

## 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 notice to inform when the API is undocumented (for experimental / beta), or when it is in deprecate state or already replace with another API endpoint.

### 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.config.apply
- POST cmd.config.discard
- POST cmd.sms.sendMessage
- POST cmd.system.reboot
- GET cmd.ap
- POST cmd.ap
- POST cmd.cellularModule.rescan
- 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
- GET config.ssid.profile
- POST config.ssid.profile
- POST config.wan.connection.priority
- GET info.firmware
- GET status.lan.profile
- GET status.pepvpn
- 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	

	Type	Notation	Mandatory	Description
<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
clientId	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
accessToken	String	<hash>	Access token
clientId	String	<hash>	Client ID
clientName	String	<string>	Client Name
authorizationType	Number	{ 3 }	Authorization type. Always get 3 for client credentials grant
scope	String	{ api, api.read-only }	The scope of the access token
createTimestamp	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
clientId	String	<hash>	require	Client ID
clientSecret	String	<hash>	require	Client Secret
scope	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.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.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
http://192.168.1.1/api/cmd.sms.sendMessage
```

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

## POST /api/cmd.system.reboot

API

Reboot the system

Available in 7.1.1 or later

### Input Parameters

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

	Type	Notation	Mandatory	Description
<b>firmwareId</b>	Number	<number>	optional	reboot with which firmware ID If the field is absent, the current firmware will be booted up.

### cURL Example

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

```
{
  "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.rescan

### API

Rescan the cellular module of the corresponding WAN connection

Available in 8.0.0 or later

### Input Parameters

	Type	Notation	Mandatory	Description
<b>connId</b>	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.rescan
```

```
{
  "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
<b>connId</b>	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
connId	Number	<int>	require	WAN Connection of the cellular module

### Return Parameters

#### Return JSON

	Type	Notation	Description
enabledSim	Array	list of {1, 2}	SIMs to be used (1 for SIM Slot A and 2 for SIM Slot B)
preferredSim	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
connId	Number	<int>	require	WAN Connection of the cellular module
enabledSim	Array	list of {1, 2}	optional	Choice of SIMs to be used (1 for SIM Slot A and 2 for SIM Slot B)
preferredSim	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

## Return JSON

	Type	Notation	Description
<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
<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
      }
    }
  ]
}
```

## 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>	order	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

## <SSID\_Profile\_Obj>

	Type	Notation	Description
<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.

Availiable 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>	order	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.priority

### API

Change the priority of the WAN connection

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	If will not active instantly if the field is absent.

## <WAN\_Config\_Priority\_Obj>

	Type	Notation	Mandatory	Description
<b>&lt;WAN_Config_Priority_Obj&gt;</b>				
<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
version	String	<string>	Firmware version
bootable	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/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>

<b>&lt;Profile_Obj&gt;</b>			
	<i>Type</i>	<i>Notation</i>	<i>Description</i>
<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, natsc }	Type of the profile. The field will not appear if lite=yes

### **<Peer\_Obj>**

	<i>Type</i>	<i>Notation</i>	<i>Description</i>
<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, natsc }	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>**

	<i>Type</i>	<i>Notation</i>	<i>Description</i>
<b>order</b>	Array	list of <peer_id>	Order of the peer ID
<b>&lt;peer_id&gt;</b>	Object	<WAN_Order_Obj>	Tunnel information by peer ID

### **<WAN\_Order\_Obj>**

	<i>Type</i>	<i>Notation</i>	<i>Description</i>
<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>**

	<i>Type</i>	<i>Notation</i>	<i>Description</i>
<b>id</b>	Number	<int>	WAN connection ID



## <WAN\_Obj>

	Type	Notation	Description
<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>rtt</b>	Number	<number>	Round trip delay time of the remote peer WAN
<b>rx</b>	Array	list of <number>	Receive bytes of the remote peer WAN
<b>tx</b>	Array	list of <number>	Transmit bytes of the remote peer WAN
<b>loss</b>	Array	list of <number>	Package loss of the remote peer WAN
<b>remote</b>	Object	<WAN_Order_Obj>	Remote WAN tunnel status This field only appear in local tunnel information

## <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.pepvpn?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": {  
        "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
    ]
},
"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"
    }
]
}
}

```

## 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, ipsec, adsl, ethernet }	Connection type of the WAN
<b>virtualType</b>	String	{ modem, wireless, gobi, ipsec, adsl, ethernet }	Connection type of the WAN
<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
<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

## <WAN\_Status\_Obj>

	Type	Notation	Description
<b>gobi</b>	Object	<Gobi_Obj>	WAN connection detail for gobi. The field will only appear

## <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
<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>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
<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

## <Band\_Obj>

	Type	Notation	Description
<b>name</b>	String	<string>	Band Name
<b>signal</b>	Object	<Signal_Obj>	Signal information

## <Signal\_Obj>

	Type	Notation	Description
<b>rsi</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

## <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

## <SIM\_Obj>

	Type	Notation	Description
<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	{ home, roaming, roaming partner }	Readable Roaming Status

## <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

## 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, connId must Cellular connection

Available in 8.0.0 or later

### Input Parameters

	Type	Notation	Mandatory	Description
connId	Array	<numlist>	require	Connection ID

### Return Parameters

#### Return JSON

	Type	Notation	Description
<conn_id>	Object	<SIM_Allowance_Obj>	Allowance for the SIM
order	Array	list of <conn_id>	WAN Connection ID order reference

#### <SIM\_Allowance\_Obj>

	Type	Notation	Description
<sim_id>	Object	<Allowance_Obj>	Allowance status
order	Array	list of <sim_id>	SIM ID order reference

#### <Allowance\_Obj>

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



---

### <Cellular\_Type\_Signal\_Obj>

	<i>Type</i>	<i>Notation</i>	<i>Description</i>
<b>band</b>	Array	list of <Band_Signal_Obj>	The signal information of the cellular

---

### <Active\_SIM\_Obj>

	<i>Type</i>	<i>Notation</i>	<i>Description</i>
<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>

	<i>Type</i>	<i>Notation</i>	<i>Description</i>
<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>

	<i>Type</i>	<i>Notation</i>	<i>Description</i>
<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=signal  
band
```

---