

## Overview

The API is a set of HTTP endpoints. Each endpoint is an HTTP GET requests or POST requests with JSON arguments and JSON responses.

The access port is same as that configured for Web Admin access. For security reason, however, the API should always be used under Secure HTTP (HTTPS) access.

## Getting Started

### API Resource URL

`https://<device_ip_address>/api/<function endpoint>`  
e.g. `https://192.168.1.1/api/status.wan.connection`

### Authentication - with Admin User account

As in Web Admin Access, Admin User account can access the API with the user name and password. After successfully login, the session will be authorized for subsequent access to the allowed APIs.

The session id is returned from cookie named "bauth" under Secure HTTP access.

### Authentication - with Client ID

API can be accessed with Client ID / secret, generated in advanced from the authenticated user, without disclosing the user name and password information

Successfully authorization with Client ID / secret with be granted with access token, which can be used along subsequent access to the allowed APIs

### Permission

- Read-Only Permission - It can only read the status and the config.
- Read-Write Permission - It can read the status and the config. It can also change the config.
- Admin Permission - It can manage the client and the token. It also have the "Read-Write Permission"

**Admin Permission** can only be granted by admin user account login

### Create Client

**Admin Permission** is needed to create the client

POST the name and scope by using the API call `/api/auth.client` endpoint

Example:

```
POST /api/auth.client HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json

{
    "name": "Client 1",
    "scope": "api.read-only"
}
```

Successful request will return client ID and client secret.

### Generate token

POST the client ID, client secret and scope(optional) by using the API call `/api/auth.token.grant`

Example:

```
POST /api/auth.token.grant HTTP/1.1
```

```
Host: 192.168.1.1
Content-Type: application/json
```

```
{
  "clientId": "9270c250111cabab02058007bb72217e",
  "clientSecret": "cf5fe1c51252a058ebd6bd7d5f493cf5"
}
```

Matched client ID and secret will return access token.

## How to use the access token

Add the access token as a GET parameter

Example:

```
GET /api/status.wan.connection?accessToken=43c65216eb16d779092fc40b184a1794 HTTP/1.1
Host: 192.168.1.1
```

Valid access token will get resource.

## HTTP Method

- **GET** to retrieves simple data
- **POST** to manipulate configuration or execute various actions, along with supplied arguments in JSON format

## GET Request Parameter

Parameters are passed in the query string (after the ? in the URL)

Example:

```
GET /api/status.wan.connection?id=1&lite=yes HTTP/1.1
Host: 192.168.1.1
```

## POST Request Parameter

Parameters in POST requests must be in JSON-encoded format

Example:

```
POST /api/login HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json
```

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

## Response

API response are in JSON-encoded format. The JSON response is an JSON object, with "stat" to indicate if the request is done successfully (ok) or not (fail)

Typically, a successfully response will have an "response" describe the retrieved information or result of the request

In failed responses, "code" is provided for the error code, and message about the failure, if any, will be described in "message"

	Type	Notation	Description
stat	String	{ok fail}	ok - API call success fail - API call not success
response	Any	-	Any additional information of the success call will be here
code	Number	<int>	Error code of the API call, only appear if the API call not success
message	String	<String>	Error message of the API call, only appear if the API call not success
notice	Object	<Object>	Extra information about this API request (but not part of the normal response). For example, the

### For success API call

```
{
  "stat": "ok"
}
```

Or

```
{
  "stat": "ok",
  "response": <Any JSON support type>
}
```

### For success API call (beta)

```
{
  "stat": "ok",
  "notice": {
    "status": "beta"
  },
  "response": <Any JSON support type>
}
```

### For fail API call

```
{
  "stat": "fail",
  "code": <int>,
  "message": <string>
}
```

# API Reference List

- POST login
- POST logout
- GET auth.client
- POST auth.client
- GET auth.client.token
- POST auth.token.grant
- POST auth.token.revoke
- POST cmd.billing.newCycle
- GET cmd.carrier.scan
- POST cmd.carrier.scan
- POST cmd.carrier.select
- POST cmd.channelPci.lock
- POST cmd.channelPci.scan
- POST cmd.config.apply
- POST cmd.config.discard
- POST cmd.port.poe.disable
- POST cmd.port.poe.enable
- POST cmd.sendUssd
- GET cmd.sms.get
- POST cmd.sms.sendMessage
- GET cmd.ap
- POST cmd.ap
- POST cmd.cellularModule.rescanNetwork
- POST cmd.cellularModule.reset
- GET cmd.wan.cellular
- POST cmd.wan.cellular
- POST cmd.wifi.connect
- POST cmd.wifi.disconnect
- POST cmd.wifi.forget
- GET cmd.wifi.result
- GET cmd.wifi.scan
- POST config.gpio
- GET config.ssid.profile
- POST config.ssid.profile
- POST config.wan.connection
- POST config.wan.connection.priority
- GET info.firmware
- GET info.location
- GET status.client
- GET status.lan.profile
- GET status.pevpn
- GET status.wan.connection
- GET status.wan.connection.allowance
- GET status.wan.connection.signal

# API Reference

## POST /api/login

### API

Acquire proper authorization for other API requests.

After a successful authentication, the obtained cookie session can be used for other API requests.

Permission GET is granted for Read-only user access, while Permission GET and POST are granted for Read-write user access.

The session is similar to that being used in Web Admin Access, and governed by the same session idle timeout.

For a more persistent API access, consider authorization with Client ID / Secret

*Available in 7.0.0 or later*

### Input Parameters

	Type	Notation	Mandatory	Description
<b>username</b>	String	<string>	require	Username
<b>password</b>	String	<string>	require	Password

### Return Parameters

#### Return JSON

	Type	Notation	Description
<b>permission</b>	Object	<Permission_Obj>	Permission granted. Most APIs require a proper permission to access.

#### <Permission\_Obj>

	Type	Notation	Description
<b>GET</b>	Number	{ 0, 1 }	1 - Allow retrieving data from the device 0 - Not allow retrieving data from the device
<b>POST</b>	Number	{ 0, 1 }	1 - Allow changing device settings 0 - Not allow changing device settings

### cURL Example

```
> curl -c cookies.txt -H "Content-Type: application/json" -X POST -d '{"username":"user","password":"pass"}' http://192.168.1.1/api/login
```

```
{
  "stat": "ok",
  "response": {
    "permission": {
      "GET": 1,
      "POST": 1
    }
  }
}
```

## POST /api/logout

### API

Properly logout the current session.

It is advised to logout immediately after use.

*Available in 7.0.0 or later*

## cURL Example

```
> curl -b cookies.txt -H "Content-Type: application/json" -X POST http://192.168.1.1/api/logout
```

```
{
  "stat": "ok"
}
```

## GET /api/auth.client

### Auth

Get the authentication client list. Only Admin Permission can access this information.

*Available in 7.1.1 or later*

### Return Parameters

#### Return JSON

	Type	Notation	Description
-	Array	list of <Client_Obj>	List of the auth client.

#### <Client\_Obj>

	Type	Notation	Description
<b>name</b>	String	<string>	Name of the client
<b>clientId</b>	String	<hash>	Client ID for granting the access token
<b>clientSecret</b>	String	<hash>	Client Secret for granting the access token
<b>confidential</b>	Boolean	<boolean>	Confidential or public client type
<b>createTimestamp</b>	Number	<number>	Create timestamp of the client
<b>scope</b>	String	{ api, api.read-only }	The scope of the client

## cURL Example

```
> curl -b cookies.txt http://192.168.1.1/api/auth.client
```

```
{
  "stat": "ok",
  "response": [
    {
      "name": "Client 1",
      "clientId": "9270c250111cabab02058007bb72217e",
      "clientSecret": "cf5fe1c51252a058ebd6bd7d5f493cf5",
      "confidential": false,
      "createTimestamp": 32172904,
      "scope": "api.read-only"
    }
  ]
}
```

## POST /api/auth.client

### Auth

### Create a new client

Create a new client by giving the name and scope. Only Admin Permission can access this information.

*Available in 7.1.1 or later*

## Input Parameters

	Type	Notation	Mandatory	Description
<b>action</b>	String	{ add }	require	
<b>name</b>	String	<string>	require	Client name
<b>scope</b>	String	{ api, api.read-only }	require	Scope of the client api - Read-Write permission api.read-only - Read-Only permission

## Return Parameters

### Return JSON

	Type	Notation	Description
<b>name</b>	String	<string>	Name of the client
<b>clientId</b>	String	<hash>	Client ID for granting the access token
<b>clientSecret</b>	String	<hash>	Client Secret for granting the access token
<b>confidential</b>	Boolean	<boolean>	Confidential or public client type
<b>createTimestamp</b>	Number	<number>	Create timestamp of the client
<b>scope</b>	String	{ api, api.read-only }	The scope of the client

## cURL Example

```
> curl -b cookies.txt -H "Content-Type: application/json" -X POST -d '{"action":"add","name":"Client 2","scope":"api"}' http://192.168.1.1/api/auth.client
```

```
{
  "stat": "ok",
  "response": {
    "name": "Client 2",
    "clientId": "0396c250111dcaef02058007bb72217e",
    "clientSecret": "de5cd1c51252a13854d6bd7ddeabbcf5",
    "confidential": false,
    "createTimestamp": 32175831,
    "scope": "api"
  }
}
```

## Remove a client

Remove the client by giving the client ID. Only Admin Permission can access this information.

*Available in 7.1.1 or later*

## Input Parameters

	Type	Notation	Mandatory	Description
<b>action</b>	String	{ remove }	require	
<b>clientId</b>	String	<hash>	require	Client ID

## cURL Example

```
> curl -b cookies.txt -H "Content-Type: application/json" -X POST -d '{"action":"remove","clientId":"0396c250111dcaef02058007bb72217e"}' http://192.168.1.1/api/auth.client
```

```
{
  "stat": "ok"
}
```

## GET /api/auth.client.token

## Auth

Obtain the access token list by providing the client ID  
Only Admin Permission can access this information.

Available in 7.1.1 or later

### Input Parameters

	Type	Notation	Mandatory	Description
<b>clientId</b>	String	<hash>	optional	Client ID. If this field is absent, all access token will be obtained.

### Return Parameters

#### Return JSON

	Type	Notation	Description
-	Array	<Access_Token_Obj>	List of access token information

#### <Access\_Token\_Obj>

	Type	Notation	Description
<b>accessToken</b>	String	<hash>	Access token
<b>clientId</b>	String	<hash>	Client ID
<b>clientName</b>	String	<string>	Client Name
<b>authorizationType</b>	Number	{ 3 }	Authorization type. Always get 3 for client credentials grant
<b>scope</b>	String	{ api, api.read-only }	The scope of the access token
<b>createTimestamp</b>	Number	<number>	Issued date in timestamp

### cURL Example

```
> curl -b cookies.txt http://192.168.1.1/api/auth.client.token?  
clientId=0396c250111dcaef02058007bb72217e
```

```
{  
  "stat": "ok",  
  "response": [  
    {  
      "accessToken": "43c65216eb16d779092fc40b184a1794",  
      "clientId": "0396c250111dcaef02058007bb72217e",  
      "clientName": "Client 1",  
      "authorizationType": 3,  
      "scope": "api.read-only",  
      "createTimestamp": 32177831  
    }  
  ]  
}
```

## POST /api/auth.token.grant

### Auth

Generate a new access token by giving the clientId and clientSecret.

Available in 7.1.1 or later

### Input Parameters

	Type	Notation	Mandatory	Description
<b>clientId</b>	String	<hash>	require	Client ID



<b>clientSecret</b>	String	<hash>	require	Client Secret
<b>scope</b>	String	{ api, api.read-only }	optional	Scope of the access token generated api - Read-write permission of API api.read-only - Read-only permission of API

## Return Parameters

### Return JSON

	Type	Notation	Description
<b>accessToken</b>	String	<hash>	Access token
<b>authorizationType</b>	Number	{ 3 }	Authorization type. Always out 3 for client credentials grant
<b>scope</b>	String	{ api, api.read-only }	The scope of the access token
<b>expiresIn</b>	Number	<number>	Expires in seconds

## cURL Example

```
> curl -b cookies.txt -H "Content-Type: application/json" -X POST -d
'{"clientId":"0396c250111dcaef02058007bb72217e","clientSecret":"de5cd1c51252a13854d6bd7ddeabbcf5","scope":"api"}' http://192.168.1.1/api/auth.token.grant
```

```
{
  "stat": "ok",
  "response": {
    "accessToken": "43c65216eb16d779092fc40b184a1794",
    "authorizationType": 3,
    "scope": "api",
    "expiresIn": 172800
  }
}
```

## POST /api/auth.token.revoke

### Auth

Revoke the access token provided.  
Only Admin Permission or self revoke can access this information.

*Available in 7.1.1 or later*

## Input Parameters

	Type	Notation	Mandatory	Description
<b>accessToken</b>	String	<hash>	require	Access token desired to revoke

## cURL Example

```
> curl -b cookies.txt -H "Content-Type: application/json" -X POST -d
'{"accessToken":"0396c250111dcaef02058007bb72217e"}' http://192.168.1.1/api/auth.token.revoke
```

```
{
  "stat": "ok"
}
```

## POST /api/cmd.billing.newCycle

### API

Start the new billing cycle by Connection ID and SIM ID

*Available in 8.1.0 or later*

## Input Parameters

	Type	Notation	Mandatory	Description
<b>connId</b>	Number	<conn_id>	require	WAN Connection ID to be renew billing cycle
<b>simId</b>	Number	[1,2]	optional	SIM ID to be renew billing cycle 1 is for SIM A, and 2 is for SIM B Always send 1 for single SIM model If the WAN Connection is not support cellular, the param will be ignored.

## cURL Example

```
> curl -b cookies.txt -H "Content-Type: application/json" -X POST -d '{"connId":4,"simId":1}'  
http://192.168.1.1/api/cmd.billing.newCycle
```

```
{  
  "stat": "ok"  
}
```

## GET /api/cmd.carrier.scan

### API

Obtain the result of discovered cellular network.  
The API will always return fail when the WAN connection is not support carrier scan.

Available in 8.0.1 or later

## Input Parameters

	Type	Notation	Mandatory	Description
<b>connId</b>	Number	<conn_id>	require	Specify which WAN connection ID is wanted to scan cellular network
<b>reference</b>	String	{ yes, no }	require	The cellular network which is wanted to select

## Return Parameters

### Return JSON

	Type	Notation	Description
<b>scanStatus</b>	String	{ scanning, done }	Report the scanning status
<b>timestamp</b>	Number	<integer>	Timestamp of the carrier list
<b>list</b>	Array	list of <Scan_Carrier_Obj>	List of discovered carrier
<b>reference</b>	Object	<Reference_Obj>	Current configuration

### <Scan\_Carrier\_Obj>

	Type	Notation	Description
<b>name</b>	String	<string>	Name of the carrier
<b>mobileType</b>	String	{ 2G, 3G, LTE }	-
<b>mcc</b>	String	3 digits <string>	Mobile Country Code
<b>mnc</b>	String	2-3 digits <string>	Mobile Network Code
<b>pcs</b>	Number	[ 0, 1 ]	-

### <Reference\_Obj>

	Type	Notation	Description
<b>activeSim</b>	Object NULL	<In_Use_SIM_Obj> NULL	Active SIM information. If there is no active SIM, this value is JSON NULL

## <In\_Use\_SIM\_Obj>

	Type	Notation	Description
<b>simId</b>	Number	{ 1, 2 }	SIM ID of the active SIM
<b>selectedCarrier</b>	Object NULL	<Carrier_Obj> NULL	The selected network If is it auto, this value is JSON NULL.

## <Carrier\_Obj>

	Type	Notation	Description
<b>name</b>	String	<string>	Name of the carrier
<b>mcc</b>	String	3 digits <string>	-
<b>mnc</b>	String	2-3 digits <string>	-
<b>pcs</b>	Number	[ 0, 1 ]	-

## cURL Example

```
> curl -b cookies.txt http://192.168.1.1/api/cmd.carrier.scan?connId=4&reference=yes
```

```
{
  "stat": "ok",
  "response": {
    "scanStatus": "scanning",
    "list": [
      {
        "name": ".csl",
        "mobileType": "LTE",
        "mcc": "454",
        "mnc": "0",
        "pcs": 0
      },
      {
        "name": "SMT HK",
        "mobileType": "LTE",
        "mcc": "454",
        "mnc": "6",
        "pcs": 0
      }
    ],
    "reference": {
      "activeSim": {
        "simId": 1,
        "cellularNetwork": null
      }
    }
  }
}
```

## POST /api/cmd.carrier.scan

### API

Obtain the result of discovered cellular network.  
The API will always return fail when the WAN connection is not support carrier scan.

*Available in 8.1.0 or later*

### Input Parameters

	Type	Notation	Mandatory	Description
<b>action</b>	String	{start}	optional	Trigger the scan start action

<b>connId</b>	Number	<conn_id>	require	Specify which WAN connection ID is wanted to scan cellular network
<b>reference</b>	String	{ yes, no }	optional	The cellular network which is wanted to select

## Return Parameters

### Return JSON

	Type	Notation	Description
<b>scanStatus</b>	String	{ scanning, done }	Report the scanning status
<b>timestamp</b>	Number	<integer>	Timestamp of the carrier list
<b>list</b>	Array	list of <Scan_Carrier_Obj>	List of discovered carrier
<b>reference</b>	Object	<Reference_Obj>	Current configuration

### <Scan\_Carrier\_Obj>

	Type	Notation	Description
<b>name</b>	String	<string>	Name of the carrier
<b>mobileType</b>	String	{ 2G, 3G, LTE }	-
<b>mcc</b>	String	3 digits <string>	Mobile Country Code
<b>mnc</b>	String	2-3 digits <string>	Mobile Network Code
<b>pcs</b>	Number	[ 0, 1 ]	-

### <Reference\_Obj>

	Type	Notation	Description
<b>activeSim</b>	Object NULL	<In_Use_SIM_Obj> NULL	Active SIM information. If there is no active SIM, this value is JSON NULL

### <In\_Use\_SIM\_Obj>

	Type	Notation	Description
<b>simId</b>	Number	{ 1, 2 }	SIM ID of the active SIM
<b>selectedCarrier</b>	Object NULL	<Carrier_Obj> NULL	The selected network If is it auto, this value is JSON NULL.

### <Carrier\_Obj>

	Type	Notation	Description
<b>name</b>	String	<string>	Name of the carrier
<b>mcc</b>	String	3 digits <string>	-
<b>mnc</b>	String	2-3 digits <string>	-
<b>pcs</b>	Number	[ 0, 1 ]	-

## cURL Example

```
> curl -b cookies.txt -H "Content-Type: application/json" -X POST -d '{"action":["start"],"connId":["4"],"reference":["yes']}' http://192.168.1.1/api/cmd.carrier.scan
```

```
{
  "stat": "ok",
  "response": {
    "scanStatus": "scanning",
    "list": [
      {
        "name": ".csl",
        "mobileType": "LTE",
        "mcc": "454",
        "mnc": "0",
```

```

        "pcs": 0
      },
      {
        "name": "SMT HK",
        "mobileType": "LTE",
        "mcc": "454",
        "mnc": "6",
        "pcs": 0
      }
    ],
    "reference": {
      "activeSim": {
        "simId": 1,
        "cellularNetwork": null
      }
    }
  }
}

```

## POST /api/cmd.carrier.select

### API

Update the cellular network selection

Available in 8.0.1 or later

### Input Parameters

	Type	Notation	Mandatory	Description
<b>connId</b>	Number	<conn_id>	require	Specify which WAN connection ID is wanted to change the carrier selection
<b>simId</b>	Number	{ 1, 2 }	optional	Specify which SIM is wanted to change the carrier selection
<b>selectedCarrier</b>	Object	<Carrier_Obj>	require	The carrier which is wanted to select

### <Carrier\_Obj>

	Type	Notation	Mandatory	Description
<b>mcc</b>	String	3 digits <string>	require	-
<b>mnc</b>	String	2-3 digits <string>	require	-
<b>pcs</b>	Number	[ 0, 1 ]	require	-
<b>name</b>	String	<String>	optional	-

### cURL Example

```
> curl -b cookies.txt -H "Content-Type: application/json" -X POST -d '{"connId":4,"selectedCarrier":{"mcc":"345","mnc":"23","pcs":0}}' http://192.168.1.1/api/cmd.carrier.select
```

```
{
  "stat": "ok"
}
```

## POST /api/cmd.channelPci.lock

### API Cellular

Lock the cellular connected LTE network on specific channel number with Physical Layer Cell Identity(PCI). PCI is optional

The API will always return fail when the WAN connection is not support channel PCI lock

Available in 8.1.1 or later

## Input Parameters

	Type	Notation	Mandatory	Description
<b>connId</b>	Number	<conn_id>	require	Specify which WAN connection ID is wanted to lock
<b>sim</b>	Array	list of <SIM_Obj>	require	Specify channel and PCI for the SIM card

### <SIM\_Obj>

	Type	Notation	Mandatory	Description
<b>id</b>	Number	<sim_id>	require	SIM ID 1 for SIM A 2 for SIM B
<b>value</b>	ObjectNull	<CH_PCI_Obj>	require	Specify channel PCI to lock Provide a JSON Null here to clear the lock for the SIM

### <CH\_PCI\_Obj>

	Type	Notation	Mandatory	Description
<b>channel</b>	Number	[0, 65535]	require	Specify channel to lock
<b>pci</b>	Number	[0, 65535]	optional	Specify PCI to lock

## cURL Example

```
> curl -b cookies.txt -H "Content-Type: application/json" -X POST -d '{"connId":4,"sim": [{"id":1,"value":{"channel":1350,"pci":77}}]}' http://192.168.1.1/api/cmd.channelPci.lock  
{  
  "stat": "ok"  
}
```

## POST /api/cmd.channelPci.scan

API Cellular

Obtain the result of discovered LTE cellular network.  
Provide action=start as parameter to rescan the channel PCI  
The API will always return fail when the WAN connection is not support channel PCI scan

Available in 8.1.1 or later

## Input Parameters

	Type	Notation	Mandatory	Description
<b>action</b>	String	{start}	optional	Trigger the scan start action
<b>connId</b>	Number	<conn_id>	require	Specify which WAN connection ID is wanted to scan channel PCI

## Return Parameters

### Return JSON

	Type	Notation	Description
<b>scanStatus</b>	String	{ scanning, done }	Report the scanning status
<b>timestamp</b>	Number	<integer>	Timestamp of the carrier list
<b>list</b>	Array	list of <CH_PCI_Obj>	List of discovered channel PCI

### <CH\_PCI\_Obj>

	Type	Notation	Description
--	------	----------	-------------

<b>pci</b>	Number	<integer>	Physical-layer Cell Identity
<b>earfcn</b>	Number	<integer>	E-UTRA Absolute radio-frequency channel number
<b>cellUtranId</b>	Number	<integer>	Cell UTRAN ID
<b>plmn</b>	Array	list of <PLMN_Obj>	Public land mobile network information

<PLMN\_Obj>

	Type	Notation	Description
<b>mcc</b>	String	3 digits <string>	Three decimal digits as Mobile Country Code(MCC)
<b>mnc</b>	String	2/3 digits <string>	Two or Three decimal digits as Mobile Network Code(MNC)

### cURL Example

```
> curl -b cookies.txt -H "Content-Type: application/json" -X POST -d '{"action":"start","connId":4}' http://192.168.1.1/api/cmd.channelPci.scan
```

```
{
  "stat": "ok",
  "response": {
    "scanStatus": "scanning",
    "timestamp": 1577836800,
    "list": [
      {
        "pci": 371,
        "earfcn": 3000,
        "cellUtranId": 23574039,
        "plmn": [
          {
            "mcc": "454",
            "mnc": "00"
          }
        ]
      }
    ]
  }
}
```

## POST /api/cmd.config.apply

API internal testing

### Apply changes

Apply the changes on pending config

*Available in 7.1.1 or later*

### Return Parameters

Return JSON

	Type	Notation	Description
<b>warning</b>	String	<string>	Changes are applied with a warning message. If there is no warning message, this field will not appear

### cURL Example

```
> curl -b cookies.txt -H "Content-Type: application/json" -X POST http://192.168.1.1/api/cmd.config.apply
```

```
{
  "stat": "ok"
}
```

## POST /api/cmd.config.discard

API internal testing

### Discard changes

Discard changes of pending config

Available in 7.1.1 or later

#### cURL Example

```
> curl -b cookies.txt -H "Content-Type: application/json" -X POST
http://192.168.1.1/api/cmd.config.discard
```

```
{
  "stat": "ok"
}
```

## POST /api/cmd.port.poe.disable

API

Disable the PoE of the port.

For Switch and Balance/MAX device:

Only Port ID is needed. To success turn off the PoE, the port must be enabled.

In Balance or MAX device, the port must be LAN port.

For modular devices like the EPX

If the device only has a single module or a fixed module, only the Port ID is needed.

If the device has more than one module, the port id, moduleType, and moduleId are all required.

To successfully turn off the PoE, the port must be enabled as a WAN or a LAN.

When the device does not support PoE or the port does not support PoE, then the API will return as fail.

Available in 8.1.1 or later

#### Input Parameters

	Type	Notation	Mandatory	Description
<b>port</b>	Number Object Array	<integer> <Port_Obj> list of {<integer>, <Port_Obj>}	require	This field support a single port or multiple port. User can port provide a single port ID, or array of port ID. Provide a <Port_Obj>, or array of <Port_Obj>

#### <Port\_Obj>

	Type	Notation	Mandatory	Description
<b>id</b>	Number	<integer>	require	Port ID
<b>moduleType</b>	String	<string>	optional	Module Type of the slot NOTE: This parameter is mandatory for modularized device.
<b>moduleId</b>	Number	<integer>	optional	Module ID of the slot NOTE: This parameter is mandatory for modularized device.

#### cURL Example

```
> curl -b cookies.txt -H "Content-Type: application/json" -X POST -d '{"port": [2,
{"id": 1, "moduleType": "E8", "moduleId": 2}]}' http://192.168.1.1/api/cmd.port.poe.disable
```

```
{
  "stat": "ok"
}
```



# POST /api/cmd.port.poe.enable

## API

Enable the PoE of the port.

For Switch and Balance/MAX device:

Only Port ID is needed. To success turn on the PoE, the port must be enabled.

In Balance or MAX device, the port must be LAN port.

For modular devices like the EPX

If the device only has a single module or a fixed module, only the Port ID is needed.

If the device has more than one module, the port id, moduleType, and moduleId are all required.

To successfully turn on the PoE, the port must be enabled as a WAN or a LAN.

When the device does not support PoE or the port does not support PoE, then the API will return as fail.

*Available in 8.1.1 or later*

## Input Parameters

	Type	Notation	Mandatory	Description
<b>port</b>	Number Object Array	<integer> <Port_Obj> list of {<integer>, <Port_Obj>}	require	This field support a single port or multiple port. User can port provide a single port ID, or array of port ID. Provide a <Port_Obj>, or array of <Port_Obj>

### <Port\_Obj>

	Type	Notation	Mandatory	Description
<b>id</b>	Number	<integer>	require	Port ID
<b>moduleType</b>	String	<string>	optional	Module Type of the slot NOTE: This parameter is mandatory for modularized device.
<b>moduleId</b>	Number	<integer>	optional	Module ID of the slot NOTE: This parameter is mandatory for modularized device.

## cURL Example

```
> curl -b cookies.txt -H "Content-Type: application/json" -X POST -d '{"port":[2, {"id":1,"moduleType":"E8","moduleId":2}]}' http://192.168.1.1/api/cmd.port.poe.enable
```

```
{
  "stat": "ok"
}
```

# POST /api/cmd.sendUssd

## API

Send USSD to target address, if there is any SIM card supported.

*Available in 8.1.0 or later*

## Input Parameters

	Type	Notation	Mandatory	Description
<b>connId</b>	Number	<conn_id>	require	Specify which WAN connection ID sends USSD
<b>simId</b>	Number	<sim_id>	optional	Specify which SIM ID sends USSD. If the information is absent, the call will choose the active SIM
<b>ussd</b>	String	{1234567890*#}	require	USSD code

## cURL Example

```
> curl -b cookies.txt -H "Content-Type: application/json" -X POST -d '{"connId":2,"ussd":"*109#"}'
```

http://192.168.1.1/api/cmd.sendUssd

```
{
  "stat": "ok",
  "response": {
    "message": "Request is sent successfully"
  }
}
```

## GET /api/cmd.sms.get

### API

Fetch the active SIM SMS according to connId.

*Available in 8.1.0 or later*

### Input Parameters

	Type	Notation	Mandatory	Description
connId	Number	<conn_id>	require	Get the SMS according to WAN connection ID

### Return Parameters

#### Return JSON

	Type	Notation	Description
connId	String	<hash>	Access token
simId	Number	{ 3 }	Authorization type. Always out 3 for client credentials grant
sms	Array	list of <SMS_Obj>	List of SMS message

#### <SMS\_Obj>

	Type	Notation	Description
sender	String	<string>	Sender of the SMS
message	Array	list of <Message_Obj>	The list of the message

#### <Message\_Obj>

	Type	Notation	Description
id	Number	<int>	The ID of the SMS
date	String	<string>	Date of the SMS
timestamp	Number	<timestamp>	Timestamp of the SMS
length	Number	<integer>	The length of the SMS message content
content	String	<string>	SMS content

### cURL Example

```
> curl -b cookies.txt http://192.168.1.1/api/cmd.sms.get?connId=6
```

```
{
  "stat": "ok",
  "response": {
    "connId": 6,
    "simId": 1,
    "sms": [
      {
        "sender": "988",
        "message": [
          {
```

```

        "id": 1,
        "date": "Feb 17 13:55",
        "timestamp": 1581774925,
        "length": "50",
        "message": "The is the 1st line SMS,\nand this is the 2nd line."
      }
    ],
  },
  {
    "sender": "+81325359875",
    "message": [
      {
        "id": 2,
        "date": "Feb 05 01:55",
        "timestamp": 1580867113,
        "length": "24",
        "message": "Multipart message part 1"
      },
      {
        "id": 6,
        "date": "Feb 05 01:55",
        "timestamp": 1580867113,
        "length": "24",
        "message": "Multipart message part 2"
      }
    ]
  }
]
}

```

## POST /api/cmd.sms.sendMessage

### API

Send SMS message to target address, if there is any SIM card supported.

*Available in 8.0.0 or later*

### Input Parameters

	Type	Notation	Mandatory	Description
<b>connId</b>	Number	<conn_id>	optional	Specify which WAN connection ID sends the SMS message
<b>address</b>	String	<string>	require	Target address of the SMS message, the address must begin with '+' and follow with 2 to 15 digits. and the first digit cannot be '0'
<b>content</b>	String	<string>	optional	Content of the SMS message

### cURL Example

```
> curl -b cookies.txt -H "Content-Type: application/json" -X POST -d
'{"address":"+85235984335","content":"SMS Content"}' http://192.168.1.1/api/cmd.sms.sendMessage
```

```
{
  "stat": "ok"
}
```

## GET /api/cmd.ap

### API alpha

Returns the status of the device Access Point

Available in 7.0.2 or later

## Return Parameters

### Return JSON

	Type	Notation	Description
<b>support</b>	Boolean	<boolean>	Indicates the support of Access Point. Products without Access Point will return false, and provides no further information.
<b>enable</b>	Boolean	<boolean>	Indicates if Access point is currently turned on
<b>wanDependent</b>	Boolean	<boolean>	[Experimental] Returns true when the engineering setting "Turn off AP when there is n Internet connectivity" is currently enabled. (This value is not officially supported and is subject to change in future

## cURL Example

```
> curl -b cookies.txt http://192.168.1.1/api/cmd.ap
```

```
{
  "stat": "ok",
  "response": {
    "support": true,
    "enable": true,
    "wanDependent": true
  }
}
```

## POST /api/cmd.ap

API alpha

Switch on or shut down the device Access Point.

Available in 7.0.2 or later

## Input Parameters

	Type	Notation	Mandatory	Description
<b>enable</b>	Boolean	<boolean>	require	true to Switch on the device Access Point; otherwise to turn off the Access Point.

## Return Parameters

### Return JSON

	Type	Notation	Description
<b>support</b>	Boolean	<boolean>	Indicates the support of Access Point. Products without Access Point will return false, and provides no further information.
<b>enable</b>	Boolean	<boolean>	Indicates if Access point is currently turned on
<b>wanDependent</b>	Boolean	<boolean>	[Experimental] Returns true when the engineering setting "Turn off AP when there is n Internet connectivity" is currently enabled. (This value is not officially supported and is subject to change in future

## cURL Example

```
> curl -b cookies.txt -H "Content-Type: application/json" -X POST -d '{"enable":true}'
http://192.168.1.1/api/cmd.ap
```

```
{
```

```
"stat": "ok",
"response": {
  "support": true,
  "enable": true,
  "wanDependent": true
}
}
```

## POST /api/cmd.cellularModule.rescanNetwork

### API

Rescan the network of the corresponding WAN connection

*Available in 8.0.0 or later*

### Input Parameters

	Type	Notation	Mandatory	Description
connId	Number	<int>	require	WAN connection ID of the cellular module to rescan

### cURL Example

```
> curl -b cookies.txt -H "Content-Type: application/json" -X POST -d '{"connId":"4"}'
http://192.168.1.1/api/cmd.cellularModule.rescanNetwork
```

```
{
  "stat": "ok"
}
```

## POST /api/cmd.cellularModule.reset

### API

Reset the cellular module of the corresponding WAN connection

*Available in 8.0.0 or later*

### Input Parameters

	Type	Notation	Mandatory	Description
connId	Number	<int>	require	WAN connection ID of the cellular module to reset

### cURL Example

```
> curl -b cookies.txt -H "Content-Type: application/json" -X POST -d '{"connId":"4"}'
http://192.168.1.1/api/cmd.cellularModule.reset
```

```
{
  "stat": "ok"
}
```

## GET /api/cmd.wan.cellular

### API

Obtain the current enabled SIM and preferred SIM settings

*Available in 8.0.0 or later*

### Input Parameters

	Type	Notation	Mandatory	Description
<b>connId</b>	Number	<int>	require	WAN Connection of the cellular module

## Return Parameters

### Return JSON

	Type	Notation	Description
<b>enabledSim</b>	Array	list of {1, 2}	SIMs to be used (1 for SIM Slot A and 2 for SIM Slot B)
<b>preferredSim</b>	Number	{1, 2, null}	Preferred SIM Slot, null to indicate no preference (Only applicable when multiple SIMs are being used)

## cURL Example

```
> curl -b cookies.txt http://192.168.1.1/api/cmd.wan.cellular?connId=4
```

```
{
  "stat": "ok",
  "response": {
    "enabledSim": [
      1,
      2
    ],
    "preferredSim": 1
  }
}
```

## POST /api/cmd.wan.cellular

### API

Change the enabled SIM and preferred SIM

*Available in 8.0.0 or later*

## Input Parameters

	Type	Notation	Mandatory	Description
<b>connId</b>	Number	<int>	require	WAN Connection of the cellular module
<b>enabledSim</b>	Array	list of {1, 2}	optional	Choice of SIMs to be used (1 for SIM Slot A and 2 for SIM Slot B)
<b>preferredSim</b>	Number	{1, 2, null}	optional	Preferred SIM Slot, null to indicate no preference (Only applicable when multiple SIMs are being used)

## cURL Example

```
> curl -b cookies.txt -H "Content-Type: application/json" -X POST -d '{"connId":4,"enabledSim": [1,2],"preferredSim":1}' http://192.168.1.1/api/cmd.wan.cellular
```

```
{
  "stat": "ok"
}
```

## POST /api/cmd.wifi.connect

### API

Connect the Wi-Fi with provide SSID if profile is defined.

If the SSID profile is not defined, connection will require additional information.

WEP or WPA-PSK connection require 'key'

WPA-EAP and 802.1x connection require the Extensible Authentication Protocol(EAP) related information.

When credential cannot be obtained from existing SSID profile, nor supplied parameters, connection cannot be done.

Available in 7.1.1 or later

## Input Parameters

	Type	Notation	Mandatory	Description
<b>connId</b>	Number	<int>	require	Wi-Fi with the WAN connection ID to be used.
<b>ssid</b>	String	<string>	require	SSID to be connected
<b>securityPolicy</b>	String	{ open, wep, wpa-eap, wpa-psk, 8021x }	require	Security Policy to connect the SSID
<b>key</b>	String	<string>	optional	Key for WEP and WAP-PSK security policy
<b>loginId</b>	String	<string>	optional	Login ID for Extensible Authentication Protocol(EAP)
<b>password</b>	String	<string>	optional	Password for Extensible Authentication Protocol(EAP)
<b>eapMethod</b>	String	{ TTLS, PEAP }	optional	Extensible Authentication Protocol(EAP) Method
<b>eapPhase2</b>	String	{ CHAP, MSCHAP, MSCHAPV2, PAP }	optional	Extensible Authentication Protocol(EAP) Phase 2 Method
<b>eapAuthenticationId</b>	String	{ anonymous, credentials } <string>	optional	Extensible Authentication Protocol(EAP) outer authentication identit
<b>eapAuthenticationId</b>	String	{ anonymous, credentials } <string>	optional	Extensible Authentication Protocol(EAP) outer authentication identit
<b>preferredBssid</b>	String	<mac>	optional	Preferred BSSID of the Wi-Fi connection

## cURL Example

```
> curl -b cookies.txt -H "Content-Type: application/json" -X POST -d '{"connId":1,"ssid":"Main SSID"}' http://192.168.1.1/api/cmd.wifi.connect
```

```
{  
  "stat": "ok"  
}
```

## POST /api/cmd.wifi.disconnect

### API

Disconnect the Wi-Fi if it is connected

Available in 7.1.1 or later

## Input Parameters

	Type	Notation	Mandatory	Description
<b>connId</b>	Number	<int>	require	Wi-Fi with the WAN connection ID to be used.
<b>ssid</b>	String	<string>	optional	SSID to be disconnected. When omitted, the current connected SSID will be disconnected.

## cURL Example

```
> curl -b cookies.txt -H "Content-Type: application/json" -X POST -d '{"connId":1,"ssid":"Main SSID"}' http://192.168.1.1/api/cmd.wifi.disconnect
```

```
{  
  "stat": "ok"  
}
```

## POST /api/cmd.wifi.forget

## API

Remove existing SSID profile, if any, by giving the SSID and Authentication method. Wi-Fi will also disconnect if it is using this SSID.

Available in 7.1.1 or later

### Input Parameters

	Type	Notation	Mandatory	Description
<b>connId</b>	Number	<int>	require	Wi-Fi with the WAN connection ID to be used.
<b>ssid</b>	String	<string>	require	SSID to be forgotten
<b>securityPolicy</b>	String	{ open, wep, wpa-eap, wpa-psk, 8021x }	require	Security Policy of the SSID

### cURL Example

```
> curl -b cookies.txt -H "Content-Type: application/json" -X POST -d '{"connId":1,"ssid":"Main SSID","securityPolicy":"wpa-psk"}' http://192.168.1.1/api/cmd.wifi.forget
```

```
{  
  "stat": "ok"  
}
```

## GET /api/cmd.wifi.result

### API

Obtain the last known result of Wi-Fi WAN Connection

Available in 7.1.1 or later

### Input Parameters

	Type	Notation	Mandatory	Description
<b>connId</b>	Number	<int>	require	Wi-Fi with the WAN connection ID to be used.

### Return Parameters

#### Return JSON

	Type	Notation	Description
<b>timestamp</b>	Number	<timestamp>	Timestamp of the last know result
<b>result</b>	String	{ CONNECTED, TIMEOUT, PSK_AUTH_FAIL, EAP_AUTH_FAIL, AP_NOT_FOUND, UNKNOWN_FAIL }	CONNECTED - Wi-Fi is success connected TIMEOUT - Wi-Fi connect timeout AP_NOT_FOUND - Cannot found the AP PSK_AUTH_FAIL - Wi-Fi connect fail and the reason is PSK not match EAP_AUTH_FAIL - Wi-Fi connect fail and the reason is username and password of EAP not match UNKNOWN_FAIL - Wi-Fi connect fail but the error cannot be classified
<b>bssid</b>	String	<mac>	BSSID of the connected AP
<b>ssid</b>	String	<string>	SSID of the connected AP
<b>securityPolicy</b>	String	{ open, wep, wpa-eap, wpa-psk, 8021x }	Security Policy of the connected AP
<b>message</b>	String	<string>	Additional information of the status

### cURL Example

```
> curl -b cookies.txt http://192.168.1.1/api/cmd.wifi.result?connId=1
```

```
{
```



```

"stat": "ok",
"response": {
  "result": "CONNECTED",
  "timestamp": 1529899328,
  "ssid": "Main SSID",
  "bssid": "A2:E5:B8:55:89:DF",
  "securityPolicy": "wpa-psk",
  "message": "connected to Main SSID (A2:E5:B8:55:89:DF)"
}
}

```

## GET /api/cmd.wifi.scan

### API

Discover nearby Wi-Fi access points

Available in 7.1.1 or later

### Input Parameters

	Type	Notation	Mandatory	Description
<b>connId</b>	Number	<int>	require	Wi-Fi with the WAN connection ID to be used.
<b>infoType</b>	String	{ status, config }	optional	Additional information can be requested along with discovered Wi-Fi access point. config - indicates if the connect profile is present status - indicates if the SSID is connected, or with connect profile
<b>sortBy</b>	String	{ name, security, signal, channel }	optional	Sort by name, security method, signal or channel. When omitted, it will sort by name and the defined SSID will be on the head of the array
<b>sortOrder</b>	String	{ asc, desc }	optional	Sort with descending or ascending order

### Return Parameters

#### Return JSON

	Type	Notation	Description
-	Array	list of <Wifi_Obj>	List of discovered Wi-Fi Access Points

#### <Wifi\_Obj>

	Type	Notation	Description
<b>ssid</b>	String	<string>	Service Set Identifier (SSID)
<b>bssid</b>	String	<mac>	Basic Service Set Identifier (BSSID)
<b>signal</b>	Number	<Number>	Signal in dBm Deprecated in firmware 8.1.0
<b>signalStrength</b>	Number	<Number>	Signal in dBm Introduced in firmware 8.1.0
<b>signalLevel</b>	Number	[0, 5]	Signal level Introduced in firmware 8.1.0
<b>channel</b>	Number	<Number>	Channel
<b>securityPolicy</b>	String	{ open, wep, wpa-eap, wpa-psk, 8021x }	Security Policy
<b>status</b>	Object	<Status_Obj>	Status information
<b>config</b>	Object	<Config_Obj>	Config information

#### <Status\_Obj>

	Type	Notation	Description
<b>inUse</b>	Boolean	<boolean>	SSID profile is targeted as connection.

<b>connected</b>	Boolean	<boolean>	Wi-Fi is currently connected to this SSID.
------------------	---------	-----------	--

<Config\_Obj>

	Type	Notation	Description
<b>profileId</b>	Number	<integer>	ID of the connect profile for this SSID.
<b>automatic</b>	Boolean	<boolean>	Indicates if Wi-Fi is configured to connect this SSID automatically.

### cURL Example

```
> curl -b cookies.txt http://192.168.1.1/api/cmd.wifi.scan?connId=1&infoType=status
```

```
{
  "stat": "ok",
  "response": [
    {
      "ssid": "Main SSID",
      "bssid": "A2:E5:B8:55:89:DF",
      "signal": -68,
      "channel": 10,
      "securityPolicy": "wpa-psk",
      "status": {
        "inUse": true,
        "connected": true
      }
    }
  ]
}
```

## POST /api/config.gpio

API beta

Obtain and updated the GPIO  
The API will return the updated config as return.

If the passing a empty 'list', it will return the current config,  
no update will be made

*Available in 8.1.1 or later*

### Input Parameters

	Type	Notation	Mandatory	Description
<b>list</b>	Array	list of <GPIO_Obj>	optional	List of GPIO config for updating
<b>reference</b>	Boolean	<boolean>	optional	GPIO reference or not

### <GPIO\_Obj>

	Type	Notation	Mandatory	Description
<b>id</b>	Number	<integer>	require	
<b>enable</b>	Boolean	<boolean>	optional	GPIO enable
<b>type</b>	String	{ digital_input, digital_output, analog_input }	optional	GPIO type
<b>mode</b>	String	{ input_sensing, ignition_sensing } { wan_status } { input_sensing, voltage_measurement, analog_testing }	optional	For type=digital_input, { input_sensing, ignition_sensing } For type=digital_output, { wan_status } For type=analog_input, { input_sensing, voltage_measurement, analog_testing }

<b>delay</b>	Number	[ 1, 3600 ]	optional	GPIO delay ONLY for input type
--------------	--------	-------------	----------	-----------------------------------

## Return Parameters

### Return JSON

	Type	Notation	Description
<b>&lt;gpio_id&gt;</b>	Object	<GPIO_Obj>	GPIO information for the <gpio_id>
<b>order</b>	Array	list of <gpio_id>	The order of the ids
<b>reference</b>	Object	<GPIO_Ref_Map_Obj>	Provide the support type and mode for each <gpio_id>

### <GPIO\_Obj>

	Type	Notation	Description
<b>enable</b>	Boolean	<boolean>	GPIO enable
<b>type</b>	String	{ digital_input, digital_output, analog_input }	GPIO type
<b>mode</b>	String	{ input_sensing, ignition_sensing } { wan_status } { input_sensing, voltage_measurement, analog_testing }	For type=digital_input, { input_sensing, ignition_sensing } For type=digital_output, { wan_status } For type=analog_input, { input_sensing, voltage_measurement, analog_testing }
<b>delay</b>	Number	[ 1, 3600 ]	GPIO delay ONLY for input type

### <GPIO\_Ref\_Map\_Obj>

	Type	Notation	Description
<b>&lt;gpio_id&gt;</b>	Object	<GPIO_Ref_Obj>	GPIO reference for the <gpio_id>
<b>order</b>	Array	list of <gpio_id>	The order of the ids

### <GPIO\_Ref\_Obj>

	Type	Notation	Description
<b>name</b>	String	<string>	GPIO name
<b>type</b>	Array	list of { digital_input, digital_output, analog_input }	Which GPIO type support for the <gpio_id>
<b>mode</b>	Object	<GPIO_Ref_Mode_Obj>	Which GPIO mode is support for specific GPIO type

### <GPIO\_Ref\_Mode\_Obj>

	Type	Notation	Description
<b>digital_input</b>	Array	list of { input_sensing, ignition_sensing }	Support mode for digital_input type
<b>digital_output</b>	Array	list of { wan_status }	Support mode for digital_output type
<b>analog_input</b>	Array	list of { input_sensing, voltage_measurement, analog_testing }	Support mode for analog_input type

## cURL Example

```
> curl -b cookies.txt -H "Content-Type: application/json" -X POST -d '{"list":
[{"id":1,"enable":true,"type":"digital_output","mode":"toggle_high"},
{"id":2,"enable":true,"type":"digital_input","mode":"input_sensing","delay":3}]}'
http://192.168.1.1/api/config.gpio
{
  "stat": "ok",
```

```

"response": {
  "1": {
    "enable": true,
    "type": "digital_output",
    "mode": "toggle_high"
  },
  "2": {
    "enable": true,
    "type": "digital_input",
    "mode": "input_sensing",
    "delay": 3
  },
  "order": [
    1,
    2
  ]
}

```

## GET /api/config.ssid.profile

### API

Obtain the SSID profile information

Available in 7.1.1 or later

### Input Parameters

	Type	Notation	Mandatory	Description
<b>id</b>	Array	<numlist>	optional	list the SSID Profile base on id, multiple value is accepted, When omitted, all profile will be return.

### Return Parameters

#### Return JSON

	Type	Notation	Description
<b>order</b>	Array	list of <profile_id>	The order of the SSID Profile ID
<b>&lt;profile_id&gt;</b>	Object	<SSID_Profile_Obj>	SSID Profile information

#### <SSID\_Profile\_Obj>

	Type	Notation	Description
<b>name</b>	String	<string>	SSID of the profile
<b>enable</b>	Boolean	<boolean>	Profile enabled or not
<b>vlanId</b>	Number	<integer>	VLAN ID of the profile, the field will not appear if use the LAN
<b>capturePortal</b>	Boolean	<boolean>	Profile will use captive portal or not
<b>incontrolManaged</b>	Boolean	<boolean>	InControl is managed this profile or not
<b>broadcast</b>	Boolean	<boolean>	Broadcast the SSID or not
<b>security</b>	Object	<SSID_Security_Obj>	The security policy and related information

#### <SSID\_Security\_Obj>

	Type	Notation	Description
<b>policy</b>	String	{ WPA2 Personal, WPA/WPA2 Personal }	Security policy of the SSID profile
<b>wpa2Personal</b>	Object	<WPA2_Personal_Obj>	WPA2 Personal related information
<b>wpaWpa2Personal</b>	Object	<WPA2_Personal_Obj>	WPA/WPA2 Personal related information

## <WPA2\_Personal\_Obj>

	Type	Notation	Description
<b>fastTransition</b>	Boolean	<boolean>	Fast Transition for WPA2, this field will not appear in WPA/WPA2 Personal This config does not take effect in 7.1.1 with WPA2 Enterprise
<b>key</b>	String	<string>	Key for WPA2 Personal and WPA/WPA2 Personal

### cURL Example

```
> curl -b cookies.txt http://192.168.1.1/api/config.ssid.profile?id=1 2
```

```
{
  "stat": "ok",
  "response": {
    "1": {
      "name": "Main SSID",
      "enable": true,
      "captivePortal": true,
      "incontrolManaged": false,
      "broadcast": true,
      "security": {
        "policy": "WPA2 Personal",
        "wpa2Personal": {
          "fastTransition": true,
          "key": "pas53or2"
        }
      }
    },
    "2": {
      "name": "Guest SSID",
      "enable": true,
      "captivePortal": true,
      "incontrolManaged": false,
      "broadcast": true,
      "vlanId": 1,
      "security": {
        "policy": "WPA2 Personal",
        "wpa2Personal": {
          "fastTransition": false,
          "key": "pass3ord"
        }
      }
    },
    "order": [
      1,
      2
    ]
  }
}
```

## POST /api/config.ssid.profile

API

### Update the SSID profile

Update the SSID profile according to the given information.  
Only given information will be affect.

*Available in 7.1.1 or later*

### Input Parameters

	Type	Notation	Mandatory	Description
<b>action</b>	String	{ update }	require	State the update action
<b>id</b>	Number	<number>	require	Profile ID which is wanted to update
<b>name</b>	String	<string>	optional	SSID of the profile
<b>enable</b>	Boolean	<boolean>	optional	Enable the profile or not
<b>vlanId</b>	Number	<integer>	optional	VLAN ID of the profile, the field will not appear if use the LAN
<b>broadcast</b>	Boolean	<boolean>	optional	Broadcast the profile or not
<b>security</b>	Object	<SSID_Security_Obj>	optional	Security information

### <SSID\_Security\_Obj>

	Type	Notation	Mandatory	Description
<b>policy</b>	String	{ "WPA2 Personal", "WPA/WPA2 Personal" }	optional	Security Policy of the SSID profile
<b>wpa2Personal</b>	Object	<WPA2_Personal_Obj>	optional	WPA2 Personal related information
<b>wpaWpa2Personal</b>	Object	<WPA2_Personal_Obj>	optional	WPA/WPA2 Personal related information

### <WPA2\_Personal\_Obj>

	Type	Notation	Mandatory	Description
<b>fastTransition</b>	Boolean	<boolean>	optional	Fast Transition for WPA2, this field cannot be set in WPA/WPA2 Personal This config does not take effect in 7.1.1 with WPA2 Enterprise
<b>key</b>	String	<string>	optional	Key for WPA2 Personal or WPA/WPA2 Personal The length must between 8 and 63 or HEX in 64

## Return Parameters

### Return JSON

	Type	Notation	Description
<b>order</b>	Array	list of <profile_id>	The order of the SSID Profile ID
<b>&lt;profile_id&gt;</b>	Object	<SSID_Profile_Obj>	SSID Profile information

### <SSID\_Profile\_Obj>

	Type	Notation	Description
<b>name</b>	String	<string>	SSID of the profile
<b>enable</b>	Boolean	<boolean>	Profile enabled or not
<b>vlanId</b>	Number	<integer>	VLAN ID of the profile, the field will not appear if use the LAN
<b>captivePortal</b>	Boolean	<boolean>	Profile will use captive portal or not
<b>incontrolManaged</b>	Boolean	<boolean>	InControl is managed this profile or not
<b>broadcast</b>	Boolean	<boolean>	Broadcast the SSID or not
<b>security</b>	Object	<SSID_Security_Obj>	The security policy and related information

### <SSID\_Security\_Obj>

	Type	Notation	Description
<b>policy</b>	String	{ "WPA2 Personal", "WPA/WPA2 Personal" }	Security policy of the SSID profile
<b>wpa2Personal</b>	Object	<WPA2_Personal_Obj>	WPA2 Personal related information
<b>wpaWpa2Personal</b>	Object	<WPA2_Personal_Obj>	WPA/WPA2 Personal related information

### <WPA2\_Personal\_Obj>

	Type	Notation	Description
<b>fastTransition</b>	Boolean	<boolean>	Fast Transition for WPA2, this field will not appear in WPA/WPA2 Personal This config does not take effect in 7.1.1 with WPA2 Enterprise
<b>key</b>	String	<string>	Key for WPA2 Personal and WPA/WPA2 Personal

## cURL Example

```
> curl -b cookies.txt -H "Content-Type: application/json" -X POST -d
'{"action": "update", "id": "1", "enable": true, "security": {"wpa2Personal": {"key": "thisIsNewPassword"}}}'
http://192.168.1.1/api/config.ssid.profile
```

```
{
  "stat": "ok",
  "response": {
    "1": {
      "name": "Main SSID",
      "enable": true,
      "captivePortal": true,
      "incontrolManaged": false,
      "broadcast": true,
      "security": {
        "policy": "WPA2 Personal",
        "wpa2Personal": {
          "fastTransition": true,
          "key": "thisIsNewPassword"
        }
      }
    }
  },
  "order": [
    1
  ]
}
```

## POST /api/config.wan.connection

API beta

Update the WAN connection settings, most of option will update only when the information is provided.

*Available in 8 or later*

*Deprecate after .*

### Input Parameters

	Type	Notation	Mandatory	Description
<b>action</b>	String	{update}	require	Action of the API, now only support update.
<b>list</b>	Array	list of <WAN_Config_Obj>	require	List of the WAN connection object which is going to update

### <WAN\_Config\_Obj>

	Type	Notation	Mandatory	Description
<b>id</b>	Number	<conn_id>	optional	WAN connection ID
<b>name</b>	String	<string>	optional	WAN connection name
<b>enable</b>	Boolean	<boolean>	optional	Enable the WAN connection
<b>schedule</b>	Number Null	<number> <null>	optional	Schedule ID for the WAN To disable schedule, give the JSON null
<b>connection</b>	Object	<Connection_Obj>	optional	Connection information
<b>modem</b>	Object	<Modem_Obj>	optional	Modem information

Only support when the WAN is modem type

<b>cellular</b>	Object	<Cellular_Obj>	optional	Cellular information Only support when the WAN is cellular
<b>wifi</b>	Object	<Wifi_Obj>	optional	Wi-Fi WAN information Only support when the WAN is Wi-Fi
<b>wifiProfile</b>	-	-	optional	Wi-Fi Profile which is used by the Wi-Fi WAN Only support when the WAN is Wi-Fi NOTE: This field is not confirmed yet  If you want to manage the Wi-Fi profile, try the API below: POST cmd.wifi.connect POST cmd.wifi.disconnect POST cmd.wifi.forget Please make sure the SSID is nearby
<b>physical</b>	Object	<Physical_Obj>	optional	Physical information
<b>healthcheck</b>	Object	<Healthcheck_Obj>	optional	Healthcheck information
<b>bandwidthAllowanceMonitor</b>	Object	<BW_Allowance_Monitor_Obj>	optional	Bandwidth allowance monitor
<b>multipleIp</b>	Array	list of <ipv4>	optional	Additional IP
<b>ddns</b>	Object	<DDNS_Obj>	optional	Dynamic DNS service

### <Connection\_Obj>

	Type	Notation	Mandatory	Description
<b>cellularModule</b>	Object	<Cellular_Module_Obj>	optional	Cellular Module
<b>routingMode</b>	String	{ IP Forwarding, NAT }	optional	Routing Mode
<b>pepvpnNat</b>	Boolean	<boolean>	optional	PepVPN traffic via this WAN connection be in bridge (IP forwarding), with no NAT involved
<b>useLanIp</b>	Boolean	<boolean>	optional	Local out to IP forwarding WAN traffic will SNAT to default trunk LAN IP instead of WAN IP
<b>method</b>	Object	<Connection_Method_Obj>	optional	Connection method information This field only for Static IP, DHCP, PPPoE, L2TP, GRE and OpenVPN
<b>dns</b>	Object	<DNS_Obj>	optional	DNS information
<b>priority</b>	Number	<integer>	optional	WAN Priority
<b>groupSet</b>	Number	<integer>	optional	Group number if support multiple groups of WAN
<b>ignoreDefaultGateway</b>	Boolean	<boolean>	optional	Ignore default gateway
<b>hotStandBy</b>	Boolean	<boolean>	optional	Hot Stand by state
<b>idleTimeout</b>	Number Null	<integer> <null>	optional	Idle timeout To disable idle timeout, give the JSON null
<b>icmpPing</b>	Boolean	<boolean>	optional	ICMP Ping
<b>bandwidth</b>	Object	<Bandwidth_Map_Obj>	optional	Bandwidth information

### <Cellular\_Module\_Obj>

	Type	Notation	Mandatory	Description
<b>networkMode</b>	String	<string>	optional	Network Mode

### <Connection\_Method\_Obj>

	Type	Notation	Mandatory	Description
<b>type</b>	String	{ staticIp, dhcp, pppoe, l2tp, gre, openvpn }	optional	Connection method type
<b>detail</b>	Object	<DHCP_Obj> <Static_IP_Obj> <PPPoE_Obj> <L2TP_Obj> <GRE_Obj> <OpenVPN_Obj>	optional	Detail of connection method To update the connection method, 'type' cannot be absent



## <DHCP\_Obj>

	Type	Notation	Mandatory	Description
<b>hostname</b>	String	<string>	optional	Hostname
<b>ipPassthrough</b>	Boolean	<boolean>	optional	IP passthrough Only valid when that is not in drop in mode and port type is cellular or ethernet
<b>staticRoute</b>	Array	list of <Network_Obj>	optional	Static Route for IP passthrough Only valid when that is not in drop in mode and port type is cellular or ethernet and 'ipPassthrough' is true

## <Static\_IP\_Obj>

	Type	Notation	Mandatory	Description
<b>ip</b>	String	<ipv4>	require	IP address
<b>mask</b>	Number	[ 0, 32 ]	require	Subnet mask
<b>gateway</b>	String	<ipv4>	require	Gateway
<b>ipPassthrough</b>	Boolean	<boolean>	optional	IP passthrough Only valid when that is not in drop in mode and port type is cellular or ethernet
<b>staticRoute</b>	Array	list of <Network_Obj>	optional	Static Route for IP passthrough Only valid when that is not in drop in mode and port type is cellular or ethernet and 'ipPassthrough' is true

## <PPPoE\_Obj>

	Type	Notation	Mandatory	Description
<b>username</b>	String	<string>	require	Username
<b>password</b>	String	<string>	require	Password
<b>service</b>	String Null	<string> <null>	optional	Service Information which is provide by Internet Service Provider(ISP) To clear the setting, give the JSON null
<b>ip</b>	String	<ipv4>	optional	IP address Information which is provide by Internet Service Provider(ISP) To clear the setting, give the JSON null
<b>managementNetwork</b>	Object	<Network_Obj>	optional	Management IP Address Information which is provide by Internet Service Provider(ISP) To clear the setting, give the JSON null
<b>keepaliveInterval</b>	Number Null	<integer> <null>	optional	Keep alive interval To clear the setting, give the JSON null
<b>keepaliveRetry</b>	Number Null	<integer> <null>	optional	Keep alive retry To clear the setting, give the JSON null

## <L2TP\_Obj>

	Type	Notation	Mandatory	Description
<b>username</b>	String	<string>	require	Username
<b>password</b>	String	<string>	require	Password
<b>host</b>	String	<ipv4>	require	Host IP address
<b>staticip</b>	Object Null	<Static_IP_Common_Obj> <null>	optional	Static IP To clear the setting, give the JSON null

## <GRE\_Obj>

	Type	Notation	Mandatory	Description
<b>staticip</b>	Object	<Static_IP_Common_Obj>	optional	Static IP This field is mandatory when the Port is ethernet
<b>host</b>	String	<ipv4>	require	Host IP address
<b>local</b>	String	<ipv4>	require	Lcoal IP address
<b>remote</b>	String	<ipv4>	require	Remote IP address
<b>nat</b>	String	<ipv4>	optional	NAT IP address

### <OpenVPN\_Obj>

	Type	Notation	Mandatory	Description
<b>username</b>	String	<string>	optional	Username
<b>password</b>	String	<string>	optional	Password

### <Static\_IP\_Common\_Obj>

	Type	Notation	Mandatory	Description
<b>ip</b>	String	<ipv4>	require	IP address
<b>mask</b>	Number	[ 0, 32 ]	require	Subnet mask
<b>gateway</b>	String	<ipv4>	require	Gateway

### <Network\_Obj>

	Type	Notation	Mandatory	Description
<b>ip</b>	String	<ipv4>	require	IP address
<b>mask</b>	Number	[ 0, 32 ]	require	Subnet mask

### <DNS\_Obj>

	Type	Notation	Mandatory	Description
<b>auto</b>	Boolean	<boolean>	optional	Auto DNS
<b>host</b>	Array	list of <ipv4>	optional	Host IP addresses

### <Bandwidth\_Map\_Obj>

	Type	Notation	Mandatory	Description
<b>upload</b>	Object	<Bandwidth_Obj>	optional	Bandwidth upload information
<b>download</b>	Object	<Bandwidth_Obj>	optional	Bandwidth download information

### <Bandwidth\_Obj>

	Type	Notation	Mandatory	Description
<b>value</b>	Number	<number>	require	Upload / Download value Minimum - 1 kbps Maximum - 10 Gbps
<b>unit</b>	String	{ kbps, Mbps, Gbps }	require	Unit

### <Modem\_Obj>

	Type	Notation	Mandatory	Description
<b>operator</b>	Object Null	<Operator_Obj> <null>	optional	Operator information To clear the setting, give the JSON null
<b>simPin</b>	String Null	<string> <null>	optional	SIM Pin
<b>mobileType</b>	String	{ 4G, 3G, 2G }	optional	Mobile type
<b>huaweiBand</b>	Array	list of { GSM1900, GSM900/GSM1800/WCDMA2100 }	optional	The Band for Huawei Modem

### <Cellular\_Obj>

	Type	Notation	Mandatory	Description
<b>useExternalAntenna</b>	Boolean	<boolean>	optional	Use external antenna
<b>simCardScheme</b>	String	{ <empty>, 1, 2, alternate, remote_sim }	optional	SIM card scheme <empty> - Default (Internal / Both SIMs) 1 - SIM A only

2 - SIM B only  
 alternate - Alternate periodically between SIM A only and SIM B only  
 remote\_sim - Remote SIM (The API error if the device not support remote SIM)

<b>preferredSim</b>	Number	{1, 2}	optional	Preferred SIM
<b>idleTimeout</b>	Number Null	<integer> <null>	optional	Idle timeout To disable idle timeout, give the JSON null
<b>failbackTimeout</b>	Number Null	<integer> <null>	optional	Failback timeout To disable failback timeout, give the JSON null
<b>remoteSim</b>	ArrayNull	list of <string> <null>	optional	Remote SIM information
<b>alternateSim</b>	Object	<Alternate_SIM_Obj>	optional	Alternate SIM information Only suport when simCardScheme is alternate
<b>sim</b>	Array	list of <SIM_Obj>	optional	SIM information
<b>signalThreshold</b>	Object	<Signal_Threshold_Obj>	optional	Signal threshold

### <Alternate\_SIM\_Obj>

	Type	Notation	Mandatory	Description
<b>day</b>	Number	<integer>	require	Alternate SIM day
<b>hour</b>	Number	<integer>	require	Altherate SIM hour

### <SIM\_Obj>

	Type	Notation	Mandatory	Description
<b>id</b>	Number	<integer>	require	SIM ID
<b>carrierSelection</b>	Array Null	list of <Carrier_Selection_Obj> <null>	require	Carrier selection
<b>mobileType</b>	String	{ LTE, 3G, 2G }	optional	Mobile type
<b>optimalNetwork</b>	Object Null	<Optimal_Network_Obj> <null>	optional	Optimal Network To clear the setting, give the JSON null
<b>bandSelection</b>	Array Null	list of <string> <null>	optional	Band Selection To disable band selection, give the JSON null  NOTE: The string format to be confirm
<b>roaming</b>	Object	<Roaming_Obj>	optional	Roaming
<b>authentication</b>	String Null	{pap, chap} <null>	optional	Authentication mode
<b>operator</b>	Object Null	<Operator_Obj> <null>	optional	Operator information
<b>simPin</b>	String Null	<string><null>	optional	SIM Pin
<b>bandwidthAllowanceMonitor</b>	Object	<BW_Allowance_Monitor_Obj>	optional	Bandwidth allowance monitor Only valid when the device support bandwidth allowance of cellular

### <Carrier\_Selection\_Obj>

	Type	Notation	Mandatory	Description
<b>mcc</b>	String	<string>	optional	MCC
<b>mnc</b>	String	<string>	optional	MNC
<b>pcs</b>	Number	<integer>	optional	PCS
<b>name</b>	String	<string>	optional	Name
<b>plmn</b>	String	<string>	optional	PLMN

### <Signal\_Threshold\_Obj>

Type	Notation	Mandatory	Description
------	----------	-----------	-------------

<b>signalLevel</b>	Array Null	list of [0, 5] <null>	optional	Signal Level
<b>rsrp</b>	Array Null	list of [-140, -44] <null>	optional	RSRP
<b>sinr</b>	Array Null	list of [-100, 100] <null>	optional	SINR
<b>rsSI</b>	Array Array Null	list of [-125, -10] list of [-192, 63] <null>	optional	RSSI For Cellular WAN - [-125, -10] For Wi-Fi WAN - [-192, 63] To remove rssi, give the JSON null

### <Optimal\_Network\_Obj>

	Type	Notation	Mandatory	Description
<b>discovery</b>	Number	[ 5, 480 ]	optional	Optimal network discovery
<b>period</b>	Array	list of <integer>	optional	Optimal network period

### <Roaming\_Obj>

	Type	Notation	Mandatory	Description
<b>enable</b>	Boolean	<boolean>	optional	Roaming enable
<b>accessControlList</b>	Array Null	list of <integer> <null>	optional	Access control list No effect at this moment
<b>mode</b>	String Null	{ whitelist, blacklist }	optional	Roaming mode
<b>name</b>	String	<string>	optional	Name
<b>plmn</b>	String	<string>	optional	PLMN

### <Operator\_Obj>

	Type	Notation	Mandatory	Description
<b>apn</b>	String	<string>	optional	APN
<b>username</b>	String	<string>	optional	Username for the APN
<b>password</b>	String	<string>	optional	Password for the APN
<b>dialNumber</b>	String	{1234567890*#}	optional	Dial Number Only support in modem type

### <Wifi\_Obj>

	Type	Notation	Mandatory	Description
<b>country</b>	Number	<integer>	optional	Country ID Only for beta, make sure you know the ID is representing the country you wanted.
<b>channelWidth</b>	String	{ 20 MHz, 20/40 MHz, 40MHz, 80 MHz, 20/40/80 MHz, auto }	optional	Channel width
<b>channel</b>	Array	list of <integer>	optional	Channel Only for beta, make sure all channels in the array are correct
<b>power</b>	String	{ custom, auto, manual, high, medium, low, max }	optional	Power
<b>powerBoost</b>	Boolean	<boolean>	optional	Power Boost
<b>dataRate</b>	String	MCS{[0, 9]}	optional	Data Rate Only for beta, make sure data string is correct and match the channel width
<b>roaming</b>	Object	<Wifi_Roaming_Obj>	optional	Roaming information
<b>autoConnect</b>	Boolean	<boolean>	optional	Auto Connect
<b>beaconMissCounter</b>	Number	[ 2, 100 ]	optional	Beacon miss counter
<b>channelScanInterval</b>	Number	[ 5, 1000 ]	optional	Channel scan interval
<b>signalThreshold</b>	Object	<Signal_Threshold_Obj>	optional	Signal Threshold

## <Wifi\_Roaming\_Obj>

	Type	Notation	Mandatory	Description
<b>enable</b>	Boolean	<boolean>	optional	Enable
<b>algorithm</b>	Object	<Wifi_Roaming_Algo_Obj>	optional	Roaming Algorithm

## <Wifi\_Roaming\_Algo\_Obj>

	Type	Notation	Mandatory	Description
<b>type</b>	String	{ normal, advanced, express }	optional	Algorithm type
<b>detail</b>	Object	<Wifi_Roaming_Algo_Detail_Obj>	optional	Algorithm detail

## <Wifi\_Roaming\_Algo\_Detail\_Obj>

	Type	Notation	Mandatory	Description
<b>signalLevel</b>	Object	<Wifi_Roaming_Algo_Signal_Level_Obj>	optional	Signal level
<b>checkInterval</b>	Number	[ 5, 3600 ]	optional	Check interval
<b>intensiveScan</b>	Object	<Wifi_Roaming_Algo_Adv_Intensive_Scan_Obj>	optional	Intensive scan This field only for advanced
<b>diagnosticLevel</b>	String	{ minimum, basic, detail }	optional	Diagnostic level This field only for express
<b>signalMode</b>	Object	<Wifi_Roaming_Algo_Exp_Signal_Mode_Obj>	optional	Signal mode This field only for express
<b>forceRoam</b>	Object	<Wifi_Roaming_Algo_Exp_Force_Roam_Obj>	optional	Intensive scan This field only for express
<b>confirmPeriod</b>	Number	<integer>	optional	Confirm period This field only for express
<b>backupDisconnect</b>	Object	<Wifi_Roaming_Algo_Exp_Backkup_Disconnect_Obj>	optional	Backup disconnect This field only for express
<b>authenticationTimeout</b>	Number	<integer>	optional	Authentication timeout This field only for express

## <Wifi\_Roaming\_Algo\_Signal\_Level\_Obj>

	Type	Notation	Mandatory	Description
<b>threshold</b>	Number	[ -95, -40 ]	optional	Signal level threshold
<b>gain</b>	Number	[ 5, 55 ]	optional	Signal level gain

## <Wifi\_Roaming\_Algo\_Adv\_Intensive\_Scan\_Obj>

	Type	Notation	Mandatory	Description
<b>enable</b>	Boolean	<boolean>	optional	Enable intensive scan
<b>signalLevel</b>	Number	[ -95, -40 ]	optional	Signal level for intensive scan
<b>scanInterval</b>	Number	[ 1, 3600 ]	optional	Scan interval for intensive scan

## <Wifi\_Roaming\_Algo\_Exp\_Signal\_Mode\_Obj>

	Type	Notation	Mandatory	Description
<b>type</b>	String	{ relative, absolute }	optional	Signal mode type
<b>detail</b>	Object	<Wifi_Roaming_Algo_Exp_Signal_Mode_Detail_Obj>	optional	Signal mode detail

## <Wifi\_Roaming\_Algo\_Exp\_Signal\_Mode\_Detail\_Obj>

	Type	Notation	Mandatory	Description
<b>minimumSignalDifference</b>	Number	[ 0, 94 ]	optional	Minimum signal difference Only valid for type is relative
<b>signalThreshold</b>	Object	<Wifi_Roaming_Algo_Exp_Signal_Threshold_Obj>	optional	Signal threshold Only valid for type is absolute

## &lt;Wifi\_Roaming\_Algo\_Exp\_Signal\_Threshold\_Obj&gt;

	Type	Notation	Mandatory	Description
<b>upper</b>	Number	[ -95, -1 ]	optional	Upper limit
<b>lower</b>	Number	[ -95, -1 ]	optional	Lower limit

## &lt;Wifi\_Roaming\_Algo\_Exp\_Dynamic\_Zone\_Obj&gt;

	Type	Notation	Mandatory	Description
<b>inner</b>	Number	[ 0, 95 ]	optional	Inner limit
<b>outer</b>	Number	[ 0, 95 ]	optional	Outer limit

## &lt;Wifi\_Roaming\_Algo\_Exp\_Force\_Roam\_Obj&gt;

	Type	Notation	Mandatory	Description
<b>enable</b>	Boolean	<boolean>	optional	Force roam enable
<b>threshold</b>	Number	[ -95, -1 ]	optional	Force roam threshold

## &lt;Wifi\_Roaming\_Algo\_Exp\_Backup\_Disconnect\_Obj&gt;

	Type	Notation	Mandatory	Description
<b>mode</b>	String	{ no, immediate, delay }	optional	Backup disconnect mode
<b>delay</b>	Number	<integer>	optional	Delay value Only valid for mode is delay

## &lt;Signal\_Threshold\_Obj&gt;

	Type	Notation	Mandatory	Description
<b>signalLevel</b>	Array Null	list of [0, 5] <null>	optional	Signal Level
<b>rsrp</b>	Array Null	list of [-140, -44] <null>	optional	RSRP
<b>sinr</b>	Array Null	list of [-100, 100] <null>	optional	SINR
<b>rsi</b>	Array Array Null	list of [-125, -10] list of [-192, 63] <null>	optional	RSSI For Cellular WAN - [-125, -10] For Wi-Fi WAN - [-192, 63] To remove rssi, give the JSON null

## &lt;Physical\_Obj&gt;

	Type	Notation	Mandatory	Description
<b>speed</b>	String	{Auto, 1000baseTx-FD, 100baseTx-FD, 100baseTx-HD, 10baseT-FD, 10baseT-HD}	optional	Speed The field only validate when the port is ethernet NOTE: 1000baseTx-FD only support Giga Ethernet port
<b>advertise</b>	Boolean	<boolean>	optional	Advertise The field only validate when the port is ethernet
<b>mtu</b>	Number Null	[ 576, 1492 ] [ 576, 1476 ] [ 576, 9000 ] [ 576, 1500 ] <null>	optional	MTU value For PPPoE, the max value is 1492 For GRE, the max value is 1476 For Jumbo frame, the max value is 9000 Otherwise, the max value is 1500 To clear the setting, give the JSON null
<b>mss</b>	Number Null	[ 536, <mtu> - 40 ] <null>	optional	MSS value To clear the setting, give the JSON null
<b>ttl</b>	Number Null	[ 1, 255 ] <null>	optional	TTL value To clear the setting, give the JSON null
<b>mac</b>	String	<mac>	optional	MAC address

	Null	<null>		The field only available when the connectionType is ethernetTo clear the setting, give the JSON null
<b>vlan</b>	Number Null	[ 1, 4094] [ 1, 10] <null>	optional	VLAN ID The field only available when the port is ethernet or VDSLFor ethernet, the max value is 4094 For VDSL, the max value is 10 To clear the setting, give the JSON null
<b>vpi</b>	Number	[ 1, 255]	optional	VPI value The field only available when the port is ADSL or VDSL
<b>vci</b>	Number	[ 32, 65535]	optional	VCI value The field only available when the port is ADSL or VDSL
<b>greUplink</b>	Number	<conn_id>	optional	GRE uplink The field only available when the port is GRE
<b>openvpn</b>	Object	<Physical_OpenVPN_Obj>	optional	OpenVPN information

### <Physical\_OpenVPN\_Obj>

	Type	Notation	Mandatory	Description
<b>uplink</b>	Array	list of <OpenVPN_Uplink_Priority_Obj>	optional	OpenVPN Uplink
<b>failback</b>	Boolean	<boolean>	optional	OpenVPN connection failback

### <OpenVPN\_Uplink\_Priority\_Obj>

	Type	Notation	Mandatory	Description
<b>id</b>	Number	<conn_id>	require	WAN connection ID
<b>priority</b>	Number	<integer>	require	Priority

### <Healthcheck\_Obj>

	Type	Notation	Mandatory	Description
<b>enable</b>	Boolean	<boolean>	require	Healthcheck enable
<b>method</b>	Object	<Healthcheck_Method_Obj>	optional	Healthcheck method
<b>timeout</b>	Number	[ 200, 10000 ]	optional	Healthcheck timeout Normally, the range is 801-10000 200-800 is for ping only
<b>interval</b>	Number	[ 5, 3600 ]	optional	Healthcheck interval
<b>retry</b>	Number	[ 1, 20 ]	optional	Healthcheck retry
<b>recovery</b>	Number	[ 1, 20 ]	optional	Healthcheck recovery

### <Healthcheck\_Method\_Obj>

	Type	Notation	Mandatory	Description
<b>type</b>	String	{ ping, nslookup, http, smartcheck, openvpn}	require	Healthcheck enable
<b>detail</b>	Object	<Healthcheck_Method_Host_Obj>	require	Healthcheck detail The field has no effect for type is openvpn

### <Healthcheck\_Method\_Host\_Obj>

	Type	Notation	Mandatory	Description
<b>includedPublic</b>	Boolean	<boolean>	optional	Included public IP This field only for method type 'nslookup'
<b>host</b>	Array	list of <ipv4> list of <Healthcheck_Method_HTTP_Obj>	optional	Host IP address The maximum array size is 2 For method type ping, nslookup, smartcheck, list of <ipv4> For method type http, list of <Healthcheck_Method_HTTP_Obj>

### <Healthcheck\_Method\_HTTP\_Obj>

	Type	Notation	Mandatory	Description
<b>host</b>	Array	list of <URL_Pattern_Obj>	optional	Host URL pattern

### <URL\_Pattern\_Obj>

	Type	Notation	Mandatory	Description
<b>url</b>	String	<string>	require	URL
<b>pattern</b>	String	<string>	require	Pattern

### <BW\_Allowance\_Monitor\_Obj>

	Type	Notation	Mandatory	Description
<b>enable</b>	Boolean	<boolean>	optional	Bandwidth allowance monitor enable
<b>action</b>	Array	list of { email, disconnect, restrict }	optional	The actions which the allowance is reach
<b>start</b>	Number	[ 0, 28 ]	optional	Bandwidth allowance monitor start day
<b>monthlyAllowance</b>	Object	<BW_Allowance_Monitor_Monthly_Obj>	optional	Bandwidth monthly allowance

### <BW\_Allowance\_Monitor\_Monthly\_Obj>

	Type	Notation	Mandatory	Description
<b>value</b>	Number	<integer>	require	Bandwidth allowance monitor monthly allowance value
<b>unit</b>	String	{ MB, GB, TB }	require	Bandwidth allowance monitor monthly allowance unit for value

### <DDNS\_Obj>

	Type	Notation	Mandatory	Description
<b>enable</b>	Boolean	<boolean>	optional	DDNS enable
<b>provider</b>	String	{ changeip, dyndns, noip, tzo, dnsomatic, others }	optional	DDNS service provider
<b>customUrl</b>	String	<string>	optional	Custom URL This field only valid for provider is others
<b>useWanIp</b>	Boolean	<boolean>	require	Use WAN IP
<b>username</b>	String	<string>	require	Username for the service
<b>password</b>	String	<string>	require	Password for the service
<b>host</b>	Array	list of <domain>	require	Host Allow empty array when the provider is dnsmatic

## Return Parameters

### cURL Example

```
> curl -b cookies.txt -H "Content-Type: application/json" -X POST -d '{"action":"update","list":[{"id":1,"enable":true}]}' http://192.168.1.1/api/config.wan.connection
```

```
{
  "stat": "ok",
  "response": {
    "1": {
      "name": "WAN 1",
      "asLan": false,
      "enable": true,
      "active": true,
      "multipleIp": [],
      "connection": {
        "method": "dhcp",
        "mode": "NAT",
        "icmpPing": true,

```



```
    "priority": 1,
    "dns": {
      "auto": true
    },
    "ddns": {
      "username": "username",
      "password": "@~HiDdEn~@",
      "host": [
        "kjkjkjkj.com"
      ],
      "provider": "noip",
      "enable": true
    },
    "bandwidth": {
      "upload": {
        "bandwidth": 100000,
        "unit": "kbps"
      },
      "download": {
        "bandwidth": 100000,
        "unit": "kbps"
      }
    },
    "schedule": 4,
    "dhcp": {
      "hostname": ""
    }
  },
  "healthcheck": {
    "method": "nslookup",
    "timeout": 5,
    "interval": 5,
    "retry": 3,
    "recovery": 3,
    "enable": true,
    "nslookup": {
      "includePublic": false,
      "host": [
        "208.67.222.222",
        "208.67.220.220"
      ]
    }
  }
},
"2": {
  "name": "WAN2",
  "asLan": false,
  "enable": true,
  "active": true,
  "multipleIp": [],
  "connection": {
    "method": "dropIn",
    "mode": "IP Forwarding",
    "icmpPing": true,
    "priority": 0,
    "dns": {
      "auto": false,
      "host": [
        "3.3.3.3"
      ]
    },
    "ddns": {
      "enable": false
    }
  },
}
```

```

        "bandwidth": {
            "upload": {
                "bandwidth": 100000,
                "unit": "kbps"
            },
            "download": {
                "bandwidth": 100000,
                "unit": "kbps"
            }
        },
        "pepVpnNat": true,
        "dropIn": {
            "ip": "169.254.0.1",
            "mask": 24,
            "gateway": "22.2.2.2"
        }
    },
    "healthcheck": {
        "method": "nslookup",
        "timeout": 5,
        "interval": 5,
        "retry": 3,
        "recovery": 3,
        "enable": true,
        "nslookup": {
            "includePublic": true
        }
    }
},
"order": [
    1,
    2
]
}
}

```

## POST /api/config.wan.connection.priority

### API

Change the priority of the WAN connection

The API will return WAN connection ID, priority and enable information which are just updated.

*Available in 7.1.1 or later*

### Input Parameters

	Type	Notation	Mandatory	Description
<b>instantActive</b>	String	<boolean>	optional	Priority should be updated and take effect immediately When omitted, the configuration will be saved normally, and pending for the explicit apply changes action to take effect
<b>list</b>	Array	list of <WAN_Config_Priority_Obj>	optional	The list of object for changing the priority.

### <WAN\_Config\_Priority\_Obj>

	Type	Notation	Mandatory	Description
<b>connId</b>	Number	<number>	require	WAN connection ID
<b>priority</b>	Number	<number>	optional	Priority of the WAN connection
<b>enable</b>	Boolean	<boolean>	optional	Enable the WAN connection

### Return Parameters

## Return JSON

	Type	Notation	Description
<b>order</b>	Array	list of <conn_id>	The order of WAN ID
<b>&lt;conn_id&gt;</b>	Object	<WAN_Config_Priority_Obj>	WAN config information

### <WAN\_Config\_Priority\_Obj>

	Type	Notation	Description
<b>name</b>	String	<string>	Name of the WAN connection
<b>priority</b>	Number	<number>	Priority of the WAN connection
<b>enable</b>	Boolean	<boolean>	WAN connection enabled or not

## cURL Example

```
> curl -b cookies.txt -H "Content-Type: application/json" -X POST -d '{"instantActive":true,"list":[{"connId":1,"priority":1},{"connId":2,"priority":2}]}'  
http://192.168.1.1/api/config.wan.connection.priority
```

```
{  
  "stat": "ok",  
  "response": {  
    "1": {  
      "name": "WAN 1",  
      "priority": 1,  
      "enable": true  
    },  
    "2": {  
      "name": "WAN 2",  
      "priority": 2,  
      "enable": true  
    },  
    "order": [  
      1,  
      2  
    ]  
  }  
}
```

## GET /api/info.firmware

API **internal testing**

Obtain the firmware information of the device.  
The API can also call before login, but it will only return the firmware version which is in used

Available in 7.1.1 or later

### Return Parameters

#### Return JSON

	Type	Notation	Description
<b>order</b>	Array	<fw_id>	The order of firmware information by ID.
<b>&lt;fw_id&gt;</b>	Object	<Firmware_Obj>	Firmware information.

### <Firmware\_Obj>

	Type	Notation	Description
<b>version</b>	String	<string>	Firmware version
<b>bootable</b>	Boolean	<boolean>	Firmware is bootable or not

inUse

Boolean

<boolean>

Firmware is running or not

### cURL Example

```
> curl -b cookies.txt http://192.168.1.1/api/info.firmware
```

```
{
  "stat": "ok",
  "response": {
    "1": {
      "version": "7.0.3 build 2765",
      "bootable": true,
      "inUse": false
    },
    "2": {
      "version": "7.1.0 build 2860",
      "bootable": true,
      "inUse": true
    },
    "order": [
      1,
      2
    ]
  }
}
```

## GET /api/info.location

API

Obtain the GPS and other information which is related location  
The API will return fail when the device is not support GPS module.

*Available in 8.0.1 or later*

### Return Parameters

#### Return JSON

	Type	Notation	Description
<b>gps</b>	Boolean	<boolean>	The GPS signal is valid or not
<b>location</b>	Object	<GPS_Location_Obj>	GPS Location information

#### <GPS\_Location\_Obj>

	Type	Notation	Description
<b>latitude</b>	Number	<double>	-
<b>longitude</b>	Number	<double>	-
<b>altitude</b>	Number	<double>	-
<b>speed</b>	Number	<double>	-
<b>heading</b>	Number	<double>	-
<b>pdop</b>	Number	<double>	Position Dilution Of Precision
<b>hdop</b>	Number	<double>	Horizontal Dilution Of Precision
<b>vdop</b>	Number	<double>	Vertical Dilution Of Precision
<b>timestamp</b>	Number	<integer>	-

### cURL Example

```
> curl -b cookies.txt http://192.168.1.1/api/info.location
```

```
{
```

```

"stat": "ok",
"response": {
  "gps": true,
  "location": {
    "latitude": 22.340134,
    "longitude": 114.152588,
    "altitude": 55.1,
    "speed": 0.026751,
    "heading": 356.887,
    "pdop": 1.3,
    "hdop": 1,
    "vdop": 0.8,
    "timestamp": 1311972720
  }
}
}

```

## GET /api/status.client

### API

Obtain client of the device

*Available in 8.0.1 or later*

### Input Parameters

	Type	Notation	Mandatory	Description
<b>vlanId</b>	Number	<integer>	optional	Filter out the clients which are not provided. If the field is not provided, all VLAN ID and untagged LAN client will be shown.
<b>activeOnly</b>	String	{ yes, no }	optional	Filter out the clients which are not active. If the field is not provided, active and inactive client will be shown.
<b>connectionType</b>	Array	list of { ethernet, wireless, pptp, stroute, l2tp, openvpn, pepvpn, other }	optional	Filter out the clients which are not provided in the array
<b>size</b>	Number	[ 1, 10000000 ]	optional	Limited the number of client return. If the field is not provided, 1000 clients will be return.
<b>outputWeight</b>	String	{ full, normal, lite }	optional	Filter of the <Client_Obj> content size. full - return all the information which is object can provide. normal - return ip, connectionType, clientType, name, mac, bssid, vlanId, essid and active lite - return ip, connectionType, clientType, name, mac, bssid and vlanId
<b>infoType</b>	Array	list of { ip, connectionType, lease, name, mac, bssid, port, vlanId, essid, active, signalStrength, speed }	optional	Filter of the <Client_Obj> content size. This field will override the outputWeight param. If both outputWeight and infoType are absent, the content will outputWeight=normal.

### Return Parameters

#### Return JSON

	Type	Notation	Description
<b>list</b>	Array	list of <Client_Obj>	The list of the client object.

#### <Client\_Obj>

	Type	Notation	Description
--	------	----------	-------------

<b>ip</b>	String	<ipv4>	IP Address
<b>connectionType</b>	String	{ ethernet, wireless, pptp, stroute, l2tp, openvpn, pepvpn, other }	Connection Type of the client If the client is not active, this param will be absent. In fw 8.1.0 or before, it return 'ethernet' accidentally. Better check the 'active' param before this.
<b>lease</b>	Object	<Lease_Obj>	Lease type and expires in second The field only available when the connectionType is ethernet or wireless
<b>name</b>	String	<string>	The name of the drive if any.
<b>mac</b>	String	<mac>	MAC address of the client
<b>bssid</b>	String	<mac>	BSSID of the Wi-Fi. This field only present when connectionType=wireless
<b>vlanId</b>	Number	<integer>	Which VLAN the client connected. When it connects to untagged LAN, this field will be absent.
<b>ssid</b>	String	<string>	SSID of the Wi-Fi. This field only present when connectionType=wireless
<b>active</b>	Boolean	<boolean>	The active state of the client
<b>signalStrength</b>	Object	<Signal_Obj>	Signal Strength information Deprecated in fw 8.1.0
<b>signal</b>	Object	<Signal_Detail_Obj>	Signal Strength and Level information First present in fw 8.1.0
<b>speed</b>	Object	<Bandwidth_Obj>	Speed information

#### <Lease\_Obj>

	Type	Notation	Description
<b>expiresIn</b>	Number	<integer>	Lease expires in second
<b>type</b>	String	{ normal, dhcp, wins }	Lease Type

#### <Signal\_Obj>

	Type	Notation	Description
<b>value</b>	Number	<number>	Strength of the Wi-Fi signal
<b>unit</b>	String	{ dBm }	Unit of the signal

#### <Signal\_Detail\_Obj>

	Type	Notation	Description
<b>strength</b>	Number	<number>	Strength of the Wi-Fi signal in dBm
<b>level</b>	Number	[1, 5]	Signal Level

#### <Bandwidth\_Obj>

	Type	Notation	Description
<b>download</b>	Number	<number>	Download rate
<b>upload</b>	Number	<number>	Upload rate
<b>unit</b>	String	{ kbps }	Unit of the speed

### cURL Example

```
> curl -b cookies.txt http://192.168.1.1/api/status.client?connectionType=ethernet wireless
```

```
{
  "stat": "ok",
  "response": {
    "list": [
      {
        "ip": "192.168.50.4",
        "connectionType": "wireless",
        "name": "Android client",
```

```

    "mac": "9C:5C:F9:2B:85:99",
    "bssid": "00:1A:DD:ED:8F:69",
    "ssid": "PEPWAVE_D3B1",
    "active": true
  },
  {
    "ip": "192.168.50.11",
    "connectionType": "ethernet",
    "name": "macOS client",
    "mac": "E4:25:E7:8A:D3:12",
    "active": false
  },
  {
    "ip": "192.168.50.17",
    "connectionType": "wireless",
    "name": "iOS client",
    "mac": "34:12:98:9B:11:D7",
    "active": false
  }
]
}

```

## GET /api/status.lan.profile

### API

Obtain Balance LAN Status

Available in 7.1.0 or later

### Input Parameters

	Type	Notation	Mandatory	Description
id	Array	<numlist>	optional	list the LAN information base on id, multiple value is accepted, if id is absent, all LAN will be return

### Return Parameters

#### Return JSON

	Type	Notation	Description
order	Array	list of <profile_id>	The order of LAN ID
<profile_id>	Object	<LAN_Status_Obj>	LAN status information

#### <LAN\_Status\_Obj>

	Type	Notation	Description
name	String	<string>	LAN / VLAN Name
vlanId	Number	[ 1, 4094 ]	VLAN ID. This field will not appear if vlanId is empty
ip	String	<ipv4>	IP address
mask	Number	<maskn>	Subnet mask

### cURL Example

```
> curl -b cookies.txt http://192.168.1.1/api/status.lan.profile
```

```

{
  "stat": "ok",
  "response": {
    "0": {

```

```

        "ip": "10.6.1.231",
        "mask": 16
    },
    "1": {
        "name": "Name 1",
        "ip": "10.6.1.231",
        "vlanId": 164,
        "mask": 16
    },
    "order": [
        0,
        1
    ]
}
}

```

## GET /api/status.pepvpn

API beta

Obtain PepVPN / SpeedFusion status

*Available in 7.1.0 or later*

### Input Parameters

	Type	Notation	Mandatory	Description
<b>infoType</b>	Array	{ profile, peer, tunnel }	optional	Choose the information which is wanted to obtain.
<b>lite</b>	String	{ yes, no }	optional	Limited data within the response will be get when this field set to "yes". Otherwise, all status information will be got.
<b>tunnelOption</b>	Array	list of <peer_id>	optional	Retrieve the tunnel information base on peer ID
<b>start</b>	Number	<number>	optional	Start number of the peer
<b>size</b>	Number	<number>	optional	Output size of the peer
<b>searchPattern</b>	String	<string>	optional	Search peer by string if the field is not empty
<b>serialNumber</b>	String	<sn>	optional	Search peer by serial number

### Return Parameters

#### Return JSON

	Type	Notation	Description
<b>profile</b>	Object	<Profile_Order_Obj>	PepVPN profile information
<b>peer</b>	Array	list of <Peer_Obj>	Peer Information
<b>tunnel</b>	Object	<Tunnel_Order_Obj>	tunnel Information, if tunnelOption is empty, the field will not be appeared

#### <Profile\_Order\_Obj>

	Type	Notation	Description
<b>order</b>	Array	list of <profile_id>	Order of the profile ID
<b>&lt;profile_id&gt;</b>	Object	<Profile_Obj>	Profile information by ID
<b>siteld</b>	String	<string>	Local ID of the device

#### <Profile\_Obj>

	Type	Notation	Description
<b>name</b>	String	<string>	PepVPN profile Name



<b>master</b>	Boolean	<boolean>	State that is master profile
<b>vlanId</b>	Number	<number>	VLAN ID of the profile. The field will not appear if lite=yes
<b>status</b>	String	{ START, AUTHEN, TUNNEL, ROUTE, CONFLICT, CONNECTED }	Status of the profile. The field will not appear if lite=yes
<b>conflictCount</b>	Number	<number>	Conflict count. The field will not appear if lite=yes
<b>peerCount</b>	Number	<number>	Peer count. The field will not appear if lite=yes
<b>userShared</b>	Boolean	<boolean>	Allow user shared. The field will not appear if lite=yes
<b>userCount</b>	Number	<number>	User count. The field will not appear if lite=yes
<b>type</b>	String	{ l3, l2, nats, natc }	Type of the profile. The field will not appear if lite=yes

### <Peer\_Obj>

	Type	Notation	Description
<b>serialNumber</b>	String	<sn>	Serial Number of the peer device
<b>status</b>	String	{ START, AUTHEN, TUNNEL, ROUTE, CONFLICT, CONNECTED }	Status of the peer
<b>name</b>	String	<string>	Peer device name
<b>profileId</b>	Number	<number>	Profile ID of the peer connecting to
<b>secure</b>	Boolean	<boolean>	State the connection is secured or not
<b>type</b>	String	{ l3, l2, nats, natc }	Type of profile peer connection
<b>username</b>	String	<string>	Account username
<b>conflictRoute</b>	Array	list of <cidr>	Conflict Route of the connection. The field will only appear in Layer3 connection
<b>inactiveRoute</b>	Array	list of <cidr>	Inactive Route of the connection. The field will only appear in Layer3 connection
<b>route</b>	Array	list of <cidr>	Route of the connection. The field will only appear in Layer3 connection
<b>server</b>	String	<ipv4>	Server IP. The field will only appear in NAT connection
<b>client</b>	String	<cidr>	Client IP with subnet mask. The field will only appear in NAT connection
<b>bridge</b>	String	<ipv4>	IP of the bridge. The field will only appear in Layer2 connection
<b>vlanId</b>	Number	<number>	VLAN ID. The field will only appear in Layer2 connection
<b>peerId</b>	String	[<number>-<number>]	Unique ID of the peer

### <Tunnel\_Order\_Obj>

	Type	Notation	Description
<b>order</b>	Array	list of <peer_id>	Order of the peer ID
<b>&lt;peer_id&gt;</b>	Object	{<Tunnel_Obj>, <WAN_Order_Obj>}	Tunnel information by peer ID For fw8.1.0 or above, use <Tunnel_Obj> Before fw8.1.0, use <WAN_Order_Obj>

### <Tunnel\_Obj>

	Type	Notation	Description
<b>wan</b>	Object	<WAN_Order_Obj>	Tunnel information by WAN
<b>overall</b>	Object	<Overall_Obj>	Overall tunnel Statistic information

### <Overall\_Obj>

	Type	Notation	Description
<b>time</b>	Object	<Time_Object>	Time information of the tunnel
<b>receive</b>	Object	<Receive_Obj>	Receive information For fw8.1.0 or later

<b>transmit</b>	Object	<Transmit_Obj>	Transmit information For fw8.1.0 or later
-----------------	--------	----------------	--

### <WAN\_Order\_Obj>

	Type	Notation	Description
<b>order</b>	Array	list of <conn_id>	Order of the WAN connection ID
<b>&lt;conn_id&gt;</b>	Object	<WAN_Obj>	Tunnel Statistic information by WAN connection ID

### <WAN\_Obj>

	Type	Notation	Description
<b>id</b>	Number	<int>	WAN connection ID
<b>state</b>	String	{ INVALID, WAN_DOWN, WAN_DISABLED, DETECTING, FAILURE, REMOTE_FAILURE, COLD, STATNDBY, P- SUSPD, D-SUSPD, U- SUSPD, P-ACTIV, D- ACTIV, U-ACTIV, ACTIVE }	Status of the tunnel
<b>name</b>	String	<string>	WAN name
<b>time</b>	Object	<Time_Object>	Time information of the tunnel
<b>nanotime</b>	Object	<Time_Object>	Time information of the tunnel
<b>rtt</b>	Number	<number>	Round trip delay time of the remote peer WAN
<b>rx</b>	Array	{list of <number>, <number>}	Receive bytes of the remote peer WAN For local tunnel information, this field is array. Otherwise this field is a number Depreated in fw8.1.0
<b>tx</b>	Array	{list of <number>, <number>}	Transmit bytes of the remote peer WAN For local tunnel information, this field is array. Otherwise this field is a number Deprecated in fw8.1.0
<b>loss</b>	Array	{list of <number>, <number>}	Package loss of the remote peer WAN For local tunnel information, this field is array. Otherwise this field is a number Deprecated in fw8.1.0
<b>receive</b>	Object	<Receive_Obj>	Receive information For fw8.1.0 or later
<b>transmit</b>	Object	<Transmit_Obj>	Transmit information For fw8.1.0 or later
<b>remote</b>	Object	<WAN_Order_Obj>	Remote WAN tunnel status This field only appear in local tunnel information

### <Receive\_Obj>

	Type	Notation	Description
<b>byte</b>	Array, Number	{list of <number>, <number>}	Receive bytes of the remote peer WAN For local tunnel information, this field is array. Otherwise this field is a number  Absent for <Overall_Obj>
<b>packet</b>	Object	<Receive_Packet_Obj>	Receive packet of the remote peer WAN

### <Transmit\_Obj>

	Type	Notation	Description
<b>byte</b>	Array	{list of <number>, <number>}	Transmit bytes of the remote peer WAN For local tunnel information, this field is array. Otherwise this field is a number  Absent for <Overall_Obj>

<b>packet</b>	Object	<Transmit_Packet_Obj>	Time in nano second
---------------	--------	-----------------------	---------------------

### <Receive\_Packet\_Obj>

	<i>Type</i>	<i>Notation</i>	<i>Description</i>
<b>wan</b>	ArrayNumber	{list of <number>, <number>}	Receive wan packet For local tunnel information, this field is array. Otherwise this field is a number  Absent for <Overall_Obj>
<b>forward</b>	ArrayNumber	{list of <number>, <number>}	Receive forward packet For local tunnel information, this field is array. Otherwise this field is a number
<b>fragment</b>	ArrayNumber	{list of <number>, <number>}	Receive fragment packet For local tunnel information, this field is array. Otherwise this field is a number  Absent for <Overall_Obj>
<b>loss</b>	ArrayNumber	{list of <number>, <number>}	Receive loss packet For local tunnel information, this field is array. Otherwise this field is a number
<b>outOfOrder</b>	ArrayNumber	{list of <number>, <number>}	Receive out of order packet For local tunnel information, this field is array. Otherwise this field is a number
<b>recover</b>	ArrayNumber	{list of <number>, <number>}	Receive recover packet For local tunnel information, this field is array. Otherwise this field is a number  Absent for <Overall_Obj>
<b>discard</b>	ArrayNumber	{list of <number>, <number>}	Receive discard packet For local tunnel information, this field is array. Otherwise this field is a number  Absent for <Overall_Obj>

### <Transmit\_Packet\_Obj>

	<i>Type</i>	<i>Notation</i>	<i>Description</i>
<b>wan</b>	ArrayNumber	{list of <number>, <number>}	Transmit wan packet For local tunnel information, this field is array. Otherwise this field is a number  Absent for <Overall_Obj>
<b>forward</b>	ArrayNumber	{list of <number>, <number>}	Transmit forward packet For local tunnel information, this field is array. Otherwise this field is a number
<b>fragment</b>	ArrayNumber	{list of <number>, <number>}	Transmit fragment packet For local tunnel information, this field is array. Otherwise this field is a number  Absent for <Overall_Obj>
<b>loss</b>	ArrayNumber	{list of <number>, <number>}	Transmit loss packet For local tunnel information, this field is array. Otherwise this field is a number
<b>outOfOrder</b>	ArrayNumber	{list of <number>, <number>}	Transmit out of order packet For local tunnel information, this field is array. Otherwise this field is a number
<b>fec</b>	ArrayNumber	{list of <number>, <number>}	Transmit forward error correct packet For local tunnel information, this field is array. Otherwise this field is a number  Absent for <Overall_Obj>
<b>redundant</b>	ArrayNumber	{list of <number>, <number>}	Transmit redundant packet For local tunnel information, this field is array. Otherwise this field is a number  Absent for <Overall_Obj>

<Time\_Obj>

	Type	Notation	Description
<b>second</b>	Number	<number>	Time in second
<b>nanoSecond</b>	Number	<number>	Time in nano second

### cURL Example

```
> curl -b cookies.txt http://192.168.1.1/api/status.pevpn?infoType=profile  
peer&lite=yes&tunnelOption=1-1
```

```
{  
  "stat": "ok",  
  "response": {  
    "profile": {  
      "1": {  
        "name": "Next (1)",  
        "master": true  
      },  
      "2": {  
        "name": "Next (2 - 2)",  
        "master": true  
      },  
      "siteId": "999",  
      "order": [  
        2,  
        1  
      ]  
    },  
    "tunnel": {  
      "1-1": {  
        "wan": {  
          "1": {  
            "time": {  
              "second": 1292258,  
              "nanoSecond": 485618662  
            },  
            "rtt": 1,  
            "rx": [  
              1423988  
            ],  
            "tx": [  
              1334004  
            ],  
            "loss": [  
              0  
            ],  
            "priority": 1,  
            "state": "ACTIVE",  
            "name": "WAN 1"  
          },  
          "2": {  
            "priority": 0,  
            "state": "WAN_DOWN",  
            "name": "WAN 2"  
          },  
          "3": {  
            "priority": 0,  
            "state": "WAN_DISABLED",  
            "name": "WAN 3"  
          },  
          "4": {  
            "priority": 0,  
            "state": "WAN_DISABLED",  
            "name": "WAN 4"  
          }  
        }  
      }  
    }  
  }  
}
```

```

    },
    "5": {
      "priority": 0,
      "state": "WAN_DISABLED",
      "name": "WAN 5"
    },
    "6": {
      "priority": 0,
      "state": "WAN_DISABLED",
      "name": "Mobile Internet"
    },
    "order": [
      1,
      2,
      3,
      4,
      5,
      6
    ]
  },
  "overall": {
    "time": {
      "second": 1292258,
      "nanoSecond": 485618662
    },
    "receive": {
      "packet": {
        "forward": 32,
        "loss": 1,
        "outOfOrder": 0
      }
    },
    "transmit": {
      "packet": {
        "forward": 12,
        "loss": 0,
        "outOfOrder": 0
      }
    }
  }
},
"order": [
  "1-1"
]
},
"peer": [
  {
    "serialNumber": "1825-4131-B4E7",
    "status": "CONNECTED",
    "name": "Ke-B580-x64-30",
    "profileId": 1,
    "secure": true,
    "type": "l3",
    "username": "dev30",
    "route": [
      "192.168.30.0/24"
    ],
    "peerId": "1-1"
  }
]
}
}
}

```

# GET /api/status.wan.connection

## API

Obtain the WAN status. In fw 8.0.0, band, and signal are updated, the API support multiple band

Available in 8.0.0 or later

### Input Parameters

	Type	Notation	Mandatory	Description
<b>id</b>	Array	<numlist>	optional	list the WAN information base on id, multiple value is accepted, if id is absent, all WAN will be return
<b>lite</b>	String	{yes, no}	optional	Limited data within the connection will be get when the field set to 'yes' Otherwise, all status information will be got.

NOTE: This parameter will not have effect on MAX device.

### Return Parameters

#### Return JSON

	Type	Notation	Description
<b>order</b>	Array	list of <conn_id>	The order of connection by ID
<b>&lt;conn_id&gt;</b>	Object	<WAN_Status_Obj	WAN Status information

#### <WAN\_Status\_Obj>

	Type	Notation	Description
<b>name</b>	String	<string>	Name of the WAN connection
<b>statusLed</b>	String	{ empty, gray, red, yellow, green, flash }	LED color for UI
<b>asLan</b>	Boolean	<boolean>	WAN port is performing WAN as LAN or not
<b>enable</b>	Boolean	<boolean>	WAN is enabled or not
<b>locked</b>	Boolean	<boolean>	WAN is locked or not.
<b>scheduledOff</b>	Boolean	<boolean>	Only appear if Connection is scheduled and currently off
<b>message</b>	String	<string>	WAN status message
<b>uptime</b>	Number	<number>	Uptime in second
<b>type</b>	String	{ modem, wireless, gobi, cellular, ipsec, adsl, ethernet }	WAN connection type For cellular WAN In fw8.0.1 or later, it will return "cellular". Before fw8.0.1, it will return "gobi"
<b>virtualType</b>	String	{ modem, wireless, gobi, cellular, ipsec, adsl, ethernet }	WAN connection type For cellular WAN In fw8.0.1 or later, it will return "cellular". Before fw8.0.1, it will return "gobi"
<b>priority</b>	Number	<number>	Priority of the WAN. The field will not appear if the WAN is disabled
<b>groupSet</b>	Number	<number>	Group set of the WAN connection
<b>ip</b>	String	<ipv4>	IP address
<b>mask</b>	Number	<maskn>	Subnet mask. The field will not appear if ip is not exist or lite=yes
<b>gateway</b>	String	ipv4	Gateway. The field will not appear if ip is not exist or lite=yes
<b>method</b>	String	{ dhcp, static }	Connection method, DHCP or Static IP. The field will not appear if lite=yes
<b>mode</b>	String	{ NAT, IP Forwarding }	Connection mode. The field will not appear if lite=yes Please use routingMode in firmware 8.0.1 or later
<b>routingMode</b>	String	{ NAT, IP Forwarding }	Connection mode. The field will not appear if lite=yes

<b>dns</b>	Array	list of <ipv4>	DNS Server list. The field will not appear if lite=yes
<b>additionalIp</b>	Array	list of <ipv4>	Additional IP address list. The field will not appear if lite=yes
<b>mtu</b>	Number	[576, 9000]	MTU value. The field will not appear if auto or lite=yes
<b>mss</b>	Number	[536, 8960]	MSS value. This field will not appear if auto or lite=yes
<b>mac</b>	String	<mac>	MAC address. This field will not appear if lite=yes
<b>wireless</b>	Object	<Wifi_Obj>	WAN connection detail for wireless. The field will only appear if type is wifi
<b>modem</b>	Object	<Modem_Obj>	WAN connection detail for modem. The field will only appear if type is modem
<b>cellular</b>	Object	<Gobi_Obj>	WAN connection detail for gobi. The field will only appear if type is cellular
<b>gobi</b>	Object	<Gobi_Obj>	WAN connection detail for gobi. The field will only appear if type is gobi NOTE: This object is deprecated in firmware 8.0.1.
<b>bandwidthAllowanceMonitor</b>	Object	<BW_Allowance_Monitor_Obj>	Bandwidth allowance monitor. This field will not appear if auto or lite=yes

### <Wifi\_Obj>

	Type	Notation	Description
<b>signal</b>	Object	<Signal_Obj>	Signal information
<b>ssid</b>	String	<string>	SSID of the Wifi. The field will not appear if lite=yes
<b>bssid</b>	String	<string>	BSSID. The field will not appear if lite=yes

### <Modem\_Obj>

	Type	Notation	Description
<b>name</b>	String	<string>	Modem adaptor name
<b>vendorId</b>	Number	<integer>	Modem adaptor vendor ID
<b>productId</b>	Number	<integer>	Modem adaptor product ID
<b>manufacturer</b>	String	<string>	Modem adaptor manufacturer
<b>carrier</b>	Object	<Carrier_Obj>	Carrier Information
<b>signalLevel</b>	Number	[0, 5]	Signal level
<b>network</b>	String	<string>	Network name
<b>imsi</b>	String	<string>	International Mobile Subscriber Identity (IMSI). The field will not appear if lite=yes
<b>iccid</b>	String	<string>	Integrate circuit card identity (ICCID). The field will not appear if lite=yes
<b>esn</b>	String	<string>	Electronic Serial Number (ESN). The field will not appear if lite=yes
<b>mtn</b>	String	<string>	Mobile Telecommunications Network (MTN). The field will not appear if lite=yes
<b>apn</b>	String	<string>	APN. The field will not appear if lite=yes
<b>username</b>	String	<string>	Username for APN. The field will not appear if lite=yes
<b>password</b>	String	<string>	Password for APN. The field will not appear if lite=yes
<b>dialNumber</b>	String	<string>	Dial number for APN. The field will not appear if lite=yes
<b>band</b>	Array	list of <Band_Obj>	Cellular band information. Including Band Name and signal info

### <Gobi\_Obj>

	Type	Notation	Description
<b>roamingStatus</b>	Object	<Roaming_Obj>	Roaming status information
<b>network</b>	String	<string>	Network name  This information will be deprecated in fw8.0.1
<b>mobileType</b>	String	<string>	Network name As "network" is deprecated in fw8.0.1, please change the key to

use "mobileType" to get the information in fw8.0.1 or later

<b>sim</b>	Object	<SIM_Group_Obj>	SIM information
<b>remoteSim</b>	Object	<Remote_SIM_Obj>	Remote SIM information, this field will only appear when remote SIM is enable
<b>carrier</b>	Object	<Carrier_Obj>	Carrier information
<b>carrierAggregation</b>	Boolean	<boolean>	Carrier Aggregation
<b>signalLevel</b>	Number	[0, 5]	Signal level
<b>meid</b>	Object	<MEID_Obj>	Hex and Dec value of Mobile Equipment Identifier (MEID). The field will not appear if lite=yes
<b>imei</b>	String	<string>	International Mobile Equipment Identity (IMEI). The field will not appear if lite=yes
<b>esn</b>	String	<string>	Electronic Serial Number (ESN). The field will not appear if lite=yes
<b>mode</b>	String	<string>	Gobi network mode. The field will not appear if lite=yes
<b>band</b>	Array	list of <Band_Obj>	Gobi band information. Including Band Name and signal info  NOTE: This field will be obsoleted in fw 8.1.2, please use 'rat' to get the band information
<b>rat</b>	Array	list of <RAT_Obj>	Radio Access Technology support Available after fw 8.1.2
<b>mcc</b>	String	<string>	Three decimal digits as Mobile Country Code(MCC)
<b>mnc</b>	String	<string>	Two or Three decimal digits as Mobile Network Code(MNC)
<b>cellTower</b>	Object	<Cell_Tower_Obj>	Cell Tower information

#### <RAT\_Obj>

	Type	Notation	Description
<b>name</b>	String	<string>	RAT Name
<b>band</b>	Array	list of <Band_Obj>	Cellular band information

#### <Band\_Obj>

	Type	Notation	Description
<b>name</b>	String	<string>	Band Name
<b>channel</b>	Number	<integer>	Band Channel Avaliable after 8.1.2
<b>signal</b>	Object	<Signal_Obj>	Signal information

#### <Signal\_Obj>

	Type	Notation	Description
<b>rsSI</b>	Number	<number>	Received Signal Strength Indicator (RSSI), only appear in Gobi and Modem
<b>sinr</b>	Number	<number>	Signal to Interference plus Noise Ratio (SINR), only appear in Gobi and Modem
<b>snr</b>	Number	<number>	Signal-to-noise ratio (SNR), only appear in Gobi and has value
<b>ecio</b>	Number	<number>	Energy to Interference Ratio (Ec/Io), only appear in Gobi and has value
<b>rsrp</b>	Number	<number>	Reference Signal Received Power (RSRP), only appear in Gobi and Modem
<b>rsrq</b>	Number	<number>	Reference Signal Received Quality (RSRQ), only appear in Gobi
<b>strength</b>	Number	<number>	Wi-Fi signal strength, only appear in Wifi

#### <SIM\_Group\_Obj>

	Type	Notation	Description
<b>order</b>	Array	<numlist>	list of <sim_id>
<b>&lt;sim_id&gt;</b>	Object	<SIM_Obj>	SIM Information for SIM ID



## <Remote\_SIM\_Obj>

	Type	Notation	Description
<b>imsi</b>	String	<string>	-
<b>serialNumber</b>	String	<string>	-
<b>slot</b>	Number	<number>	Number of slot
<b>autoApn</b>	Boolean	<boolean>	Indicate the APN, Username and Password fields are auto detect or custom values Only available in fw8.1.1 or later
<b>apn</b>	String	<string>	APN. The field will not appear if lite=yes Only available in fw8.1.1 or later
<b>username</b>	String	<string>	Username for APN. The field will not appear if lite=yes Only available in fw8.1.1 or later
<b>password</b>	String	<string>	Password for APN. The field will not appear if lite=yes Only available in fw8.1.1 or later

## <Carrier\_Obj>

	Type	Notation	Description
<b>name</b>	String	<string>	Carrier name
<b>country</b>	String	<string>	Carrier country. The field will not appear if lite=yes

## <MEID\_Obj>

	Type	Notation	Description
<b>hex</b>	String	<string>	MEID value in HEX
<b>dec</b>	String	<string>	MEID value in DEC

## <SIM\_Obj>

	Type	Notation	Description
<b>status</b>	String	{ In Use, SIM Card Detected, No SIM Card Detected }	SIM card status
<b>active</b>	Boolean	<boolean>	SIM card active status
<b>apn</b>	String	<string>	APN. The field will not appear if lite=yes
<b>username</b>	String	<string>	Username for APN. The field will not appear if lite=yes
<b>password</b>	String	<string>	Password for APN. The field will not appear if lite=yes
<b>imsi</b>	String	<string>	International Mobile Subscriber Identity (IMSI). The field will not appear if lite=yes
<b>iccid</b>	String	<string>	Integrate circuit card identity (ICCID). The field will not appear if lite=yes
<b>mtn</b>	String	<string>	Mobile Telecommunications Network (MTN). The field will not appear if lite=yes

## <Roaming\_Obj>

	Type	Notation	Description
<b>code</b>	Number	{ 0, 1, 2 }	Romaing Status Code
<b>message</b>	String	{ roaming, home, roaming partner }	Readable Roaming Status Code and message relation: 0 - roaming 1 - home 2 - roaming partner

## <Cell\_Tower\_Obj>

	Type	Notation	Description
<b>cellId</b>	Number	<number>	Cell ID of the each base transceiver status
<b>cellPlmn</b>	Number	<number>	Cell Public Land Mobile Network (Cell PLMN) of the tower

<b>cellUtranId</b>	Number	<number>	Cell UTRAN ID
<b>tac</b>	Number	<number>	Tracking Area Code for LTE network
<b>lac</b>	Number	<number>	Location Area Code for GSM/UMTS network

### <BW\_Allowance\_Monitor\_Obj>

	Type	Notation	Description
<b>enable</b>	Boolean	<boolean>	Bandwidth Allowance enable
<b>hasSntp</b>	Boolean	<boolean>	Email notification is enabled or not
<b>action</b>	Array	list of {email, disconnect, restrict}	Action will take when reach the allowance limit email - send the email, disconnect - disconnect the WAN connection restrict - allow traffic to hostname peplink.com and user defined ICA host only for management purpose
<b>start</b>	Number	[0, 28]	Start date of the allowance monitor. When the value is '0', that means the start day is the last day of that month
<b>monthlyAllowance</b>	Object	<Monthly_Allowance_Obj>	Monthly Allowance Information

### <Monthly\_Allowance\_Obj>

	Type	Notation	Description
<b>value</b>	Number	<integer>	Monthly Allowance Limit
<b>unit</b>	String	{MB}	The unit for 'value'.

### cURL Example

```
> curl -b cookies.txt http://192.168.1.1/api/status.wan.connection?id=1 2
```

```
{
  "stat": "ok",
  "response": {
    "1": {
      "name": "CUST WAN 1",
      "enable": true,
      "asLan": false,
      "message": "Connected",
      "uptime": 27037017,
      "type": "ethernet",
      "virtualType": "ethernet",
      "priority": 0,
      "ip": "192.168.123.144",
      "statusLed": "green",
      "mask": 24,
      "gateway": "12.23.34.0",
      "method": "dhcp",
      "mode": "NAT",
      "dns": [
        "12.22.32.12",
        "12.34.67.89"
      ],
      "mtu": 576
    },
    "2": {
      "name": "WAN2",
      "enable": true,
      "asLan": false,
      "message": "No Cable Detected",
      "uptime": 27066417,
      "type": "ethernet",
      "virtualType": "ethernet",
      "priority": 0,
      "statusLed": "red",
    }
  }
}
```

```

        "method": "static",
        "mode": "IP Forwarding",
        "mtu": 1440
    },
    "order": [
        1,
        2
    ]
}
}

```

## GET /api/status.wan.connection.allowance

### API

Obtain the bandwidth allowance of the WAN connection or SIM

Available in 8.0.0 or later

### Input Parameters

	Type	Notation	Mandatory	Description
<b>connId</b>	Array	<numlist>	optional	<p>Connection ID</p> <p>In firmware 8.0.0, this field is mandatory and ONLY cellular WAN is allowed API user needs to provide the ID to obtain the information</p> <p>In firmware 8.0.1, this field is optional and allow any type of WAN when the field is absent, all WAN connection bandwidth allowance monitor information will be got.</p>

### Return Parameters

#### Return JSON

	Type	Notation	Description
<b>&lt;conn_id&gt;</b>	Object	{<SIM_Allowance_Obj>, <Allowance_Obj>}	<p>In firmware 8.0.0, only cellular WAN is supported, It will return &lt;SIM_Allowance_Obj&gt; for the allowance monitor.</p> <p>In firmware 8.0.1 or later, all WAN type is supported, it will return &lt;Allowance_Obj&gt; if that is not cellular WAN.</p> <p>In firmware 8.0.1 or later, the output of Cellular WAN will same as firmware 8.0.0</p>
<b>order</b>	Array	list of <conn_id>	WAN Connection ID order reference

#### <SIM\_Allowance\_Obj>

	Type	Notation	Description
<b>&lt;sim_id&gt;</b>	Object	<Allowance_Obj>	Allowance status
<b>order</b>	Array	list of <sim_id>	SIM ID order reference

#### <Allowance\_Obj>

	Type	Notation	Description
<b>enable</b>	Boolean	<boolean>	-
<b>usage</b>	Number	<integer>	Data used in MB
<b>limit</b>	Number	<integer>	Monthly allowance in MB
<b>percent</b>	Number	[ 0, 100 ]	Percentage of the usage
<b>start</b>	Number	[ 0, 28 ]	Start day of the allowance, 0 mean the last day of the month
<b>unit</b>	String	{ MB }	-

## cURL Example

```
> curl -b cookies.txt http://192.168.1.1/api/status.wan.connection.allowance?connId=1
```

```
{
  "stat": "ok",
  "response": {
    "1": {
      "1": {
        "enable": true,
        "usage": 5,
        "limit": 1024,
        "percent": 0,
        "start": 1,
        "unit": "MB"
      }
    },
    "order": [
      1
    ]
  }
}
```

## GET /api/status.wan.connection.signal

API beta

Obtain cellular, modem and Wi-Fi WAN signal information.  
When no filter is apply, all WAN will be shown.  
If the WAN is not cellular, modem or Wi-Fi, null will be show.

*Available in 8.0.1 or later*

### Input Parameters

	Type	Notation	Mandatory	Description
<b>connId</b>	Array	<numlist>	optional	Connection ID
<b>virtualType</b>	Array	list of { wireless, cellular, modem }	optional	Filter of the virtual type
<b>infoType</b>	Array	list of { virtualType, signal, activeSim, wifi, band, signalLevel }	optional	The information section in the return object

### Return Parameters

#### Return JSON

	Type	Notation	Description
<b>&lt;conn_id&gt;</b>	Object		The signal information. The object will also provide some reference information. Virtual Type is modem will use the <Cellular_Type_Signal_Obj>
<b>order</b>	Array	list of <conn_id>	WAN Connection ID order reference

#### <Wifi\_Type\_Signal\_Obj>

	Type	Notation	Description
<b>virtualType</b>	String	{ wireless }	Virtual type of the WAN connection
<b>wifi</b>	Object	<Wifi_Obj>	Wi-Fi information. SSID and the security policy
<b>signal</b>	Object	<Wifi_Signal_Obj>	Wi-Fi Signal

## <Wifi\_Obj>

	Type	Notation	Description
<b>ssid</b>	String	<string>	SSID
<b>securityPolicy</b>	String	{ open, wep, wpa, wpa-eap, wpa-psk, 8021x }	Security policy of the Wi-Fi connection

## <Wifi\_Signal\_Obj>

	Type	Notation	Description
<b>strength</b>	Number	<number>	Signal strength of the Wi-Fi signal in dBm
<b>level</b>	Number	[ 0, 5 ]	Signal Level of the Wi-Fi signal for the signal indicator bar.

## <Cellular\_Type\_Signal\_Obj>

	Type	Notation	Description
<b>virtualType</b>	String	{ modem, cellular }	Virtual type of the WAN connection
<b>activeSim</b>	Object	<Active_SIM_Obj>	The active SIM information of the cellular
<b>band</b>	Array	list of <Band_Signal_Obj>	The signal information of the cellular

## <Active\_SIM\_Obj>

	Type	Notation	Description
<b>carrierName</b>	String	<string>	Carrier name of the active SIM
<b>network</b>	String	{ 2G, 3G, LTE }	Carrier Network of the active SIM

## <Band\_Signal\_Obj>

	Type	Notation	Description
<b>name</b>	String	<string>	Band name
<b>signal</b>	Object	<Cellular_Signal_Obj>	Signal information. The field inside this object will not be shown when the information is missing.

## <Cellular\_Signal\_Obj>

	Type	Notation	Description
<b>rsi</b>	Number	<number>	RSSI
<b>sinr</b>	Number	<number>	SINR
<b>snr</b>	Number	<number>	SNR
<b>ecio</b>	Number	<number>	ECIO
<b>rsrp</b>	Number	<number>	RSRP
<b>rsrq</b>	Number	<number>	RSRQ
<b>rscp</b>	Number	<number>	RSCP

## cURL Example

```
> curl -b cookies.txt http://192.168.1.1/api/status.wan.connection.signal?connId=1 2&infoType=band
```

```
{
  "stat": "ok",
  "response": {
    "1": {
      "band": [
        {
          "name": "LTE Band 7 (2600 MHz)",
          "signal": {
            "rsi": -60,
            "sinr": 9,

```

```
        "rsrp": -101
      }
    ]
  },
  "2": {
    "band": [
      {
        "name": "LTE Band 7 (2600 MHz)",
        "signal": {
          "rssi": -90,
          "sinr": 6.6,
          "rsrp": -100,
          "rsrq": -3
        }
      },
      {
        "name": "LTE Band 7 (3500 MHz)",
        "signal": {
          "rssi": -55
        }
      }
    ]
  },
  "order": [
    1,
    2
  ]
}
}
```