valid API Key with **WRITE\_EDIT\_SYSTEM\_SETTINGS** permission.

### Endpoint

```
POST /v4/gateway/{sn}/syslog
```

### Request Example

```
POST https://api.wisdm.network/v4/gateway/{sn}/syslog
X-API-KEY: <Your_API_Key>
Accept-Encoding: gzip
Content-Type: application/json

{
  "shareLogs": "external",
  "logsRotationCycle": 7,
  "wisdmLogsRetrievalCycle": 10,
  "wisdmLogsRotationCycle": 14,
  "externalLogServer": {
    "ip": "192.168.1.10",
    "port": 514
  }
}
```

### Response Example

```
204 No Content
```

## Request Fields

| Field                   | Type    | Description                                |
|-------------------------|---------|--------------------------------------------|
| shareLogs               | string  | none / external / wisdm / externalAndWisdM |
| logsRotationCycle       | integer | Log rotation window (days)                 |
| wisdMLogsRetrievalCycle | integer | WisDM log pull interval (minutes)          |
| wisdMLogsRotationCycle  | integer | WisDM log retention (days)                 |
| externalLogServer       | object  | Required when shareLogs includes external  |

## Get Gateway Syslog Download URLs

Returns gateway syslog **pre-signed download URLs** filtered by timestamps.

Requires a valid API Key with Read permission.

### Endpoint

```
GET /v4/gateway/{sn}/syslog/download
```

### Request Example

```
GET https://api.wisdM.network/v4/gateway/{sn}/syslog/download?startDate=2025-12-01T00:00:00Z&endDate=2025-12-08T00:00:00Z&maxFiles=20

X-API-KEY: <Your_API_Key>

Accept-Encoding: gzip
```

### Response Example

```
{
  "sn": "918100RN6040814001",
  "startDate": "2025-12-01T00:00:00Z",
  "endDate": "2025-12-08T00:00:00Z",
  "totalFiles": 2,
  "files": [
    {
      "key": "logs/918100RN6040814001/2025-12-07/syslog.txt.gz",
      "sizeBytes": 123456,
      "lastModified": "2025-12-07T23:10:00Z",
      "downloadUrl": "https://...presigned...",
      "expiresAt": "2025-12-08T01:10:00Z"
    }
  ]
}
```

## Request Parameters

| Parameter | Type    | Description                                           |
|-----------|---------|-------------------------------------------------------|
| startDate | string  | Start timestamp (UTC, ISO-8601)                       |
| endDate   | string  | End timestamp (UTC, ISO-8601)                         |
| maxFiles  | integer | Maximum number of files to return (1–200, default 20) |

## Response Fields

| Field      | Type    | Description                        |
|------------|---------|------------------------------------|
| sn         | string  | Gateway serial number              |
| startDate  | string  | Start timestamp used for filtering |
| endDate    | string  | End timestamp used for filtering   |
| totalFiles | integer | Total matching files               |
| files      | array   | List of log file objects           |

## File Object (files[])

| Field        | Type    | Description                 |
|--------------|---------|-----------------------------|
| key          | string  | Log file object key         |
| sizeBytes    | integer | File size in bytes          |
| lastModified | string  | Last modification timestamp |
| downloadUrl  | string  | Pre-signed download URL     |
| expiresAt    | string  | URL expiration timestamp    |

# Support

For questions regarding the WisDM API, use the WisDM Support Form in the WisDM platform or reach us at [support@rakwireless.com](mailto:support@rakwireless.com)

Document Version: 1.1.1  
 WisDM API Version: v4.1.0  
 Last Updated: December 2025  
 Contact: [support@rakwireless.com](mailto:support@rakwireless.com)