



Operating instructions (supplement)
REST API - Commands

GB

AE9000
AE9001

Contents

1	Preliminary note	3
1.1	Purpose of the document	3
1.2	Legal and copyright information	3
1.3	Symbols used	3
1.4	Warnings.	3
1.6	Change history	4
2	Authentication	5
3	REST commands	6
3.1	Kommando readconfigtree	6
3.2	Kommando writeconfigtree	8
3.3	Kommando backup	11
3.4	Kommando restore	12
3.5	Kommando factoryreset	13
3.6	Kommando info.	14
3.7	Kommando log	17
3.8	Kommando sysloginfo.	18
3.9	Kommando reboot	19
3.10	Kommando ulmp.	20
3.11	Kommando update	21
3.12	Kommando dhcpleases.	22
3.13	Kommando simcard	23
3.14	Kommando position	27
3.15	Kommando openvpnconfig	28
3.16	Kommando openvpnlog	29
3.17	Kommando openvpnstat	30
3.18	Kommando sms	31

1 Preliminary note

You will find instructions, technical data, approvals and further information using the QR code on the unit / packaging or at documentation.ifm.com.

1.1 Purpose of the document

This document describes the available commands for REST API of the Industrial LTE 4G router.

This document is intended for persons who are familiar with the product and have specialised knowledge of wireless communication and network technology.

1.2 Legal and copyright information

© All rights reserved by ifm electronic gmbh. No part of these instructions may be reproduced and used without the consent of ifm electronic gmbh.

All product names, pictures, companies or other brands used are the property of the respective rights owners.

- Manufacturer:

HMS Industrial Networks AB
Stationsgatan 37
302 50 Halmstad
Schweden

- Sales:

ifm electronic gmbh
Friedrichstraße 1
45128 Essen
Deutschland

1.3 Symbols used

- ✓ Requirement
- Instructions
- ▷ Reaction, result
- [...] Designation of keys, buttons or indications
- Cross-reference
-  Important note
Non-compliance may result in malfunction or interference.
-  Information
Supplementary note

1.4 Warnings

Warnings indicate the possibility of personal injury and damage to property. This enables safe product handling. Warnings are graded as follows:



WARNING

Warning of serious personal injury

- ▷ If the warning is not observed, fatal and serious injuries are possible.

**CAUTION**

Warning of minor to moderate personal injury

▷ If the warning is not observed, minor to moderate injuries are possible.

ATTENTION

Warning of damage to property

▷ If the warning is not observed, damage to property is possible.

1.5 Exclusion of liability

The information contained in this document is for information purposes only. Please inform ifm electronic of any inaccuracies or omissions found in this document. ifm electronic disclaims any responsibility or liability for any errors that may appear in this document.

ifm electronic reserves the right to modify its products in line with its policy of continuous product development. The information in this document shall therefore not be construed as a commitment on the part of ifm electronic and is subject to change without notice. ifm electronic makes no commitment to update or keep current the information contained in this document.

The data, examples and illustrations found in this document are included for illustrative purposes and are only intended to help improve understanding of the functionality and handling of the product. In view of the wide range of possible applications of the product, and because of the many variables and requirements associated with any particular implementation, ifm electronic cannot assume responsibility or liability for actual use based on the data, examples or illustrations included in this document nor for any damages incurred during installation of the product. Those responsible for the use of the product must acquire sufficient knowledge in order to ensure that the product is used correctly in their specific application and that the application meets all performance and safety requirements including any applicable laws, regulations, codes and standards. Furthermore, ifm electronic will under no circumstances assume liability or responsibility for any problems that may arise as a result from the use of undocumented features or functional side effects found outside the documented scope of the product. The effects caused by any direct or indirect use of such aspects of the product are undefined and may include e.g. compatibility issues and stability issues.

1.6 Change history

Issue	Subject	Date
00	New creation of the document	12 / 2024

2 Authentication

All HTTP requests to the REST API must be authenticated with the "Basic HTTP Authentication Scheme" (RFC7617).

Die Basic-Authentifizierung erfolgt durch Hinzufügen eines Header-Parameters zur HTTP-Anfrage:

```
Authorization: Basic <Base64 encoding of user-id:password>
```

Die Benutzerkennung lautet immer "admin" und das Passwort ist das derzeit verwendete Web-Passwort, entweder das mit dem Produkt gelieferte Standardpasswort oder, falls es geändert wurde, das neue Passwort.

Example:

```
User-id: admin  
Password: abc123  
String to Base64-encode: admin:abc123  
Completed header parameter:  
Authorization: Basic YWRtaW46YWJjMTIz
```

3 REST commands

3.1 Kommando readconfigtree

Informationen aus einem Konfigurationsbaum lesen und als JSON-Datum ausgeben

3.1.1 Syntax

```
GET <target_ip>/cgi-bin/readconfigtree.cgi?path=<tree_path>
```

3.1.2 Request parameters

Name	Description
target_ip	IP-Adresse des Geräts (z. B. 192.168.0.98)
tree_path	Pfad zu einem Knoten des Konfigurationsbaums

3.1.3 Return Values

JSON-Struktur mit den angeforderten Daten.

3.1.4 Examples

Example: Einen einzelnen Konfigurationsknoten lesen

- Request:

```
GET http://<target_ip>/cgi-bin/readconfigtree.cgi?path=web:/settings/lock
```

- Response:

```
{
  "name": "lock",
  "type": "bool",
  "value": true
}The example ...
```

Example: Einen Konfigurationsbaum lesen

- Request:

```
GET http://<target_ip>/cgi-bin/readconfigtree.cgi?path=web:
```

- Response:

```
{
  "name": "",
  "type": "tree",
  "children": [{
    "name": "settings",
    "type": "stem",
    "children": [{
      "name": "lock",
      "type": "bool",
      "value": true
    }]
  }]
}
```

3.2 Kommando writeconfigtree

Informationen in einen Konfigurationsbaum schreiben

3.2.1 Syntax

```
GET <target_ip>/cgi-bin/writeconfigtree.cgi?path=<tree_path>&value=<new_value>
```

3.2.2 Request parameters

Name	Description
target_ip	IP-Adresse des Geräts (z. B. 192.168.0.98)
tree_path	Pfad zu einem Knoten des Konfigurationsbaums
new_value	Value that is to be written to the output

3.2.2.1 Konfigurationspfade

Verfügbare Konfigurationspfade:

Path	Description
router:/settings/ipAddr	IP address
router:/settings/netMask	Subnet mask
router:/settings/dhcpEnable	DHCP-Server aktivieren (true) oder deaktivieren (false)
router:/settings/auth	APN-Authentifizierung aktivieren (true) oder deaktivieren (false)
router:/settings/authUser	APN authentication User
router:/settings/authPass	APN authentication Password
router:/settings/dhcpRangeStart	DHCP-Startadresse: 1...254
router:/settings/dhcpRangeStop	DHCP-Startadresse: >254
router:/settings/apn	Access Point Name (APN)
router:/settings/rat	radio access technology (RAT) auto, lte-m1+gsm, lte-m1+lte-nb1, lte-nb1+gsm, lte-nb1+gsm, lte-m1, lte-nb1 oder gsm
router:/settings/ratOrder	Liste mit Modemvoreinstellungen, RAT-Suchreihenfolge: 6 Kombinationen aus LTE-M1, LTE-NB1 und GSM (IoT only)
router:/settings/natMode	NAT mode: disabled, nat11
router:/settings/natTarget	Lokale Zieladresse für NAT1:1
router:/settings/natSourceFilter	NAT – Quellenfilter: any, host, network, range
router:/settings/natSourceIp	NAT - Quellen-IPs: Kommagetrennte Liste von bis zu 5 IPs (nur anwendbar, wenn der Filter host ist)
router:/settings/natSourceBaseIp	NAT - Quellen-Basis-IP (nur anwendbar, wenn der Filter network ist)
router:/settings/natSourceMask	NAT - Quellen-Netzmaske (nur anwendbar, wenn der Filter network ist)
router:/settings/natSourceRangeStart	NAT - Quellen-IP-Bereich Anfang (nur anwendbar, wenn der Filter range ist)
router:/settings/natSourceRangeStop	NAT - Quellen-IP-Bereich Ende (nur anwendbar, wenn der Filter range ist)
router:/settings/openvpnEnable	OpenVPN Client aktivieren (true) oder deaktivieren (false)
web:/settings/lock	Konfigurationssperre aktivieren (true) oder deaktivieren (false)
web:/settings/password	Passwort für Web und REST API (gültige Zeichen: a-z, A-Z, 0-9, +, _)
position:/settings/enable	Positionierungsdienst aktivieren (true) oder deaktivieren (false)

► Um die neuen Einstellungen zu übernehmen und zu nutzen, das Gerät neu starten.

3.2.3 Return Values

```
{
  "success":<value>
}
```

value	Description
true	Data transfer completed successfully.
false	Anfrage fehlgeschlagen oder Validierungsfehler oder Konfiguration gesperrt

3.2.4 Examples

Example:

- Request:

```
GET http://<target_ip>/cgi-bin/writeconfigtree.cgi?path=router:/settings/
apn&value=my_apn.company.com
```

- Response:

```
{
  "success":true
}
```

Example:

- Request:

```
http://192.168.0.99/cgi-bin/writeconfigtree.cgi?path=router:/settings/
ratOrder&value=GSM,LTE-NB1,LTE-M1
```

- Response:

```
{
  "success":true
}
```

Example:

- Request:

```
GET http://<target_ip>/cgi-bin/writeconfigtree.cgi?path=router:/settings/
natSourceFilterMode&value=host
```

- Response:

```
{
  "success":true
}
```

Example:

- Request:

```
GET http://<target_ip>/cgi-bin/writeconfigtree.cgi?path=router:/settings/
natSourceIp&value=10.10.0.1,10.20.0.1,10.30.0.1,10.40.0.1,10.50.0.1
```

- Response:

```
{  
  "success":true  
}
```

3.3 Kommando backup

Konfiguration des Geräts in einer Datei sichern

3.3.1 Syntax

```
GET <target_ip>/cgi-bin/backup.cgi
```

3.3.2 Query Parameters

None

3.3.3 Examples

Example:

- Request:

```
GET http://<target_ip>/cgi-bin/backup.cgi
```

- Response:

```
{  
  "success":true  
}
```

3.4 Kommando restore

Konfiguration des Geräts aus einer Datei wiederherstellen

3.4.1 Syntax

```
POST http://<target_ip>/cgi-bin/restore.cgi
```

3.4.2 Query Parameters

None

3.4.3 Anfragedaten

Inhalte der Sicherungsdatei

3.4.4 Return Values

```
{
  "success":<value>
}
```

value	Description
true	Data transfer completed successfully.
false	Anfrage fehlgeschlagen oder Validierungsfehler oder Konfiguration gesperrt

3.4.5 Examples

Example:

- Request:

```
POST http://<target_ip>/cgi-bin/restore.cgi
```

- Response:

```
{
  "success":true
}
```

3.5 Kommando factoryreset

Reset device to factory settings

3.5.1 Syntax

```
GET <target_ip>/cgi-bin/factoryreset.cgi
```

3.5.2 Query Parameters

None

3.5.3 Return Values

```
{
  "success":<value>
}
```

value	Description
true	Data transfer completed successfully.
false	Anfrage fehlgeschlagen oder Konfiguration gesperrt

3.5.4 Examples

Example:

- Request:

```
GET http://<target_ip>/cgi-bin/factoryreset.cgi
```

- Response:

```
{
  "success":true
}
```

3.6 Kommando info

Call up fieldbus information

3.6.1 Syntax

```
GET <target_ip>/cgi-bin/info.cgi
```

3.6.2 Query Parameters

None

3.6.3 Return Values

JSON-Struktur mit den folgenden Informationen:

Value	Description
uptime	Operating time of device
time	read system time of the device
radio_power	Stromversorgungszustand des Funkmoduls 0: Off 1: On
sim	the SIM card operator
signal_strength	Signal strength Value range: 0...5
signal_strength_raw	Signalstärke (raw CSQ); Value range: 0...31
signal_quality	Signalqualität (in dBm)
cell_id	Identifikator der Funkzelle
lac	Location Area Code der Funkzelle
tac	Tracking Area Code der Funkzelle (nur LTE)
rat	Aktuell genutzte Funkzugangstechnologie 0: Unknown 1: GSM 2: UTMS 3: LTE
operator	Aktuell genutzter Betreiber
status	network registration status 0: Nicht registriert und nicht suchend 1: registered with home network 2: Nach neuem Betreiber suchend 3: Registrierung abgelehnt 4: Registrierung am Roaming-Netzwerk 5: Unknown status
amplifier_temp	the temperature of the amplifier (value in °C)
controller_temp	Temperatur des Controllers (in °C)
connection_state	Status der Datensitzung 0: not connected 1: Authentifizierung läuft 2: Connected 3: suspended 4: Eingehend (MT-PDP context request)
voltage	Voltage (in mV)

Value	Description
iotbolt_version	Firmware version of the device
modem_model	Modellname des Modems
modem_version	modem firmware version
pri	carrier name and version
apn	Aktuell genutzter APN
rat_specific	Zugriffstechnologie der Funkzelle • 0: GSM • 1: GSM Compact • 2: UTRAN • 3: GSM mit EGPRS • 4: UTRAN mit HSDPA • 5: UTRAN mit HSUPA • 6: UTRAN mit HSDPA und HSUPA • 7: CAT-M1 • 8: GSM-IoT • 9: NB- IoT , 2G
imsi	international mobile subscriber identity
imei	international mobile equipment identity
cellular_gateway	IP address of the gateway of the cellular interface
cellular_ip	ID der Funkzelle
iccid	ICCID number of the SIM card
active_band	Genutztes Frequenzband

3.6.4 Examples

Example:

- Request:

```
GET http://<target_ip>/cgi-bin/info.cgi
```

- Response:

```
{
  "uptime": "4287",
  "time": "2019-11-19 13:35:48",
  "radio_power": "1",
  "sim": "2",
  "signal_strength": "4",
  "signal_strength_raw": "23",
  "signal_quality": "1",
  "cell_id": "26269452",
  "lac": "4294967295",
  "tac": "0x85",
  "rat": "4",
  "operator": "TELIA S",
  "status": "4",
  "amplifier_temp": "35",
  "controller_temp": "35",
  "connection_state": "2",
  "voltage": "3807",
  "iotbolt_version": "1.00.20-dev",
  "modem_model": "WP7607",
  "modem_version": "SWI9X06Y_02.16.06.00",
  "pri": "GENERIC_001.028_001",
  "apn": "",
  "rat_specific": "9",
  "imsi": "238208700452254",
  "imei": "352653090129735",
  "cellular_ip": "10.81.177.33",
  "iccid": "89450421170203490357",
  "active_band": "B20"
}
```

3.7 Kommando log

System-Log-Datei anzeigen

3.7.1 Syntax

```
GET <target_ip>/cgi-bin/log.cgi
```

3.7.2 Query Parameters

None

3.7.3 Return Values

Log-Datei im Klartext

3.7.4 Examples

Example:

- Request:

```
GET http://<target_ip>/cgi-bin/log.cgi
```

- Response:

```
Oct 9 10:03:18 Legato: INFO | Version: 1.00.18-dev
Oct 9 10:03:18 Legato: INFO | Boot reason: Power-on
Jan 6 00:00:23 Legato: INFO | eth0 IP address: 192.168.0.98 - netmask:
255.255.254.0
Jan 6 00:00:23 Legato: INFO | eth0 DHCP server: on
Jan 6 00:00:23 Legato: INFO | APN auth is: off
Jan 6 00:00:23 Legato: INFO | DHCP range start - stop: 192.168.0.100 -
192.168.0.200
Jan 6 00:00:24 Legato: INFO | IP address/netmask set to:
192.168.0.98/255.255.254.0
Jan 6 00:00:24 Legato: INFO | Setting no auth for APN
Jan 6 00:00:24 Legato: INFO | Setting configured RAT: "gsm": 0x01
Jan 6 00:00:24 Legato: INFO | Data connection requested
Oct 17 12:02:43 Legato: INFO | Data connected, interface 'rmnet_data0'
Oct 17 12:02:43 Legato: INFO | Restarting DHCP/DNS services
```

3.8 Kommando sysloginfo

Ein Archiv mit allen System-Log-Dateien erzeugen

3.8.1 Syntax

```
GET <target_ip>/cgi-bin/sysloginfo.cgi
```

3.8.2 Query Parameters

None

3.8.3 Examples

Example:

- Request:

```
GET http://<target_ip>/cgi-bin/sysloginfo.cgi
```

- Response:

```
<octet stream containing the archive>
```

3.9 Kommando reboot

Rebooting the device.

3.9.1 Syntax

```
GET <target_ip>/cgi-bin/reboot.cgi
```

3.9.2 Query Parameters

None

3.9.3 Return Values

```
{
  "success":<value>
}
```

value	Description
true	Data transfer completed successfully.
false	Anfrage fehlgeschlagen

3.9.4 Examples

Example:

- Request:

```
GET http://<target_ip>/cgi-bin/reboot.cgi
```

- Response:

```
{
  "success":true
}
```

3.10 Kommando ulmp



Der Industrial LTE 4G Router US (Artikelnummer: AE9001) unterstützt das REST-Kommando `ulmp` nicht.

Das Gerät für eine bestimmte Dauer in den Ultra Low Power Mode (ULPM) versetzen. Nach Ablauf der definierten Zeitspanne wird das Gerät wieder in den normalen Betriebsmodus versetzt.

3.10.1 Syntax

```
GET <target_ip>/cgi-bin/ulpm.cgi?duration=<time in seconds>
```

3.10.2 Query Parameters

Name	Description
duration	Dauer, für die das Gerät in ULPM versetzt werden soll (Wert in s); Value range: 60...86400

3.10.3 Return Values

```
{
  "success":<value>,
  "message": "<additional info>"
}
```

value	Description
true	Data transfer completed successfully.
false	Anfrage fehlgeschlagen oder Validierungsfehler

3.10.4 Examples

Example: Ultra Low Power Mode aktivieren und Sleep-Intervall auf 300 s setzen (5 min)

- Request:

```
GET http://<target_ip>/cgi-bin/ulpm.cgi?duration=300
```

- Response:

```
{
  "success":true,
  "message":"sleeping for 300 s"
}
```

3.11 Kommando update

Update the firmware of the device

3.11.1 Syntax

```
POST http://<target_ip>/cgi-bin/update.cgi
```

3.11.2 Query Parameters

None

3.11.3 Anfragedaten

Inhalt der Firmware-Datei (.fws)

3.11.4 Return Values

Ausgabe des Status des Aktualisierungsprozesses im HTML-Format.

- Um Ergebnis zu ermitteln, nach Zeichenketten "SUCCESS" (erfolgreich) und "FAILED" (fehlgeschlagen) suchen.

3.11.5 Examples

Example:

- Request:

```
POST http://<target_ip>/cgi-bin/update.cgi
```

- Response:

```
Trying to update...Unpacking: 0% FAILED: error code 2
```

3.12 Kommando dhcpleases

Rückgabe einer Liste mit DHCP-Leases

3.12.1 Syntax

```
GET <target_ip>/cgi-bin/dhcpleases.cgi
```

3.12.2 Query Parameters

None

3.12.3 Return Values

Textausgabe der DHCP-Leases im Format dnsmasq.leases.

3.12.4 Examples

Example:

- Request:

```
GET http://<target_ip>/cgi-bin/dhcpleases.cgi
```

- Response:

```
1573275080 00:e0:4c:34:92:e9 192.168.0.168 LT-5Q1XWT2 01:00:e0:4c:34:92:e9
```

3.13 Kommando simcard

Abfrage des Status der PIN und PUK der SIM-Karte oder das Resultat einer Eingabe der PIN oder PUK der SIM-Karte.

Die eingegebene und validierte PIN wird bei allen nachfolgenden Neustarts genutzt, solange dieselbe SIM-Karte im Gerät eingelegt ist.

Bei der Entsperrung der SIM-Karte mit einer PUK muss eine neue PIN eingegeben werden. Die neue PIN wird bei allen nachfolgenden Neustarts genutzt, solange dieselbe SIM-Karte im Gerät eingelegt ist.

3.13.1 Syntax

Status der PIN und PUK abfragen:

```
GET http://<target_ip>/cgi-bin/simcard.cgi?status=
```

SIM-Karte mit PIN validieren:

```
GET http://<target_ip>/cgi-bin/simcard.cgi?pin=12345678
```

SIM-Karte mit PUK entsperren und neue PIN setzen:

```
GET http://<target_ip>/cgi-bin/simcard.cgi?pin=12345678&puk=87654321
```

3.13.2 Query Parameters

Name	Description
status	Status of the enquiry
pin	PIN number of the SIM card
puk	PUK number of the SIM card

3.13.3 Return Values

```
{
  "success":<value>,
  "needs-pin":<pin validation status>,
  "needs-puk":<puk unblock sim status>,
  "message": "<additional info>"
}
```

value	Description
true	SIM-Karte ist bereit
false	Nutzerinteraktion benötigt (SIM-Karte einlegen, PIN oder PUK eingeben etc.)

pin validation status	Description
true	PIN der SIM-Karte wird benötigt
false	Pin der SIM-Karte wird nicht benötigt

puk unblock sim status	Description
true	PUK für Entsperrung der SIM-Karte wird benötigt
false	PUK für Entsperrung der SIM-Karte wird nicht benötigt

3.13.4 Examples

Example:

- Request:

```
GET http://<target_ip>/cgi-bin/simcard.cgi?status=
```

- Response:

```
{
  "success":true,
  "needspin":false,
  "needspuk":false,
  "message":"SIM PIN code correctly entered, or not required. "
}
```

```
{
  "success":false,
  "needspin":false,
  "needspuk":false,
  "message":"SIM card absent. "
}
```

```
{
  "success":false,
  "needspin":true,
  "needspuk":false,
  "message":"SIM card present and needs PIN, 3 tries left. "
}
```

```
{
  "success":false,
  "needspin":false,
  "needspuk":true,
  "message":"SIM card blocked, 10 PUK tries left. "
}
```

Example:

- Request:

```
GET http://<target_ip>/cgi-bin/simcard.cgi?pin=0000
```

- Response:

```
{
  "success":true,
  "needspin":false,
  "needspuk":false,
  "message":"SIM PIN code correctly entered, or not required."
}
```

```
{
  "success":true,
  "needspin":false,
  "needspuk":false,
  "message":"SIM PIN code correctly entered, or not required. SIM PIN validated
successfully, and stored for subsequent use"
}
```

```
{
  "success":false,
  "needspin":true,
  "needspuk":false,
  "message":"SIM card present and needs PIN, 2 tries left. SIM PIN validation
failed"
}
```

```
{
  "success":false,
  "needspin":false,
  "needspuk":true,
  "message":"SIM card blocked, 10 PUK tries left. SIM PIN validation failed"
}
```

Example:

- Request:

```
GET http://<target_ip>/cgi-bin/simcard.cgi?pin=12345678&puk=87654321
```

- Response:

```
{
  "success":true,
  "needspin":false,
  "needspuk":false,
  "message":"SIM PIN code correctly entered, or not required."
}
```

```
{
  "success":true,
  "needspin":false,
  "needspuk":false,
  "message":"SIM PIN code correctly entered, or not required. Successfully
unblocked the SIM, and stored PIN for subsequent use"
}
```

```
{
  "success":false,
  "needspin":true,
  "needspuk":false,
  "message":"SIM card present and needs PIN, 3 tries left."
}
```

```
{
  "success":false,
  "needs-pin":false,
  "needs-puk":true,
  "message":"SIM card blocked, 9 PUK tries left. Set PUK failed"
}
```

```
{
  "success":false,
  "needs-pin":false,
  "needs-puk":false,
  "message":"SIM PIN terminated, no PUK tries left. Set PUK failed"
}
```

Example:

- Request:

```
GET http://<target_ip>/cgi-bin/simcard.cgi?something
```

```
GET http://<target_ip>/cgi-bin/simcard.cgi?puk=12345678
```

- Response:

```
{
  "success":false,
  "message":"Missing parameters: no 'status' or 'pin'"
}
```

3.14 Kommando position

Rückgabe der Position des Geräts (GNSS).

3.14.1 Syntax

```
GET <target_ip>/cgi-bin/position.cgi
```

3.14.2 Query Parameters

None

3.14.3 Return Values

JSON-Struktur mit den folgenden Informationen:

Value	Description
state	Status of the positioning service - Disabled: GNSS-Gerät ist deaktiviert - Uninitialized: GNSS-Gerät ist nicht initialisiert - Ready: GNSS-Gerät ist bereit - Active: GNSS-Gerät ist aktiv - Unknown: Status unknown
valid	Validity of position Y. - true : Valid position Y - false: ungültige Position
latitude	Breitengrad (dezimaler Wert)
longitude	Längengrad (dezimaler Wert)
accuracy	Accuracy in metres

3.14.4 Examples

Example:

- Request:

```
GET http://<target_ip>/cgi-bin/position.cgi
```

- Response:

```
{
  "state": "Active",
  "valid": true,
  "latitude": 56.665728,
  "longitude": 12.863710,
  "accuracy": 26.000000
}
```

3.15 Kommando openvpnconfig

Die OpenVPN-Client-Konfiguration (.ovpn-Datei) hochladen, die verwendet werden soll.

- Sicherstellen, dass der Parameter [openvpnEnable] in den Router-Einstellungen den Wert [true] besitzt, damit die Konfiguration verwendet werden kann.

3.15.1 Syntax

```
POST http://<target_ip>/cgi-bin/openvpnconfig.cgi
```

3.15.2 Query Parameters

None

3.15.3 Anfragedaten

Der Inhalt der OpenVPN-Konfigurationsdatei.

3.15.4 Return Values

```
{  
  "success":<value>  
}
```

value	Description
true	Data transfer completed successfully.
false	Anfrage fehlgeschlagen oder Validierungsfehler oder Konfiguration gesperrt

3.16 Kommando openvpnlog

OpenVPN-Log abrufen.

3.16.1 Syntax

```
GET http://<target_ip>/cgi-bin/openvpnlog.cgi
```

3.16.2 Query Parameters

None

3.16.3 Return Values

Log-Datei im Klartext

3.16.4 Examples

Example:

- Request:

```
GET http://<target_ip>/cgi-bin/openvpnlog.cgi
```

- Response:

```
2023-03-03 14:57:20 DEPRECATED OPTION: --cipher set to 'AES-256-CBC' but
missing in --data-ciphers (AES-256-GCM:AES-128-GCM). Future OpenVPN version
will ignore --cipher for cipher negotiations. Add 'AES-256-CBC' to --data-
ciphers or change --cipher 'AES-256-CBC' to --data-ciphers-fallback 'AES-256-
CBC' to silence this warning.
2023-03-03 14:57:20 OpenVPN 2.5.7 arm-poky-linux-gnueabi [SSL (OpenSSL)] [LZO]
[LZ4] [EPOLL] [MH/PKTINFO] [AEAD] built on Sep 15 2022
2023-03-03 14:57:20 library versions:OpenSSL 1.0.2p 14 Aug 2018, LZO 2.10
2023-03-03 14:57:20 Outgoing Control Channel Authentication: Using 256 bit
message hash 'SHA256' for HMAC authentication
```

3.17 Kommando openvpnstat

OpenVPN-Statistiken abrufen.

3.17.1 Syntax

```
GET http://<target_ip>/cgi-bin/openvpnstat.cgi
```

3.17.2 Query Parameters

None

3.17.3 Return Values

Statistikdatei im Klartext.

3.17.4 Examples

Example:

- Request:

```
GET http://<target_ip>/cgi-bin/openvpnstat.cgi
```

- Response:

```
OpenVPN STATISTICS
Updated,2023-03-03 14:59:21
TUN/TAP read bytes,0
TUN/TAP write bytes,0
TCP/UDP read bytes,4391
TCP/UDP write bytes,3988
Auth read bytes,224
pre-compress bytes,0
post-compress bytes,0
pre-decompress bytes,0
post-decompress bytes,0
END
```

3.18 Kommando sms

Empfangen oder senden von SMS.

Erlaubte Zeichen - Senden:

- ASCII character set

Erlaubte Zeichen - Empfangen:

- UTF8-Zeichensatz
- Binär- und PDU-SMS werden in zwei Zeichen Hex-Text umgewandelt. Die Dekodierung muss vom Anrufer vorgenommen werden.
- Es können mehrere Aufrufe an die API erforderlich sein, um alle Teile einer PDU-SMS zu erhalten.

3.18.1 Syntax

receive:

```
GET http://<target_ip>/cgi-bin/sms.cgi?receive=
```

transmit:

```
GET http://<target_ip>/cgi-bin/sms.cgi?sendtext=My SMS text!
&sendnumber=+460123456789
```

3.18.2 Query Parameters

Name	Description
receive	Empfangen und SMS im Modemspeicher löschen
sendtext	String to be sent
sendnumber	Telefonnummer, an die gesendet werden soll

3.18.3 Return Values

JSON-Struktur mit den folgenden Informationen:

Value	Description
success	Status of the enquiry • false: Senden/empfangen fehlgeschlagen; keine Nachricht verfügbar • true: Nachricht verfügbar, Nachricht gesendet
type	Message type text, PDU und binäre PDU und bin werden zu two char hex text
message	Nachrichteninhalt: PDU-Typ kann mehr als einen Aufruf erfordern, um die Nachricht vollständig zu erhalten
fromnumber	Abendernummer /-Alias (wenn PDU-Typ, dann leer; Nummer ist in PDU eingebettet)
error	Fehlerinformationen über Funk, SIM, Netzwerk etc.

3.18.4 Examples

Example:

- Request:

```
GET http://<target_ip>/cgi-bin/sms.cgi?receive=
```

- Response:

```
{
  "success":false,
  "message":,
  "fromnumber":,
  "error":"SIM card absent"
}
```

```
{
  "success":false,
  "message":,
  "fromnumber":,
  "error":"No messages found"
}
```

```
{
  "success":true,
  "type":"text",
  "message":"A nice short SMS",
  "fromnumber":"+461234567890"
}
```

```
{
  "success":true,
  "type":"text",
  "message":"A text with newline:\n and other json protected chars like \" and
  \\/ and \\",
  "fromnumber":"+461234567890"
}
```

Example:

- Request:

```
GET http://<target_ip>/cgi-bin/sms.cgi?receive=
```

- Response:

```
{
  "success":true,
  "type":"PDU",
  "message":"07916407970900F1400B916407644905F5000032303180456440A00500038D0201
8220769A4E679741ECB7FB5C9683E8653C5D7145A7C768D0F8EDA687D36E907AFE7683E0F237BD3
CA697C9A0313A2C9F83D8E97519240285DD64D0CB050ABAC9A0F01C94A683D2735018C44ED3E9EC
32489CA683D86FF7B92C07A5E9A076380F129741E3379B5D1ED3CB6450780E0A83A0C42A889E879
74166F9BB0DA2A3CB2066F91CA6BF41",
  "fromnumber":""
}
```

Example:

- Request:

```
GET http://<target_ip>/cgi-bin/sms.cgi?receive=
```

- Response:

```
{
  "success":true,
  "type":"PDU",
  "message":"07916407970900F1440B916407644905F5000032303180345040340500038C0202
E42072D90D7A8FD120771D547683E869361B2402B5CB6410DDFE009D40F00768CEAED3CBF4D3C90
5",
  "fromnumber":""
}
```

Example:

- Request:

```
GET http://<target_ip>/cgi-bin/sms.cgi?sendtext=My SMS text!
&sendnumber=+460123456789
```

- Response:

```
{
  "success":false,
  "message":,
  "fromnumber":,
  "error":"Unknown network registration"
}
```

```
{
  "success":true
}
```