Avocent® ACS800/8000 Advanced Console System

User Guide

Application Programming Interface (API)

The information contained in this document is subject to change without notice and may not be suitable for all applications. While every precaution has been taken to ensure the accuracy and completeness of this document, Vertiv assumes no responsibility and disclaims all liability for damages resulting from use of this information or for any errors or omissions. Refer to other local practices or building codes as applicable for the correct methods, tools, and materials to be used in performing procedures not specifically described in this document.

The products covered by this instruction manual are manufactured and/or sold by Vertiv This document is the property of Vertiv and contains confidential and proprietary information owned by Vertiv. Any copying, use or disclosure of it without the written permission of Vertiv is strictly prohibited.

Names of companies and products are trademarks or registered trademarks of the respective companies. Any questions regarding usage of trademark names should be directed to the original manufacturer.

Technical Support Site

If you encounter any installation or operational issues with your product, check the pertinent section of this manual to see if the issue can be resolved by following outlined procedures.
Visit https://www.Vertiv.com/en-us/support/ for additional assistance.

TABLE OF CONTENTS

1 OVERVIEW

This document defines the native RESTful Application Programming Interface (API) for the Avocent® ACS800/8000 Advanced Console System from Vertiv.

1.1 Base URL

The base URL format is: https://<IP_ADDRESS>:<PORT_NUMBER>/api/v1/.

NOTE: The version number is mandatory to identify future enhancements and expansions.

The HTTPS port number is 48048 by default. HTTP may also be used with the default port number, 8080, but is completely insecure. By default, HTTP access is disabled and only HTTPS access is enabled. Both HTTP and HTTPS access can be enabled or disabled by the admin and their respective port numbers changed. Changes to these settings will restart the RESTful API server and disconnect any existing RESTful sessions.

1.2 Methods

The following basic HTTP methods are supported.

Table 1.1 Supported HTTP Methods

METHOD	DESCRIPTION
POST	Used to create a new resource (specified in the JSON body) underneath the resource specified by the URL. The created resource is returned in the body or enough information is returned to find the new (an ID or URL). Also used to initiate actions.
GET	Requests a representation of the specified resource. This has no other effects other than reading and returning the data.
PUT	Modifies an existing resource (specified in the URL) with the data present in the JSON body. Only the items present in the body are modified and the rest are left unchanged. The response code is typically 204 (Status No Content) with no content in the body unless otherwise requested by a specific parameter.
PATCH	Similar to the PUT method, but is typically intended only for modifying a portion of the specified resource. PUT in this API is also allowed to only modify a portion of a resource, so PATCH is included just for those applications that already use it.
DELETE	Deletes the resource specified by the URL.

1.3 Body

The body for GET/POST/PUT/PATCH requests uses JSON syntax. This means the "Content-Type" and "Accept" headers should be set to application/json. All parameters and values are case sensitive. Parameters that are string types must have their value enclosed in quotes even if the content is numerical, as is the case with some of the parameters that have a drop-down menu in the Web UI. All parameters are string type unless otherwise specified to be an integer, array or something else.

1.4 Query Parameters

1.4.1 Fields

The fields query parameter is supported for many resources to enable you to limit the fields that are returned. The following example returns an array of all the serial ports with only the speed and pinout fields of each:
GET /serialPorts?fields=pinout,speed

Sub-fields such as speed, which is under the physical portion of a serial port, are unique and can be specified without any reference to the parent (physical in this example).

1.4.2 Filtering

Filtering can be performed on certain resources that return an array of items by specifiying attributes of that resource that should be matched for each item.

GET /serialPorts?status=enabled&speed=9600

This request returns an array of serial ports that are enabled and set to a speed of 9600.

NOTE: The ampersand (&) is not meant to imply a logical "and" operation, but is instead a standard URL delimiter. A logical "and" is implicit because all specified filters must match for the element to be included in the returned resource list.

For any single parameter, multiple acceptable values can be specified and separated by commas.

GET /serialPorts?speed=1200, 2400, 4800

This request returns an array of serial ports that have port speeds of 1200, 2400 or 4800.

Ranges for integer fields can be specified using a hyphen (-) and can be combined with the comma separated list.

GET /serialPorts?port=1,3-5,7

This request returns an array of serial ports consisting of ports 1, 3, 4, 5 and 7.

A value prefixed with "~" inverts the condition.

GET /serialPorts?profile=~cas

This request returns an array of serial ports with the profile NOT set to cas.

1.4.3 Ranges

Ranges are supported for certain numeric parameters when filtering and will use the following format:

GET /serialPorts?port=1,3-5,7

This returns only the items specified. PUT and PATCH should also behave in this manner.

1.5 Response Codes

The API utilizes standard HTTP response codes where appropriate. The following table lists the supported response codes and typical usage.

Table 1.2 Response Code Descriptions

RESPONSE CODE	MEANING	DESCRIPTION
200 OK	Success	Returned for a successful request;may include a JSON body with results.
201 Created	Created	Returned for a successful request that has resulted in the creation of a new resource.
204 No Content	Success	Returned for asuccessful request; does NOT include a JSON body with results.

Table 1.2 Response Code Descriptions (continued)

RESPONSE CODE	MEANING	DESCRIPTION
400 Bad Request	Failure	Returned on afailed request; includes a JSON encoded list of errors for one or more of the problematic parameters. May also indicate other system errors.
401 Unauthorized	Authorization Failure	Returned for arequest without the proper authentication.
404 Not Found	API not active	Returned for arequest where the resource is not found; typically includes a JSON-encoded error structure with more error detail.

1.5.1 Error information

In addition to returning a failing response code, error information is returned in the response body, which provides more detail. This response is in JSON format as follows:.

```
{
    "error": {
        "code": "AE003",
        "message": "invalid parameter",
        "detail": "bob"
    }
}
```

NOTE: Not all error information responses include the detail field.

1.5.2 Ignored keys

If unknown keys are sent as part of the JSON body of a PUT or PATCH request, they are ignored by the API. This is part of the RESTful way of supporting different devices and different versions of API implementation. If a device doesn't understand or support something, it is permitted to ignore it.

Rather than ignoring mis-spellings and leaving the caller to wonder why a parameter didn't get Set, the API returns a response body along with the 200 response code that contains a successful response message, along with a list of keys that were ignored. This permits the caller to look for this information, if desired..

```
{
    "status": "success",
    "ignoredKeys": [
        "datea",
        "timee"
    ]
}
```

If a parent key is ignored, all children below it are ignored but are not processed and listed as ignored keys. Even if all keys are ignored, a successful response is still returned because the command didn't fail to write anything that it attempted to write.

1.6 Authentication

The ACS RESTful API supports two different methods of authentication: JSON Web Token (JWT) and Basic Authentication.

1.6.1 JSON Web Token (JWT)

Using the JWT method, you can log in and authenticate using the /sessions/login resource, passing it aa valid appliance username and password in the JSON body of the request. If successful, the appliance returns a JWT that must be included in the header of all subsequent requests as the "Authorization" key with a value of "Bearer <JWT>". This minimizes some of the appliance authentication overhead on each individual RESTful API call.

The JWT remains valid for 60 minutes. A GET request (or call) via the /sessions/refresh resource can be sent before the token expires to refresh and provide a new token.

NOTE: The username and password are transmitted as unencrypted plain text in the original /sessions/login request body, so it is recommended to use HTTPS for RESTful communications.

1.6.2 Basic authentication

Basic authentication takes a username/password pair and encodes it using base64. The resulting base64 value must then be included in every request header as the "Authorization" key with a value of "Basic <BASE64_VALUE>". After the request is authenticated by the appliance, the request is executed and the session terminated.

NOTE: The username and password are transmitted unencrypted (base64 is NOT a secure encryption) in every request, so it is recommended to use HTTPS for RESTful communications.

1.7 Document Conventions

1.7.1 Abbreviated URLs

To make this document more readable, the URLs in the examples are generally abbreviated to show only the portion of the URL after the /v1. The full URL is necessary when using the API. For example:

/system/info is shown instead of https://10.20.30.40:48048/api/v1/system/info

1.7.2 Examples

In the RESTful examples throughout this document, the request is shown in **bold** type, including both URL components and the message body. The response body is shown in normal type.

Example:

```
POST /sessions/login {"username": "admin", "password": "avocent"}
        {
            "token":"eyJhbGciOiJIUzI1NiIsInR...fVzCM"
        }
```

2 API RESOURCES AND METHODS

The following table lists the resources the API provides.

Table 2.1 URL Descriptions

URL	METHOD	DESCRIPTION
Sessions		
/sessions/login	POST	Logs in with the username and password, creates an API session, and returns a token in JSON body.
/sessions/logout	POST	Exits and cleans up the current session.
/sessions/refresh	GET	Using the token provided, obtains a new refreshed token which is returned in the JSON body response.
/sessions	GTE	List of all active sessions.
/sessions/<ID>	GET DELETE	Returns details of specified session or kills the specified session.
Access		
/access/serialPorts /access/serialPorts/<PORT#>/on /access/serialPorts/<PORT#>/off /access/serialPorts/<PORT#>/cycle	GET POST	Returns a list of ports and devices that can be accessed and provides actions to control power if configured.
System		
/system/info	GET	Reads basic system info such as serial number and type.
/system/reboot	POST	Reboots the appliance. Returns immediately.
/system/shutdown	POST	Shuts down the appliance. Returns immediately.
/system/factoryDefault	POST	Resets the appliance to the factory defaults and reboots.
/system/firmware/version	GET	Returns version information of firmware, bootcode and date of build.
/system/firmware/download	POST	Downloads a firmware image file to the appliance using ftp, sftp or scp.
/system/firmware/install	POST	Installs a previously downloaded firmware image file.
/system/firmware/downloaded	GET	Returns the version information of a previously downloaded firmware file.
/system/config/save	POST	Saves the appliance configuration to a file.
/system/config/restore	POST	Restores the appliance configuration from a saved file.
/system/dateAndTime	GET PUT PATCH	Reads and configures system time parameters.
/system/dateAndTime/timezones	GET	Returns a list of all recognized time zones.
/system/general	GET PUT PATCH	Reads and configures general system parameters, including online Help, language, banner and so on.
/system/bootConfig	GET PUT PATCH	Reads and configures the boot configuration parameters.
/system/certificate/download /system/certificate/generate /system/certificate/apply	POST	Manages certificates.
/system/integrity/generate /system/integrity/verify	POST	Generates and verifies MD5 system configuration integrity.
/system/usage/memory	GET	Provides read-only system usage information.

Table 2.1 URL Descriptions (continued)

URL	METHOD	DESCRIPTION
/system/usage/flash		
Security		
/security	GET PUT PATCH	Reads and configures various security profile parameters.
Network		
/network/settings	GET PUT PATCH	Reads and configures various appliance specific network parameters.
/network/devices /network/devices/<ETH#>	GET PUT PATCH	Reads and configures network device-specific parameters: method (dhcp or static), IP address, netmask, gateway and so on.
/network/staticRoutes/[ipv4/ipv6]	GET PUT PATCH POST DELETE	Reads and configures network static routes.
/network/hosts /network/hosts/<ADDR>	GET PUT PATCH POST DELETE	Configures network host names on local network.
/network/firewall/[ipv4|ipv6]	GET PUT PATCH POST DELETE	Creates and configures chains for the ipv4 and ipv6 filter tables.
/network/ipsec/certificates	GET	Lists IPSec PKCS12 files.
/network/ipsec/certificates/<NAME>	DELETE	Deletes IPSec PKCS12 files.
/network/ipsec/certificates/download	POST	Downloads an IPSec certificate.
/network/ipsec/connections[/<NAME>]	GET PUT PATCH POST DELETE	Reads and configures IPSec connections and settings.
/network/snmp[/<ID>]	GET PUT PATCH POST DELETE	Reads and configures SNMP settings.
/network/snmp/system	GET PUT PATCH	Reads and configures SNMP system settings.
Ports		
/serialPorts /serialPorts/<PORT#>	GET PUT PATCH	Reads and configures various serial port parameters: status, pinout, parity, profile and so on.
/serialPorts/<PORT#>/alerts /serialPorts/<PORT#>/alerts/<ID> /serialPorts/<PORT#>/alerts/clear /serialPorts/<PORT#>/alerts/deleteAny	GET POST DELETE	Reads and configures alert strings.
/serialPorts/<PORT#>enable	POST	Enables the serial port.
/serialPorts/<PORT#>disable	POST	Disables the serial port.
/serialPorts/<PORT#>resetToFactory	POST	Resets the serial port to factory defaults.
/serialPorts/<PORT#>/power /serialPorts/<PORT#>/power/<ID>	GET POST DELETE	Merges PDU outlets or UPS outlet groups with the serial port. Views all merged outlets.
/auxPorts/ /auxPorts/<NAME>	GET PUT PATCH	Reads and configures Auxiliary Port parameters.
/casProfile	GET PUT	Reads and configures the CAS profile parameters.

Table 2.1 URL Descriptions (continued)

URL	METHOD	DESCRIPTION
	PATCH	
/casProfile/probeStrings[/<ID>]	GET POST DELETE	Reads and configures probe strings used by auto discovery.
/casProfile/matchStrings[/<ID>]	GET POST DELETE	Reads and configures match strings used by auto discovery.
/casProfile/autoAnswer[/<ID>]	GET POST DELETE	Reads and configures auto answer string pairs.
/dialinProfile	GET PUT PATCH	Reads and configures the Dial-In Profile settings.
/dialinProfile/callbackUsers	GET POST DELETE PUT PATCH	Reads and configures the list of callback users and their numbers from the Dial-In Profile.
/dialinProfile/pppOtpUsers	GET POST DELETE	Reads and configures the list of PPP OTP users from the Dial-In Profile.
Pluggable		
/pluggableDevices /pluggableDevices/<NAME>	GET	Returns a list of attached pluggable devices and their details or returns details on the named pluggable device.
/pluggableDevices/<NAME>/eject	POST	Ejects the specified device so that it is safe to remove.
/pluggableDevices/<NAME>/delete	POST	Deletes the specified device after unplugging it.
/pluggableDevices/<NAME>/setConsole	POST	Sets up the specified pluggable device as a console port.
Authentication		
/authentication/	GET PUT PATCH	Reads and configures general appliance authentication parameters.
/authentication/dsview /authentication/kerberos /authentication/ldap /authentication/radius /authentication/tacacs	GET PUT PATCH	Reads and configures authentication parameters specific to each type of authentication server.
Users		
/users /users/<NAME>	GET PUT PATCH POST DELETE	Reads and configures user-specific parameters, including listing all users, adding a user and editing existing users.
/users/<NAME>/unlock	POST	Unlocks the user account if it is locked.
/users/passwordRules	GET PUT PATCH	Reads and configures the user password rules.
Events and Logs		
/events[/<ID>]	GET PUT PATCH	Enables or disables the various events and configures event notifications.
/events/applianceLogging	GET PUT PATCH	Reads and configures appliance logging settings.
/events/dataBuffering	GET PUT PATCH	Reads and configures data buffer settings.

Table 2.1 URL Descriptions (continued)

URL	METHOD	DESCRIPTION
/events/dsview	GET PUT PATCH	Reads and configures the DSView event destination parameters.
/events/snmp	GET PUT PATCH	Reads and configures the event destination SNMP parameters.
/events/sms	GET PUT PATCH	Reads and configures the event destination SMS parameters.
/events/email	GET PUT PATCH	Reads and configures the event destination email parameters.
/events/syslog	GET PUT PATCH	Reads and configures the event destination syslog parameters.
/events/trapForward[/<ID>]	GET PUT PATCH POST DELETE	Reads and configures the Trap Forward entries.
Power Management		
/power/login	GET PUT PATCH	Reads and configures the login credentials for various brands of PDUs.
/power/networkPdus[/<ADDR>]	GET PUT PATCH POST DELETE	Reads and configures the list of Network PDUs.
/power/networkUps[/<ADDR>]	GET PUT PATCH POST DELETE	Reads and configures the list of Network UPS devices.
/power/pdus[/<NAME>]	GET PUT PATCH	Read and configure PDU settings.
/power/pdus/<NAME>/banks[/<ID>]	GET PUT PATCH	Read and configure bank settings for a PDU.
/power/pdus/<NAME>/banks/<ID>/resetValues	POST	Resets the electrical monitoring values for a PDU.
/power/pdus/<NAME>/cycle /power/pdus/<NAME>/factoryDefaults /power/pdus/<NAME>/off /power/pdus/<NAME>/on /power/pdus/<NAME>/reboot /power/pdus/<NAME>/rename /power/pdus/<NAME>/resetValues	POST	Provides control of the PDU.
/power/pdus/<NAME>/outlets[/<ID>]	GET PUT PATCH	Views the list of outlets on the PDU. Configures outlet settings.
/power/pdus/<NAME>/outlets/<ID>/cycle /power/pdus/<NAME>/outlets/<ID>/lock /power/pdus/<NAME>/outlets/<ID>/off /power/pdus/<NAME>/outlets/<ID>/on /power/pdus/<NAME>/outlets/<ID>/resetValues /power/pdus/<NAME>/outlets/<ID>/unlock	POST	Provides control of the outlet.
/power/pdus/<NAME>/phases[/<ID>]	GET PUT PATCH	Views the list of phases on the PDU. Configures phase settings.
/power/pdus/<NAME>/phases/<ID>/resetValues	POST	Resets the electrical monitoring values for a phase.
/power/pdus/<NAME>/sensors[/<ID>]	GET PUT PATCH	Views the list of sensors on the PDU. Configures sensor settings.
/power/pdus/<NAME>/sensors/<ID>/resetValues	POST	Resets the monitoring value for a sensor.
/power/ups[/<NAME>]	GET PUT PATCH	Read and configure UPS settings.

Table 2.1 URL Descriptions (continued)

URL	METHOD	DESCRIPTION
/power/ups/<NAME>/outletGroups[/<ID>]	GET	Views the list of controllable outlet groups on the UPS.
/power/ups/<NAME>/outletGroups/<ID>/cycle /power/ups/<NAME>/outletGroups/<ID>/off /power/ups/<NAME>/outletGroups/<ID>/on	POST	Provides control of the outlet group.
/power/ups/<NAME>/outputCycle /power/ups/<NAME>/outputOff /power/ups/<NAME>/outputOn /power/ups/<NAME>/rename /power/ups/<NAME>/resetPowerStats /power/ups/<NAME>/restoreName /power/ups/<NAME>/silenceAlarm /power/ups/<NAME>/testBattery	POST	Provides control of the UPS.
Sensors		
/sensors/internal	GET PUT PATCH	Reads the internal temperature sensors, and configures threshold values.
/sensors/1wire[/<ADDR>]	GET DELETE PATCH PUT	Provides access to the attached 1-wire sensors.
/sensors/1wire/<ADDR>/refresh	POST	Refreshes the list of 1-wire sensors.
/sensors/digitalIn		
/sensors/pdu /sensors/pdu/<NAME>/reset	GET POST	Resets the monitoring values for a sensor.
Monitoring		
/monitoring/network/devices[/<INT>]	GET	Lists the network devices and their IP addresses.
/monitoring/serialPorts[/<ID>]	GET DELETE	Provides information about the current state of the serial ports. The DELETE method clears the transmission counter to zero for an individual serial port.
/monitoring/ipsec[/<NAME>]	GET	Provides information about the IPSec tunnel status.
/monitoring/network/routingTables/[ipv4|ipv6]	GET	Lists the IPv4/IPv6 routing tables.
Miscellaneous		
/resources	GET	Return a list of available API resources.

2.1 Sessions

2.1.1 /sessions/login

This command establishes a connection using the username and password provided in the JSON body with the configured authentication of the appliance.

A web token is returned to be sent in the header of ALL subsequent requests as the "Authorization" key with a value of "Bearer <TOKEN>".

An alternative to this log-in session is to send a base64 encoded username/password pair in every API request, using an "Authorization" key with a value of "Basic <BASE64_VALUE>". For more information, see Authentication on page 3.

Methods

POST

Parameters

PARAMETER	DESCRIPTION
username	Valid username of an account on the appliance (root, admin and so on)
password	Valid password for the specified username

Query

None

Response Body

JSON object

Response Codes

200	OK
401	Not Authorized
40x	Failure

Examples

```
POST /sessions/login {"username": "admin", "password": "avocent"}
        {
            "token":"eyJhbGciOiJIUzI1NiIsInR...fVzCM"
        }
POST /sessions/login {"username": "bad", "password": "bad"}
        {
            "error": {
                "code": "AE017",
                "message": "user authentication failed"
            }
        }
```

2.1.2 /sessions/logout

This action disconnects the current connection and lets the appliance free any web token or other session authentication and clean up the session.

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
40x	Failure

Examples

```
POST /sessions/logout
        {
            "logout": "OK",
            "username": "root"
        }
```

2.1.3 /sessions/refresh

This command refreshes the current connection token to keep it active.

Methods

GET

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
40x	Failure

Examples

```
GET /sessions/refresh
        {
            "token":"eyJhbGciOiJIUzI1NiIsInR...fVzCM"
        }
```

2.1.4 /sessions/<ID>

This resource provides the ability to view all active sessions and their session information and to kill specific sessions.

Methods

GET, DELETE

Parameters

Table 2.2 <ID#> Parameters

PARAMETER	DESCRIPTION
id	Unique session ID number. Integer.
user	Username which owns the session.
clientIp	IP Address of the client which originated the session.
creationTime	Date and time on which the session was created.
sessionType	Type of session: ssh/telnet/console/http/https/raw/unknown.
connectionType	Type of connection: serial/wmi/cli/api/unknown.
targetName	Name of the target of the session, or blank if the appliance.
parentId	Unique session ID number.

Query

Fields are supported for all parameters. Filtering is supported for all parameters except for creationTime on GET/sessions.

Response Body

JSON object

Response Codes

200	OK
404	Not Found
40x	Failure

Examples

GET /sessions

```
{
    "sessions": [
        {
            "id": 24,
            "user": "root",
            "clientIp": "10.20.30.40",
            "creationTime": "Tue 01 Aug 2017 07:15:36 PM UTC",
            "sessionType": "http",
            "connectionType": "wmi",
            "target": "",
            "parentId": "",
        },
        {
            "id": 43,
            ...
        },
        ...
    ]
}
```

GET /sessions/100

```
{
    "id": 100,
    "user": "root",
    "clientIp": "10.20.30.40",
    "creationTime": "Tue 01 Aug 2017 07:15:36 PM UTC",
    "sessionType": "http",
    "connectionType": "wmi",
    "target": "",
    "parentId": "",
}
```

DELETE /sessions/101

2.2 System

2.2.1 /system/info

This resource provides access to read only system information about the appliance's identity, versions, power and CPU information.

Methods

GET

Parameters

PARAMETER	DESCRIPTION
serialNumber	Serial number assigned to the appliance at the factory.
type	Description of the type of unit, including the model number with port count, power supplies and modem presence. Example: ACS8048 with single power supply.
bootcode	Version number of the installed bootcode. Example: 1.17
firmware	Full version number of the installed firmware. Example: 1.3.75.2779+551+28+11
firmwareDate	Date of the installed firmware. Example: Sep 1 2016 - 04:07:14
bootedFrom	Identifies whether the appliance is currently booted from hardware (internal Flash) or network.
powerSupply1	Status of power supply 1: on/off
powerSupply2	Status of power supply 2, if present: on/off
cpu	Description of the cpu: ARMv7 Processor rev 0 (v71)
cores	Number of cores in the cpu: integer 2

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
400	Bad request

Examples

GET /system/info

```
{
    "serialNumber": "1234567890",
    "type": "ACS8048 with single power supply",
    "bootcode": "1.17",
    "firmware": "1.3.75.2779+551+28+11",
    "firmwareDate": "Sep 1 2016 - 04:07:14",
    "bootedFrom": "hardware",
    "powerSupply1": "on",
    "cpu": "ARMv7 Processor rev 0 (v7l)",
    "cores": 2
}
```

2.2.2 /system/reboot

This command causes the appliance to reboot.

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
40x	Failure

Examples

```
POST /system/reboot
        {
            "status": "initiated reboot"
        }
```

2.2.3 /system/shutdown

This command causes the appliance to shut down.

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
40x	Failure

Examples

```
POST /system/shutdown
        {
            "status": "initiated shutdown"
        }
```

2.2.4 /system/factoryDefault

This action restores the appliance to the factory default and reboots the appliance.

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
40x	Failure

Examples

```
POST /system/factoryDefault
        {
            "status": "initiated factoryDefault"
        }
```

2.2.5 /system/firmware/version

This resource provides information about the currently installed and running firmware, including build date and version numbers of various components.

Methods

GET

Parameters

PARAMETER	DESCRIPTION
version	Full version number of the installed firmware. Example: 1.3.75.2779+551+28+11
bootVersion	Version number of the installed bootcode. Example: 1.18
date	Date of the firmware build. Example: Aug 12 2017 - 09:12:24

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
GET /system/firmware/version
        {
            "version": "1.3.75.2779+551+28+11",
            "bootVersion": "1.17",
            "date": "Aug 12 2017 - 09:12:24"
        }
```

2.2.6 /system/firmware/download

This action causes the appliance to download the firmware file specified in preparation for subsequent firmware updating. The command does not return until the file download is complete or fails. Depending on the network speed, this could take a couple minutes.

Methods

POST

Parameters

PARAMETER	DESCRIPTION
protocol	Specifies the protocol (**ftp/scp/sftp**) to use to download the firmware.
ipAddress	IP address of the remote server from which to download the file.
username	Username to access the remote server.
password	Password to access the remote server.
directory	Directory path on the remote server, typically relative to the ftp root directory.
filename	Filename of the firmware file on the remote server.

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
POST /system/firmware/download
        {
                "protocol": "ftp",
                "ipAddress": "10.20.30.80",
                "username": "anonymous",
                "password": "anonymous",
                "directory": "pub/firmware/",
                "filename": "firmware_acs8_1_2_9.fl"
        }
Response is:
        {
            "status": "download successful",
            "firmware": {
                "version": "1.2.9.2449+540+23+11",
                "date": "03/01/17"
            }
        }
```

2.2.7 /system/firmware/install

This action causes the appliance to install a previously downloaded firmware image into flash memory. This request does not return until the installation is complete, which may take up to two minutes.

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

POST /system/firmware/install

```
{
    "status": "install successful",
    "firmware": {
    "version": "1.2.9.2449+540+23+11",
    "date": "03/01/17"
    }
}
```

2.2.8 /system/firmware/downloaded

This resource provides information about a firmware image that has previously been downloaded to the appliance.

Methods

GET

Parameters

PARAMETER	DESCRIPTION
version	Full version number of the firmware file. Example: 1.2.9.2449+540+23+11
date	Date of the build of the firmware file. Example: 03/01/17

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
GET /system/firmware/downloaded
        {
            "version": "1.2.9.2449+540+23+11",
            "date": "03/01/17"
        }
```

2.2.9 /system/config/save

This action saves the system configuration of the appliance and does not return until the save is complete. It may take several minutes depending upon the format.

NOTE: XML format is not supported in the API.

Methods

POST

Parameters

PARAMETER	DESCRIPTION
format	Format used to save the configuration. Default is cli if nothing is specified. Otherwise choose from: **cli/compressed**.
where	Where to save the file, either to the local appliance file system or to a remote server: **local/remote**.
protocol	Protocol used to transfer the file to a remote server: **ftp/scp/sftp**.
ipAddress	IP Address of the remote server.
username	Username of the account used to log in to the remote server.
password	Password for the specified username to log into the remote server. Defaults to "anonymous" if none is provided.
directory	Directory where the file is to be written. If the directory starts with "/", it is considered an absolute path. Otherwise, the directory is relative to /mnt/hdUser/backup for local, which is the default location of local configuration files, or relative to the specified protocol's configured base directory on the remote server.
filename	Filename to use for the saved configuration file.

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
POST /system/config/save
        {
                "format": "cli",
                "where": "remote",
                "protocol": "ftp",
                "ipAddress": "10.20.30.70",
                "username": "anonymous",
                "password": "anonymous",
                "directory": "pub",
                "filename": "myconfig.cli"
        }
Response is:
        {
                "status": "backup configuration saved as cli config filename: pub/myconfig.cli"
        }
```

2.2.10 /system/config/restore

Restores the appliance configuration from a specified file and does not return until the restore is complete. This may take several minutes depending upon the format.

NOTE: XML format is not supported in the API.

Methods

POST

Parameters

PARAMETER	DESCRIPTION
where	Where the configuration file is located: **local/remote**.
protocol	Protocol used to retrieve the remote file: **ftp/scp/sftp**.
ipAddress	IP address of the remote server.
username	Username of the account to use on the remote server.
password	Password for the specified username on the remote server. Defaults to "anonymous" if none is provided.
directory	Directory where the file is located. If the directory starts with "/", it is considered an absolute path. Otherwise, the directory is relative to /mnt/hdUser/backup for local, which is the default location of the local configuration files, or relative to the ftp/scp/sftp configured base directory on the remote server.
filename	Name of the configuration file to restore.

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
POST /system/config/restore
        {
            "where": "remote",
            "protocol": "ftp",
            "ipAddress": "10.20.30.70",
            "username": "anonymous",
            "password": "anonymous",
            "directory": "pub",
            "filename": "myconfig.cli"
        }
Response is:
    {
        "status": "backup configuration restored from cli config filename:
    pub/myconfig.cli"
    }
```

The following command restores from the local file /mnt/hdUser/backup/myconfig.cli, which is sitting in the default local directory for backup configuration files. The directory parameter is not actually needed in this case, as it is assumed to be blank if not provided.

```
POST /system/config/restore
{
    "where": "local",
    "directory": "",
    "filename": "myconfig.cli"
}
Response is:
{
   "status": "backup configuration restored from cli config filename: myconfig.cli"
}
```

2.2.11 /system/integrity/generate

This action generates an MD5 Tag for the running configuration and takes no parameters. It returns an MD5 Tag string of the form "bf86785ade6fa3d5999f2f85752a4fd".

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

POST /system/integrity/generate
Response is:

```
{
    "md5Tag": "bf86785ade6fa3d5999f2f85752a4fd"
}
```

2.2.12 /system/integrity/verify

This action verifies the running configurations MD5 against a supplied MD5 value. If no MD5 is supplied, the current running configuration is compared against the last generated MD5.

Methods

POST

Parameters

PARAMETER	DESCRIPTION
md5Tag	MD5 tag to compare against the MD5 of the running system. Example: "bf86785ade6fa3d5999f2f85752a4fd"

Query

None

Response Body

JSON object

Response contains the Configuration Integrity Status, either "Unchanged" or "Modified".

Response Codes

200	OK
400	Bad Request

Examples

POST /system/integrity/verify '{"md5Tag":"badbadbadbadbadbadbadbadbadd"}'
Response is:

```
{
    "integrityStatus": "Modified"
}
```

2.2.13 /system/certificate/generate

This action generates a self-signed certificate for the appliance from the information provided.

Methods

POST

Parameters

PARAMETER	DESCRIPTION
country	Country Name (two letter code). Example: "us."
state	State or Province name (full name). Example: "washington."
city	City or locality name. Example: "seattle."
organization	Organization or Company name.
unit	Organizational unit or section name.
commonName	Server FQDN or another name.
emailAddress	Email address.
netscapeComment	Comment to display in Netscape's comment listbox.
passphrase	Passphrase to protect private key file.

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
POST /system/certificate/generate
        {
            "country": "us",
            "state": "washington",
            "city": "seattle",
            "organization": "",
            "unit": "",
            "commonName": "mycertificate",
            "emailAddress": "me@myemail.com",
            "netscapeComment": "",
            "passphrase": ""
        }
```

```
Response is:
    {
        "status": "certificate generation successful",
    }
```

2.2.14 /system/certificate/download

This action causes the appliance to download a certificate from the specified server information. The action does not return until the file download is complete or fails. Depending on the network speed, this could take a couple minutes.

Methods

POST

Parameters

PARAMETER	DESCRIPTION
protocol	Specifies the protocol to use to download file: **ftp/scp/sftp**
ipAddress	The IP address of the remote server from which to download the file.
username	Username to access the remote server.
password	Password to access the remote server.
directory	Directory path on the remote server, typically relative to the ftp root directory.
filename	Filename of the certificate file on the remote server.

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
POST /system/certificate/download
        {
            "protocol": "ftp",
            "ipAddress": "10.20.30.80",
            "username": "anonymous",
            "password": "anonymous",
            "directory": "pub/certificate/",
            "filename": "mycertificate"
        }
```

```
Response is:
    {
        "status": "certificate download successful",
    }
```

2.2.15 /system/certificate/apply

This action causes the appliance to apply a previously downloaded or generated certificate.

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
POST /system/certificate/apply
        {
            "status": "certificate application successful",
        }
```

2.2.16 /system/dateAndTime

Gets/Sets parameters related to the system's date and time.

NOTE: When NTP is enabled, setting the time and date is not permitted and results in an error.

Methods

GET, PUT, PATCH

Parameters

PARAMETER	DESCRIPTION
time	Current time.
date	Current date.
ntp	Network Time Protocol. **enabled/disabled**.
ntpSettings (only valid when ntp is enabled)	
ntpServer	IP address or name.
ntpServer2	IP address or name.
timezone	Name of the timezone. UTC is the default.

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

```
GET /system/dateAndTime
        {
            "time": "18:00:50",
            "date": "07/12/2017",
            "ntp": "enabled",
            "ntpSettings": {
                "ntpServer": "10.20.30.40",
                "ntpServer2": "ntp.pool.org"
            },
            "timezone": "UTC"
        }
PUT /system/dateAndTime
        {
                "ntp": "disabled"
        }
PUT /system/dateAndTime
        {
            "time": "05:00:00",
            "date": "08/01/2017"
        }
PUT /system/dateAndTime
        {
            "ntp": "enabled",
            "ntpSettings": {
                "ntpServer": "ntp.pool.org",
                "ntpServer2": ""
            }
        }
PUT /system/dateAndTime {"timezone":"US/Central"}
PUT /system/dateAndTime {"timezone":"Custom"}
```

2.2.17 /system/dateAndTime/timezones

Gets a list of all the recognized timezones. These values can be used when setting the timezone via the /system/dateAndTime resource.

Methods

GET

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
GET /system/dateAndTime/timezones
        {
            "timezones": [
                "Africa/Abidjan",
                "Africa/Accra",
                ....
                "Zulu"
            ]
         }
```

2.2.18 /system/dateAndTime/timezone/custom

Configure settings for a custom timezone.

Methods

GET, PUT, PATCH

Parameters

PARAMETER	DESCRIPTION
name	Time zone name
acronym	Standard time acronym
gmtOffset	Example: -7:00
dst	Daylight savings time. **enabled/disabled**
(Only used when dst is enabled):	
dstAcronym	Daylight Saving Time acronym.
saveTime	Example: 0:00 to 5:00
startMonth	January/…/December
startWeek	Week of the month. Integer 1-5
startWeekday	Sunday/…/Saturday
startHour	Example: 01:00
endMonth	January/…/December
endWeek	Week of the month. Integer 1-5
endWeekday	Sunday/…/Saturday
endHour	Example: 01:00

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

GET /system/dateAndTime/timezone/custom


```
{
    "name": "mycustom",
    "acronym": "myc",
    "gmtOffset": "-4:00",
    "dst": "enabled",
    "dstAcronym": "mydst",
    "saveTime": "1:00",
    "startMonth": "april",
    "startWeek": 1,
    "startWeekday": "sunday",
    "startHour": "02:00",
```

```
        "endMonth": "november",
        "endWeek": 1,
        "endWeekday": "sunday",
        "endHour": "02:00",
    }
```

```
PUT /system/dateAndTime/timezone/custom
        {
            "name": "mycustom",
            "acronym": "myc",
            "gmtOffset": "-4:00",
            "dst": "disabled"
        }
```

2.2.19 /system/general

Gets and sets general appliance level parameters.

Methods

GET, PUT, PATCH

Parameters

PARAMETER	DESCRIPTION
language	Language used for SSH, Telnet and console port sessions to the appliance. **english/chinese/french/german/japanese/spanish**.
onlineHelp	URL for the online help product documentation. Default: http://global.avocent.com/us/olh/acs8x/en/index.html.
banner	Controls whether a login banner is displayed. **enabled/disabled**.
bannerText	Text of the login banner to display. It may include special characters for formatting including \t for tabs and \n for newlines. Only shown in the full return body when banner is enabled.
viewer	Type of viewer to use when opening serial or appliance sessions. **html5/jnlp**.

Query

Fields are supported for all parameters.

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

```
GET /system/general
    {
```

```
        "language": "english",
        "onlineHelp": "http://global.avocent.com/us/olh/acs8x/en/index.html",
        "banner": "enabled",
        "bannerText":
            "=========================================================================\n
            WARNING! The use of this system is restricted to authorized users. \n  \n  All information
            and communications on this system are subject \n  to review, monitoring and recording at any
            time, without notice\n  or permission. Users should have no expectation of privacy.
            \n=========================================================================\n",
        "viewer": "html5"
    }
```

PUT /system/general {"bannerText":"==========\n Multiple\n Line\n Banner\n==========\n"}

PUT /system/general {"language":"german","banner":"disabled","viewer":"jnlp"}

2.2.20 /system/bootConfig

Gets and sets parameters related to the system's boot configuration.

Methods

GET, PUT, PATCH

Parameters

PARAMETER	DESCRIPTION
bootMode	flash, network
bootImage	Integer image number when booted from flash. **1 or 2**
flashImage1	Read only firmware version and date of the first image in flash.
flashImage2	Read only firmware version and date of the second image in flash.
applianceIp	Appliance IP address when booted from network. Example: 10.20.30.40
tftpServerIp	TFTP Server IP address when booted from network. Example: 10.20.30.41
filename	Filename when booted from network.
watchdog	**enabled/disabled**
consoleSpeed	Console port speed: **1200/2400/4800/9600/19200/38400/57600/115200**

Query

None

Response Body

None

Response Codes

200	OK
400	Bad Request

Examples

```
GET /system/bootConfig
        {
            "bootFrom": "flash",
            "bootImage": 1,
            "flashImage1": "1.3.12.2683+551+26+1 (Jun 15 2017 - 16:26:07)",
            "flashImage2": "1.3.75.2779+551+28+11 (Jul 26 2017 - 10:02:10)",
            "watchdog": "enabled",
            "consoleSpeed": 9600
        }
GET /system/bootConfig
        {
            "bootFrom": "network",
            "applianceIp": "10.20.30.40",
            "tftpServerIp": "10.20.30.41",
            "filename": "acs.fl",
            "watchdog": "enabled",
            "consoleSpeed": 9600
        }
```

2.2.21 /system/usage/memory

Gets read only system information about memory usage.

Methods

GET

Parameters

PARAMETER	DESCRIPTION
active	Memory that has been used more recently and usually not reclaimed unless absolutely necessary. Example: "66856 kB"
anonPages	Non-file backed pages mapped into userspace page tables
bounce	Memory used for block device "bounce buffers"
buffers	Relatively temporary storage for raw disk blocks
cached	In-memory cache for files read from the disk
commitLimit	Total amount of memory currently available to be allocated on the system
committedAS	Amount of memory presently allocated on the system.
dirty	Memory which is waiting to get written back to the disk
highFree	Highmem is all memory above ~860MB of physical memory; Highmem areas are for use by userspace programs, or for the pagecache
highTotal	Highmem total
inactive	Memory which has been less recently used

PARAMETER	DESCRIPTION
kernelStack	Memory the kernel stack uses
lowFree	Memory used for everything that highmem can be used for; also available for the kernel's use for its own data structures
lowTotal	Lowmem total
mapped	Files which have been mmapped, such as libraries
memAvailable	Estimate of how much memory is available for new applications, without swapping
memFree	Sum of LowFree+HighFree
memTotal	Total usable ram
mlocked	Pages locked to memory using the mlock( ) system call
nfsUnstable	NFS pages sent to the server, but not yet committed to stable storage
pageTables	Amount of memory dedicated to the lowest level of page tables
shmem	Total memory used by shared memory (shmem) and tmpfs
slab	In-kernel data structures cache
sReclaimable	Part of Slab, that might be reclaimed, such as caches
sUnreclaim	Part of Slab, that cannot be reclaimed on memory pressure
swapCached	Memory present within main memory, but also in the swapfile
swapFree	Memory evicted from RAM, and temporarily on the disk
swapTotal	Total amount of swap space available
unevictable	Unevictable pages can't be swapped out for a variety of reasons
vmallocChunk	Largest contiguous block of vmalloc area which is free
vmallocTotal	Total size of vmalloc memory area
vmallocUsed	Amount of vmalloc area used
writeback	Memory being actively written back to the disk
writebackTmp	Memory used by FUSE for temporary writeback buffers

Query

None

Response Body

None

Response Codes

200	OK
400	Bad Request

Examples

```
GET /system/usage/memory
        {
            "active": "66856 kB",
```

```
        "activeAnon": "38180 kB",
        ...
        "writebackTmp": "0 kB"
    }
```

2.2.22 /system/usage/flash

Gets read only system information about flash usage and returns an array of file systems and their individual usage information.

Methods

GET

Parameters

PARAMETER	DESCRIPTION
filesystem	Name of the file system
1kBlocks	Total number of 1k blocks
used	Number of blocks used
available	Number of block available
use	Percent of blocks in use
mountedOn	Directory path where the file system is mounted

Query

None

Response Body

None

Response Codes

200	OK
400	Bad Request

Examples

```
GET /system/usage/flash
    {
        "flashUsage":  [
            {
                "filesystem": "/dev/mmcblk0p8",
                "1kBlocks": "59361",
                ...
            },
            {
                "filesystem": "/dev/mmcblk0p11",
```

```
                ...
            }
        ]
    }
```

2.3 Security Profile

2.3.1 /security

Gets and sets parameters in the security profile of the appliance.

Methods

GET, PUT, PATCH

Parameters

PARAMETER	DESCRIPTION	OPTONS
idleTimeout	Session idle timeout in seconds: integer. 0 disables the timeout entirely, otherwise 90 seconds is the minimum accepted value.	
rpc	RPC service	**enabled/disabled**
pluggableDevices	Pluggable device detection.	**enabled/disabled**
pluggableStorage	Pluggable storage devices.	**enabled/disabled**
access	Serial port access can be configured to allow access for all users, or allow the configuration of group and user-specific authorizations to restrict access.	**all/user_group**
allAccessSettings (only valid when access is all)		
session	Sessions can be set to allow single or multiple session read/write.	**single/multiple**
killMultiSession	Enables you to kill other sharers of a session.	**enabled/disabled**
sendMessageMultiSession	Enables you to send a message to other users sharing a session.	**enabled/disabled**
powerControl	Enables you to control power of a multi session.	**enabled/disabled**
dataBufferManagement	Enables you for data buffer management of a session.	**enabled/disabled**
restfulClientMenu	Enables the restfulClient menu.	**enabled/disabled**
eraseFlash	Control whether entire flash configuration is erased on a factory default.	**enabled/disabled**
bootp	Controls bootp configuration retrieval.	**enabled/disabled**
bootpSettings (only valid when bootp is enabled)		
bootpInterface	Specifies network interface used by bootp.	**eth#**
liveConfigurationRetrieval	Enables live configuration retrieval any time DHCP renews.	**enabled/disabled**
sshUserPass	Controls whether SSH allows authentication via username/password.	**enabled/disabled**
profile	Sets the appliance security profile.	**open/moderate/secure/custom**
customProfile (only valid when profile is custom)		
telnet	Telnet service.	**enabled/disabled**
ftp	FTP service.	**enabled/disabled**

PARAMETER			DESCRIPTION	OPTONS
	snmp		SNMP service.	**enabled/disabled**
	ipsec		IPSec.	**enabled/disabled**
	answerIcmp		Answer ICMP message.	**enabled/disabled**
	sshVersion		SSH Version.	**1 / 2 / 1|2 / 2|1**
	sshPort		SSH TCP port number.	integer (default 22)
	sshRootAccess		SSH allow root access.	**enabled/disabled**
	sshCipherLevel		SSH minimum cipher and mac suite level.	**low/high**
	httpSession		HTTP sessions.	**enabled/disabled**
	httpSettings (only valid when httpSession is enabled)			
		httpPort	HTTP port number.	integer (default 80)
	httpsSession		HTTPS sessions.	**enabled/disabled**
	httpsSettings (only valid when httpsSession is enabled)			
		httpsTlsVersion	HTTPS TLS version.	**1.1 / 1.1|1.2 / 1.2 / 1.1+1.0 / 1.1|1.2+1.0 / 1.2+1.0**
		httpsCipherLevel	HTTPS minimum cipher suite level.	**low/medium/high**
		httpsPort	HTTPS Port number.	integer (default 443)
		redirectHttp	Redirect HTTP/HTTPS.	**enabled/disabled**
consolePort			Appliance console port.	**enabled/disabled**
apiHttpAccess			Allow RESTful API access via HTTP.	**enabled/disabled**
apiHttpSettings (only valid when apiHttpAccess is enabled)				
	apiHttpPort		HTTP port number for RESTful API access.	integer
apiHttpsAccess			Allows RESTful API access via HTTPS.	**enabled/disabled**
apiHttpsSettings (only valid when apiHttpsAccess is enabled)				
	apiHttpsPort		HTTPS port number for RESTful API access.	integer
fips			FIPS 140-2 module. NOTE: Changing the fips value causes the appliance to reboot.	**enabled/disabled**
dsview			Allows appliance to be managed by DSView software.	**enabled/disabled**

Query

Fields are supported for all parameters.

2.3.2 Response Body

JSON object

2.3.3 Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

NOTE: Pluggable Storage Device changes are effective after reboot only.

NOTE: Disabling Pluggable Device Detection is effective after reboot only.

NOTE: Profile changes affecting HTTP and HTTPS terminates all http sessions.

NOTE: Disabling the Console Port can make the appliance inaccessible and should only be done in the most extreme cases.

NOTE: API changes made while using the API may terminate the API session.

Examples

GET /security


```
{
    "idleTimeout": 63955,
    "rpc": "disabled",
    "pluggableDevices": "enabled",
    "pluggableStorage": "disabled",
    "access": "all",
    "allAccessSettings": {
        "session": "multiple",
        "killMultiSession": "enabled",
        "sendMessageMultiSession": "enabled",
        "powerControl": "enabled",
        "dataBufferManagement": "enabled",
        "restfulClientMenu": "enabled"
    },
    "eraseFlash": "disabled",
    "bootp": "enabled",
    "bootpSettings": {
        "bootpInterface": "eth0",
        "liveConfigurationRetrieval": "enabled"
    },
    "sshUserPass": "enabled",
    "profile": "custom",
    "customProfile": {
        "telnet": "enabled",
        "ftp": "disabled",
        "snmp": "disabled",
        "ipsec": "disabled",
        "answerIcmp": "enabled",
        "sshVersion": "1|2",
        "sshPort": 22,
        "sshRootAccess": "enabled",
        "sshCipherLevel": "low",
        "httpSession": "enabled",
        "httpSettings": {
            "httpPort": 80
    },
    "httpsSession": "enabled",
    "httpsSettings": {
        "httpsTlsVersion": "1.1|1.2,
        "httpsCipherLevel": "low",
        "httpsPort": 443,
        "redirectHttp": "disabled"
    }
    },
    "consolePort": "enabled",
    "apiHttpAccess": "enabled",
    "apiHttpSettings": {
        "apiHttpPort": 8080
```

```
            },
            "apiHttpsAccess": "enabled",
            "apiHttpsSettings": {
            "apiHttpsPort": 48048
            },
            "fips": "disabled",
            "dsview": "enabled"
        }
GET /security
        {
            "idleTimeout": 63955,
            "rpc": "disabled",
            "pluggableDevices": "disabled",
            "access": "user_group",
            "eraseFlash": "disabled",
            "bootp": "disabled",
            "sshUserPass": "enabled",
            "profile": "open",
            "consolePort": "enabled",
            "apiHttpAccess": "disabled",
            "apiHttpsAccess": "disabled",
            "fips": "disabled",
            "dsview": "disabled"
        }
GET /security?fields=idleTimeout
        {
            "idleTimeout": 300
        }
GET /security?fields=rpc,pluggableDevices
        {
            "rpc": "disabled",
            "pluggableDevices": "enabled"
        }
PUT /security {"customProfile": {"httpsSettings": {"httpsPort": 444,
"redirectHttp": "disabled"}}}
```


2.4 Network

2.4.1 /network/settings

Get and set various network parameter settings of the console system.

NOTE: When configuring failoverSettings, the primary and secondary interface cannot be set to the same value, so when swapping them the changes need to be made in one PUT command rather than separately.

Methods

GET, PUT, PATCH

Parameters

PARAMETER		DESCRIPTION
hostname		User-defined name of the appliance on the network. Defaults to the "ACS80xx-<serialNumber>" format.
primaryDns		IP address of the primary DNS server.
secondaryDns		IP address of the secondary DNS server.
domain		Domain address. Default: corp.avocent.com.
search		Search address. Default: corp.avocent.com.
lldp		Link Layer Discovery Protocol: **enabled/disabled**.
ipv6		IPv6 support: **enabled/disabled**.
ipv6Settings (only valid when ipv6 is enabled)		
	dhcpv6Dns	Get the IPv6 DNS server address from DHCPv6: **enabled/disabled**.
	dhcpv6Domain	Get the IPv6 Domain name from DHCPv6: **enabled/disabled**.
ipsecFrequency		IPSec Tunnel Check Frequency.
ipsecMaxTime		IPSec Tunnel Check Maximum Time.
multipleRouting		Enable network failover or IPv4 Multiple Routing Tables: **none/enable_network_failover/enable_multiple_routing**.
bonding		
bondingSettings: (only valid when bonding is enabled)		
miiMon		Integer milliseconds.
upDelay		Integer milliseconds.
failoverSettings (only valid when multipleRouting is enable_network_failover).		
	primaryInterface	Primary network interface: **eth#**.
	secondaryInterface	Secondary network interface: **eth#**.
	vpnConnection	
	trigger	Failover trigger: **primary_interface_down/unreachable_primary_default_gateway/ unreachable_ip_address/unreachable_dsview**.
	unreachableIp	IP address to probe when trigger is unreachable_ip_address.

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

GET /network/settings


```
{
    "hostname": "ACS8048-123456789",
    "primaryDns": "10.20.30.40",
    "secondaryDns": "10.20.30.50",
    "domain": "corp.avocent.com",
    "search": "corp.avocent.com",
    "lldp": "disabled",
    "ipv6": "enabled",
    "ipv6Settings": {
        "dhcpv6Dns": "disabled",
        "dhcpv6Domain": "disabled",
    }
"ipsecFrequency": "1m",
"ipsecMaxTime": "24h",
"multipleRouting": "enable_network_failover",
"failoverSettings": {
    "primaryInterface": "eth0",
    "secondaryInterface": "eth1",
    "trigger": "unreachable_ip_address",
    "unreachableIp": "10.20.30.50"
},
"bonding": "disabled"

}
```


PUT /network/settings {"hostname": "myhostname"}

2.4.2 /network/devices[/<INT>]

Get and set various network device parameters for individual network interfaces.

Methods

GET, PUT, PATCH

Parameters

PARAMETER	DESCRIPTION
interface	Read-only name of the network interface: eth0, eth1 and so on.
isPrimary	Read-only. Identifies if this is the primary interface: **enabled/disabled**
status	Status of the interface: **enabled/disabled**

PARAMETER		DESCRIPTION
ipv4Method		Method used to configure IPv4: **dhcp/static/unconfigured**
ipv6Method		Method used to configure IPv6: **stateless/dhcpv6/static/unconfigured**
mac		Read only hardware MAC address of this interface. Example: 00:e0:86:01:02:03
ipv4Static (only valid when ipv4Method is static)		
	ipv4Address	IPv4 address for static configuration. Example: 10.20.30.40
	ipv4Mask	Subnet mask for static configuration. Example: 255.255.255.0
	ipv4Gateway	Gateway to use for static configuration. Example: 10.20.30.1
ipv6Static (only valid when ipv6Method is static)		
	ipv6Address	IPv6 address for static configuration. Example: fd00:0024:0000:0000:02e0:86ff:fe10:2c3b
	ipv6PrefixLength	IPv6 prefix length for static configuration. Example: 112
mtu		Maximum Transmission Unit size in bytes.

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

GET /network/devices

```
{
    "devices": [
        {
            "interface": "eth0",
            "isPrimary": "enabled",
            "status": "enabled",
            "ipv4Method": "dhcp",
            "ipv6Method": "stateless",
            "mac": "00:11:22:33:44:55",
            "mtu": 1500
        },
        {
            "interface": "eth1",
            ...
```

```
            }
        ]
    }
GET /network/devices/eth1
    {
        "interface": "eth1",
        "isPrimary": "disabled",
        "status": "enabled",
        "ipv4Method": "static",
        "ipv6Method": "stateless"
        "mac": "00:11:22:33:44:56",
        "ipv4Static": {
            "ipv4Address": "10.20.30.40",
            "ipv4Mask": "255.255.255.0",
            "ipv4Gateway": ""
        }
        "mtu": 1500
    }
```

2.4.3 /network/hosts[/<ADDRESS>]

Get and set network hosts entries for the local network.

Methods

GET, PUT, PATCH, POST, DELETE

Parameters

PARAMETER	DESCRIPTION
ipAddress	IPv4 or IPv6 IP address for the host; read-only on a PUT/PATCH
hostname	Hostname
alias	Alternate hostname or alias

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

GET /network/hosts

```
{
    "networkHosts": [
        {
            "ipAddress": "127.0.0.1",
            "hostname": "localhost.localdomain",
            "alias": "localhost"
        },
        {
            "ipAddress": "::1",
            "hostname": "localhost",
            "alias": "ip6-localhost ip6-loopback"
        },
        ...
    ]
}
```

GET /network/hosts/10.20.30.40

```
{
    "ipAddress": "10.20.30.40",
    "hostname": "myserver",
    "alias": ""
}
```

GET /network/hosts/fe00::0

```
{
    "ipAddress": "fe00::0",
    "hostname": "ip6-mcastprefix",
    "alias": ""
}
```

POST /network/hosts

```
{
    "ipAddress": "10.20.30.70",
    "hostname": "anotherhost.domain",
    "alias": "anotherhost"
}
```

Response is:

```
{
    "ipAddress": "10.20.30.70",
    "hostname": "anotherhost.domain",
    "alias": "anotherhost"
}
```

PUT /network/hosts/10.20.30.70

```
{
    "hostname": "newhost.domain",
    "alias": "newhost",
```

```
        }
DELETE /network/hosts/10.20.30.70
```

2.4.4 /network/staticRoutes/<Table>[/<ID>]

Get and set static route table entries.

The TABLE portion of the resource is either "ipv4" or "ipv6".

Methods

GET, PUT, PATCH, POST, DELETE

Parameters

PARAMETER	DESCRIPTION
id	Read-only identifier composed of destination and metric fields
destination	Destination IP / Mask bits or "default", for example: "10.20.30.40/32"
gateway	Gateway IP address
interface	eth#
metric	Integer between 0 and 32765

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
201	Created
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

GET /network/staticRoutes/ipv4

```
{
    "staticRoutes": [
        {
            "id": "10.20.30.40/32-123",
            "destination": "10.20.30.40/32",
            "gateway": "10.20.30.41",
            "interface": "eth0",
            "metric": 123
        },
        {
            "id": "default-3",
            "destination": "default",
            "gateway": "1.2.3.4",
            "interface": "eth0",
            "metric": 3
        }
    ]
}
```

GET /network/staticRoutes/ipv4/default-3

```
{
    "id": "default-3",
    "destination": "default",
    "gateway": "1.2.3.4",
    "interface": "eth0",
    "metric": 3
}
```

DELETE /network/staticRoutes/ipv4/default-3

POST /network/staticRoutes/ipv4

```
{
        "destination": "10.20.30.40/32",
        "gateway": "1.2.3.4",
        "interface": "eth1",
        "metric": 123
    }
```

Response is:

```
{
    "id": "10.20.30.40/32-123",
    "destination": "10.20.30.40/32",
    "gateway": "",
    "interface": "eth1",
    "metric": 123
}
```

2.4.5 /network/firewall/<TABLE>[/<CHAIN>]

Create and configure chains for the IPv4 and IPv6 filter tables.

The TABLE portion of the resource is either "ipv4" or "ipv6".

Parameters

PARAMETER	DESCRIPTION
name	Chain name. **FORWARD / INPUT / OUTPUT / ***
policy	Chain policy: **accept/drop**
packets	Integer number of packets processed for built-in chains only
bytes	Integer number of bytes processed for built-in chains only

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

GET /network/firewall/ipv4


```
        {
            "chains": [
                {
                    "name": "FORWARD",
                    "policy": "accept",
                    "packets": 0,
                    "bytes": 0,
                },
                {
                    "name": "INPUT",
                    "policy": "accept",
                    "packets": 124294,
                    "bytes": 15373123,
                },
                {
                    "name": "INPUT",
                    ...
                }
            ]
        }
```


GET /network/firewall/ipv6/FORWARD


```
        {
            "name": "FORWARD",
            "policy": "accept","packets": 0,
            "bytes": 0,
        }
```


POST /network/firewall/ipv4


```
            {
                "name":"mychain"
            }
```


Response is:


```
{
    "name": "mychain",
    "policy": "-",
    "packets": 0,
    "bytes": 0,
}
```


DELETE /network/firewall/ipv4/mychain

2.4.6 /network/firewall/<TABLE>/<CHAIN>rules[/<RULE>]

Configure rules for a firewall chain.

The TABLE portion of the resource is either "ipv4" or "ipv6".

Parameters

PARAMETERS	DESCRIPTION
index	Integer rule index assigned by system for identification
target	**accept/drop/log/reject/return**
source	Source address and mask or prefix. Simple IP or cidr or network with dotted quad netmask.
destination	Destination address and mask or prefix. Simple IP or cidr or network with dotted quad netmask.
protocol	**numeric/tcp/udp/icmp**
(Only used when protocol is tcp):	
tcpSourcePort	Port number, blank, or a colon separated range of port numbers
tcpDestPort	Port number, blank, or a colon separated range of port numbers
tcpFlagSyn	**any/set/unset**
tcpFlagAck	**any/set/unset**
tcpFlagFin	**any/set/unset**
tcpFlagRst	**any/set/unset**
tcpFlagUrg	**any/set/unset**
tcpFlagPsh	**any/set/unset**
(Only used when protocol is udp):	
udpSourcePort	Port number, blank or a colon separated range of port numbers
udpDestPort	Port number, blank, or a colon separated range of port numbers
(Only used when protocol is icmp):	
icmpType	One of the following: **address_mask_reply / address_mask_request / any / bad_ip_header / communication_prohibited / destination_unreachable / echo_reply / echo_request / fragmentation_needed / host_precedence_violation / host_prohibited / host_redirect / host_unknown / host_unreachable / network_prohibited / network_redirect / network_unknown / network_unreachable / parameter_problem / port_unreachable / precedence_cutoff / protocol_unreachable / redirect / required_option_missing / router_advertisement / router_solicitation / source_quench / source_route_failed / time_exceeded / timestamp_reply / timestamp_request / tos_host_redirect / tos_host_unreachable / tos_network_redirect / tos_network_unreachable / ttl_zero_during_reassembly / ttl_zero_during_transit**
inputInterface	**any/lo/eth#**
outputInterface	**any/lo/eth#**
fragments	**all_packets_and_fragments / unfragmented_packets_and_1st_packets / 2nd_and_further_packets**
matchSourceIpDifferent	**enabled/disabled**
matchDestIpDifferent	**enabled/disabled**
matchSourcePortsDifferent	**enabled/disabled**
matchDestPortsDifferent	**enabled/disabled**
matchProtocolsDifferent	**enabled/disabled**
matchTcpFlagsDifferent	**enabled/disabled**
matchIcmpTypesDifferent	**enabled/disabled**
matchInputInterfacesDifferent	**enabled/disabled**
matchOutputInterfacesDifferent	**enabled/disabled**
(Only used when target is log):	

PARAMETERS	DESCRIPTION
logLevel	debug/info/notice/warning/error
logPrefix	n/a
(Only used when target is reject):	
rejectWith	administratively_prohibited / host_ prohibited / host_unreachable / network_ prohibited / network_unreachable / port_unreachable / protocol_unreachable / tcp_reset
packets	Number of packets processed
bytes	Number of bytes processed

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

```
GET /network/firewall/ipv4/FORWARD/rules
{
    "rules": [
        {
            "index": 0,
            "target": "log",
            "sourceIpMask": "",
            ...
        },
        {
            "index": 1,
            ...
        }
    ]
}
GET /network/firewall/ipv4/FORWARD/rules/1
{
    "index": 1,
    "target": "log",
    "sourceIpMask": "10.20.0.0/16",
    "destIpMask": "10.20.30.0/24",
    "protocol": "numeric",
    "protocolNumber": 123,
    "inputInterface": "any",
    "outputInterface": "any",
    "fragments": "all_packets_and_fragments",
```

```
    "matchSourceDifferent": "disabled",
    …
    "logLevel": "debug",
    "logPrefix": "myPrefix"
}
POST /network/firewall/ipv4/mychain/rules
            {
      "target":"accept",
      "sourceIpMask": "10.20.0.0/16",
      "destIpMask": "10.20.30.0/24",
      "protocol": "numeric",
      "protocolNumber": 123,
                  …
            }
```

```
Response is:
{
    "index": 0,
    "target": "accept",
    "sourceIpMask": "10.20.0.0/16",
    "destIpMask": "10.20.30.0/24",
    "protocol": "numeric",
    "protocolNumber": 123,
    "inputInterface": "any",
    "outputInterface": "any",
    "fragments": "all_packets_and_fragments",
    "matchSourceDifferent": "disabled",
    …
}
DELETE /network/firewall/ipv4/mychain/rules/0
```

2.4.7 /network/firewall/<TABLE>/<CHAIN>/rules/<RULE>/move

Moves a firewall rule up or down the list or to a specific location in the list of rules.

The TABLE portion of the resource is either "ipv4" or "ipv6".

Methods

POST

Parameters

PARAMETER	DESCRIPTION
direction	Move up or down by one. up/down
index	Desired integer index, shift accordingly

Query

None

Response Body

None

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

```
POST /network/firewall/ipv4/INPUT/rules/1/move {"direction": "down"}
POST /network/firewall/ipv6/INPUT/rules/3/move {"index": 0}
```

2.4.8 /network/ipsec/connections[/<NAME>]

Configure IPSec connections and settings.

Methods

GET, PUT, PATCH, POST, DELETE

Parameters

PARAMETER	DESCRIPTION
name	IPSec connection name
ikeVersion	KE version. **ike_v1/ike_v2**
bootAction	Boot action; **add/start/ignore**
aggressive	Aggressive; **yes/no**
dpdAction	DPD action; **none/restart**
remoteId	
remoteIpAddress	
remoteSubnet	Subnet IP address and mask in format "10.20.30.00/24"
localId	
localVirtualIp	
localIpAddress	
localSubnet	Subnet IP address and mask in format "10.20.30.00/24"
authenticationMethod	Authentication method. **pre_shared_secret/psk_and_xauth/rsa_certificate**
rsaCertificateFile	
rsaFallback	RSA Fallback. **enabled/disabled**
xauthPreSharedSecret	

PARAMETER	DESCRIPTION
xauthUsername	
xauthPassword	
preSharedSecret	
ikeEncryption	3des/aes128/aes192/aes256/des
ikeHash	sha256/sha384/sha512/sha1/md5
ikeDhGroup	modp768/modp1024/modp1536/modp2048/modp3072/modp4096/modp6144/ modp8192/modp1024s160/modp2048s224/modp2048s256/ecp192/ecp224/ecp256/ecp384/ecp521/ecp224bp/ecp256bp/ecp384bp/ecp512bp
espEncryption	3des/aes128/aes192/aes256/des
espHash	sha256/sha384/sha512/sha1/md5
espDhGroup	modp768/modp1024/modp1536/modp2048/modp3072/modp4096/modp6144/ modp8192/modp1024s160/modp2048s224/modp2048s256/ecp192/ecp224/ecp256/ecp384/ecp521/ecp224bp/ecp256bp/ecp384bp/ecp512bp
reauthentication	yes/no
ikeLifetime	Default: 24h
keyLifetime	Default: 60m
rekey	yes/no
keyingTries	Default: 3
rekeyMargin	Default: 3m
dpdDelay	Default: 300s

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
201	Created
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

```
GET /network/ipsec/connections
{
    "connections": [
        {
            "name": "myconnection",
            "ikeVersion": "ike_v2",
            "bootAction": "add",
```

```
            "aggressive": "no",
            "dpdAction": "none",
            "remoteId": "",
            "remoteIpAddress": "",
            "remoteSubnet": "",
            …
        },
        {
            "name": "anotherconnection",
            "ikeVersion": "ike_v1",
            "bootAction": "add",
            "aggressive": "no",
            …
        },
        …
    ]
}

GET /network/ipsec/connections/myconnection
{
    "name": "myconnection",
    "ikeVersion": "ike_v2",
    "bootAction": "add",
    "aggressive": "no",
    "dpdAction": "none",
    "remoteId": "",
    "remoteIpAddress": "10.20.30.40",
    "remoteSubnet": "",
    "localId": "",
    "localVirtualIp": "",
    "localIpAddress": "10.20.30.41",
    "localSubnet": "",
    "authenticationMethod": "pre_shared_secret",
    "rsaCertificateFile": "",
    "rsaFallback": "disabled",
    "xauthPreSharedSecret": "",
    "xauthUsername": "",
    "xauthPassword": "",
    "preSharedSecret": "",
    "ikeEncryption": "aes256",
    "ikeHash": "sha512",
    "ikeDhGroup": "modp768",
    "espEncryption": "aes256",
    "espHash": "sha512",
    "espDhGroup": "modp2048",
    "reauthentication": "no",
    "ikeLifetime": "23h",
    "keyLifetime": "60m",
    "rekey": "yes",
    "keyingTries": "3",
    "rekeyMargin": "3m",
    "dpdDelay": "300s"
}

DELETE /network/ipsec/connections/myconnection

PUT /network/ipsec/connections/myconnection
{
```

```
    "remoteId": "mine",
    "remoteIpAddress": "10.20.30.40",
    "remoteSubnet": "10.20.30.00/24"
}

POST /network/ipsec/connections
{
    "name": "myconnection",
    "remoteId": "mine",
    "remoteIpAddress": "10.20.30.40",
    "remoteSubnet": "10.20.30.00/24"
}
Response is:
{
    "name": "myconnection",
    "ikeVersion": "ikev2",
    "bootAction": "ignore",
    "aggressive": "no",
    "dpdAction": "none",
    "remoteId": "mine",
    "remoteIpAddress": "10.20.30.40",
    "remoteSubnet": "10.20.30.00/24",
    ...
}
```

2.4.9 /network/ipsec/certificates

List and delete IPSec PKCS12 files.

Methods

GET, DELETE

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
GET /network/ipsec/certificates
{
    "ipsecCertificates": [
        {
            "name": "cert1.p12",
        },
        {
            "name": "cert2.p12",
        }
    ]
}

DELETE /network/ipsec/certificates/cert1.p12
```

2.4.10 /network/ipsec/certificates/download

This action downloads an IPSec certificate for the appliance from a remote server.

Methods

POST

Parameters

PARAMETER	DESCRIPTION
protocol	Specifies the protocol to use to download file: **ftp/scp/sftp**
ipAddress	IP address of the remote server from which to download the file
username	Username to access the remote server
password	Password to access the remote server
directory	Directory path on the remote server, typically relative to the ftp root directory
filename	Filename of the certificate file on the remote server

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
POST /network/ipsec/certificates/download
{
    "protocol": "ftp",
    "ipAddress": "10.20.30.80",
    "username": "anonymous",
    "password": "anonymous",
    "directory": "pub/certificate/",
    "filename": "mycertificate.p12"
}
Response is:
{
    "status": "ipsec certificate download successful",
}
```

2.4.11 /network/snmp[/<ID>]

Configure SNMP settings.

Methods

GET, PUT, PATCH, POST, DELETE

Parameters

PARAMETERS	DESCRIPTION
id	Read-only identifier composed of name and potentially source fields
name	SNMP community name for v1/v2 or the username for v3
version	version SNMP version. **v1_v2/ipv6_v1_v2/v3**
source	Subnet address for v1/v2.
oid	Unique snmp identifier
permission	Access permission. **read_and_write/read_only**
authenticationType	SNMP v3 authentication type. **md5/sha**
authenticationPassphrase	SNMP v3 authentication passphrase
encryptionMethod	SNMP v3 encryption method. **des/aes**
privacyPassphrase	SNMP v3 privacy passphrase
minimumSecurityLevel	SNMP v3 minimum security level. **auth_no_priv/auth_priv/no_auth_no_priv**

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
201	Created
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

GET /network/snmp

```
{
    "snmp": [
        {
            "id": "mycommunity~10.20.30.40",
            "name": "mycommunity",
            "version": "v1_v2",
            "source": "10.20.30.40",
            "oid": "",
            "permission": "read_only"
        },
        {
            "id": "myusername",
            "name": "myusername",
            "version": "v3",
            "permission": "read_only",
            "authenticationType": "md5",
            "authenticationPassphrase": ****,
            "encryptionMethod": "des",
            "privacyPassphrase": ****,
            "minimumSecurityLevel": "no_auth_no_priv"
        }
        ...
    ]
}
```

GET /network/snmp/1

```
{
    "id": "mycommunity~10.20.30.40",
    "name": "mycommunity",
    "version": "v1_v2",
    "source": "10.20.30.40"
    "oid": "",
    "permission": "read_only",
}
```

DELETE /network/snmp/mycommunity~10.20.30.40

GET /network/snmp/myusername

```
{
    "id": "myusername",
    "name": "myusername",
    "version": "v3",
    "oid": "",
```

```
    "permission": "read_only",
    "authenticationType": "md5",
    "authenticationPassphrase": ****,
    "encryptionMethod": "des",
    "privacyPassphrase": ****,
    "minimumSecurityLevel": "no_auth_no_priv"
}
POST /network/snmp
{
      "name": "mycommunity",
      "oid": "",
      "permission": "read_only",
      "version": "v1_v2",
      "source": "10.20.30.40"
}
Response is:
{
    "id": "mycommunity~10.20.30.40",
    "name": "mycommunity",
    "version": "v1_v2",
    "source": "10.20.30.40"
    "oid": "",
    "permission": "read_only",
}
PUT /network/snmp/mycommunity~10.20.30.40
{
    "oid": ".1",
    "permission": "read_write",
}
```

2.4.12 /network/snmp/system

Configure SNMP system settings.

Methods

GET, PUT, PATCH

PARAMETER	DESCRIPTION
sysContact	SNMP system contact information. Defaults to Avocent_Corporation
sysLocation	SNMP system location information. Defaults to Avocent_ACS8000

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

```
GET /network/snmp/system
{
    "sysContact": "Avocent_Corporation",
    "sysLocation": "Avocent_ACS8000"
}
PUT /network/snmp/system
{
      "sysContact": "corporate IT @ 1-800-xxx-yyyy",
      "sysLocation": "Server room 123"
}
```

2.5 Serial Ports

2.5.1 /serialPorts[/<PORT#>]

Get and set various serial port parameters for one or more serial ports.

Methods

GET, PUT, PATCH

Parameters

PARAMETER	DESCRIPTION	OPTIONS
port	Read-only port number.	
profile	Port profile. Note: Unconfigured is a read-only state that cannot be set.	**cas/power/dial_in/dial_out/socket_client/unconfigured**
deviceName	Read-only device name assigned by the operating system. Examples: ttyS1, ttyUSB0, ttyACM0.	Examples: ttyS1, ttyUSB0, ttyACM0
status	Port status	**enabled/disabled**
physical (only valid when the profile is cas, power or socket_client)		
pinout	Serial port pinout.	**auto/cisco/cyclades/RS422/rRS485**
speed	Serial port speed: integer	**1200/2400/4800/9600/19200/38400/57600/115200/230400**
parity	Parity.	**even/odd/none**
dataBits	Number of data bits: integer.	**5/6/7/8**
stopBits	Number of stop bits.	integer **1/2**
flowControl	Flow control mechanism.	**none / hardware / software/rxon_software / txon_software**
cas (only valid when the profile is cas)		

PARAMETER	DESCRIPTION	OPTIONS
name	User-assigned or default name for the port. Default is the last half of the appliance MAC address followed by "-p-" and the port number.	Example: 10-2c-3d-p-1
autoDiscovery	Enables auto discovery. The target name is discovered and associated with this serial port. If it fails, the default port name is used.	**enabled/disabled**
speedAutoDetection	Enables speed auto detection to try to discover the speed of the serial port.	**enabled/disabled**
protocol	The protocol used by authorized users to access the serial port or target.	**ssh / telnet / raw_mode / telnet|ssh / telnet|raw_mode / ssh|raw_mode / telnet|ssh|raw_mode**
authentication	Type used to authenticate a user during the target session.	**none / dsview_down_local / dsview / dsview|local / ldap_ down_local / ldap / ldap|local / local / local|radius / local|tacacs+ / otp / otp|local / radius_down_local / radius / radius|local / tacacs+_down_local / tacacs+ / tacacs+|local**
textHotKey	Hotkey to suspend the target session and return to the cli prompt. Not available for raw.	Default is Ctrl-Z (^Z)
powerHotKey	Hotkey to suspend the target session and display the Power Management Menu to control the outlets merged to the target. Not available for raw.	Default is Ctrl-P (^P)
restfulHotKey	Hotkey to suspend the target session and display the RESTful Client Menu, which is used to send user-defined RESTful actions to a RESTful server.	Default is not configured
telnetAliasPort	TCP port used to connect directly to a serial port using Telnet protocol.	Default is 7000+port number
sshAliasPort	TCP port used to connect directly to a serial port using SSH protocol.	Default is not configured
rawModeAliasPort	TCP port used to connect directly to a serial port using raw socket for connection.	Default is not configured
ipv4AliasAddress	IPv4 address used to connect directly to a serial port.	
ipv4AliasInterface	Interface associated with the ipv4AliasAddress.	**eth#**
ipv6AliasAddress	IPv6 address used to connect directly to a serial port.	
ipv6AliasInterface	Interface associated with the ipv6AliasAddress.	**eth#**
dcdSensitivity	Allows a session only if DCD is on.	**enabled/disabled**
autoAnswer	Enables processing of input data so when the input data matches one input string configured in Auto Answer, the configured output string is transmitted to the serial port.	**enabled/disabled**
dtrMode	Sets DTR mode.	**always_on** - DTR status is always on **normal** (default) - DTR status depends on the existence of a CAS session **off_interval** - when a CAS session is closed; DTR stays down during this interval
dtrOffInterval	Interval used by DTR Mode off_interval in milliseconds.	integer, Default: 100
linefeedSuppression	Enables the suppression of the line-feed character after the carriage return character.	**enabled/disabled**
nullAfterCrSuppression	Enables the suppression of the NULL character after the carriage return character.	**enabled/disabled**
transmissionInterval	The interval the port waits to send data to a remote client in milliseconds.	integer

PARAMETER		DESCRIPTION	OPTIONS
	breakSequence	An administrator can configure the control key as the break sequence, entering ^ before the letter.	Default: ~break
	breakInterval	Interval for the break signal in milliseconds.	integer, Default: 500
	multiSessionMenu	Enables the multi-session menu when connecting to a port that is already being accessed by another user.	**enabled/disabled**
	loginNotification	Enables the notification to multi-session users when a user logs in or logs out.	**enabled/disabled**
	infoNotification	Displays an information message when a target session is opened.	**enabled/disabled**
cas/dataBuffering (only valid when the profile is cas)			
	bufferingStatus	Enables or disables data buffering.	**enabled/disabled**
	bufferingType	Controls the type of data buffering:	**local** - stores the data buffering file on the appliance's local file system **nfs** - stores the data buffering file on an NFS server **syslog** - sends the data to the syslog server **dsview** - sends the data to the DSView server.
	localDevice	When the bufferingType is set to local, this field specifies where on the local system the data buffering files are stored.	Options are the built-in memory (**mmcblk0**) or a connected USB storage or SD card location.
	timeStamp	When enabled, adds the time stamp to the data buffering line for a local or NFS server.	**enabled/disabled**
	loginMessage	Includes special notification for logins and logouts in data buffering.	**enabled/disabled**
	sessionLogging	Controls when data is stored.	**enabled/disabled**
power (only valid when the profile is power)			
	speedAutoDetect	Enables speed auto detect for power device. Tries to discover the speed of the attached power device.	**enabled/disabled**
	pollingRate	Interval in seconds to update information from the PDU	integer, Default: 20
	deviceType	Type of power device connected to the serial port.	**pdu/ups**
	pduType	Defines the type or vendor of the PDU connected to the serial port.	**auto/cyclades/enp/spc/servertech/raritan/apc/eaton**
	powerCycleInterval	The interval in seconds between Off and On actions for the power cycle command.	integer, Default: 15
	syslog	When enabled, the PDU sends syslog messages to the appliance.	**enabled/disabled**
	buzzer	Enables or disables the PDU's buzzer.	**enabled/disabled**
	overcurrentProtection	When enabled, the software's overcurrent protection is on.	**enabled/disabled**
	upsType	Defines the type or vendor of the UPS connected to the serial port.	**gxt4**
socketClient (only valid when the profile is socket_client)			
	remoteServer	IPv4 or IPv6 address of the remote server.	
	remoteTcpPort	TCP port to be used to establish a connection with a remote server.	
	establishConnection	Configures the event that triggers the establishment of the connection.	**dcd/always**

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

GET /serialPorts/1

```
{
    "port": "1",
    "profile": "cas",
    "deviceName": "ttyS1",
    "status": "enabled",
    "physical": {
        "pinout": "cisco",
        "speed": 115200,
        "parity": "none",
        "dataBits": 8,
        "stopBits": 1,
        "flowControl": "none"
    },
    "cas": {
        "name": "MyServer",
        "autoDiscovery": "disabled",
        ...
    }
}
```

GET /serialPorts

```
        {
            "serialPorts": [
        {
            "port": "1",
            "profile": "cas",
            "deviceName": "ttyS1",
            "status": "enabled",
            ...
        },
        ...
        {
             "port": "48",
            ...
        }
    ]
}
```

PUT /serialPorts/1 '{"cas":{"sshAliasPort":"8001"}}'

```
GET /serialPorts/75
        {
            "error": {
                "code": "AE002",
                "message": "resource id not found",
                "detail": "75 is not a valid port id"
    }
}
```

2.5.2 /serialPorts/<PORT#>/enable

Enable the serial port.

Methods

POST

Parameters

None.

Query

None.

Response Body

JSON object.

Response Codes

200	OK
400	Bad Request

Examples

```
POST /serialPorts/3/enable
```

Response is:

```
{
"status": "success. port 3 enabled."
}
```

2.5.3 /serialPorts/<PORT#>/disable

Disable the serial port.

Methods

POST

Parameters

None.

Query

None.

Response Body

JSON object.

Response Codes

200	OK
400	Bad Request

Examples

POST /serialPorts/3/disable

Response is:

```
{
"status": "success. port 3 disabled."
}
```

2.5.4 /serialPorts/<PORT#>/resetToFactory

Reset the serial port to factory defaults.

Methods

POST

Parameters

None.

Query

None.

Response Body

JSON object.

Response Codes

200	OK
400	Bad Request

Examples

POST /serialPorts/3/resetToFactory

Response is:

```
{
"status": "success. port 3 restored to factory."
}
```

2.5.5 /serialPorts/<PORT#>/alerts

Configure alert strings.

Methods

GET, POST

Parameters

PARAMETER	DESCRIPTION
index	Read-only index.
alertString	String to match.
scriptName	Name of the script to run when the matching alertString is detected. This script should be placed in the /etc/alerts_scripts folder. The script name should be the script name only, with a '.sh' extension and no folder path.
emergency	If set to 'yes' and an alert occurs, the serial port LED blinks amber quickly. **yes/no**

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

GET /serialPorts/3/alerts

Response:

```
{
    "alertStrings": [
        {
            "index": 1,
            "alertString": "Invalid",
            "scriptName": "sample_script.sh",
            "emergency": "no"
        },
        {
            "index": 2,
            "alertString": "Error occurred",
            "scriptName": "sample_script.sh",
            "emergency": "yes"
        },
    ]
}
```

To add a new alert string:

```
POST /serialPorts/3/alerts
{
            "alertString": "new string",
            "scriptName": "new_script.sh",
            "emergency": "yes"
}
```

2.5.6 /serialPorts/<PORT#>/alerts/<ID>

Delete an alert string. The ID is the index of the alert string.

Methods

DELETE

Parameters

None

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

DELETE /serialPorts/3/alerts/1

2.5.7 /serialPorts/<PORT#>/alerts/clear

Clear active alerts on a serial port.

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

POST /serialPorts/3/alerts/clear

Response:

```
{
    "status": "success. alerts cleared."
}
```

2.5.8 /serialPorts/<PORT#>/alerts/deleteAny

Deletes all alert strings for the port.

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

POST /serialPorts/3/alerts/deleteAny

Response:

```
{
    "status": "success. alert strings deleted."
}
```

2.5.9 /serialPorts/<PORT#>/power

Merge PDU outlets or UPS outlet groups with the serial port, and view all merged outlets.

NOTE: Either PDU outlets or UPS outlet groups can be merged with a serial port, but not a mix of both.

Methods

GET, POST

Parameters

PARAMETER	DESCRIPTION
Id	Read-only Id for the outlet consisting of the PDU name and the outlet number, or the UPS name and the UPS outlet group number, in this format: **devicename~outletnumber**
deviceType	Identifies the power device: **pdu** or **ups**
deviceId	Name of the PDU or UPS
outlet	PDU outlet number, or UPS outlet group number

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

GET /serialPorts/1/power


```
{
    "power": [
        {
            "id": "MPX~3",
            "deviceType": "pdu",
            "deviceId": "MPX",
            "outlet": "3"
        },
        {
            "id": "MPX~4",
            "deviceType": "pdu",
            "deviceId": "MPX",
            "outlet": "4"
        }
    ]
}
```


GET /serialPorts/2/power


```
{
    "power": [
        {
            "id": "GXT4~1",
            "deviceType": "ups",
            "deviceId": "GXT4",
            "outlet": "1"
        },
        {
```

```
            "id": "GXT4~2",
            "deviceType": "ups",
            "deviceId": "GXT4",
            "outlet": "2"
        }
    ]
}
```

To merge a new outlet with the serial port:

```
POST /serialPorts/1/power
        {
        "deviceType": "pdu",
        "deviceId": "MPX",
        "outlet": "5"
        }
```

Response:

```
{
    "deviceType": "pdu",
    "deviceId": "MPX",
    "outlet": "5"
}
```

```
POST /serialPorts/2/power
        {
        "deviceType": "ups",
        "deviceId": "GXT4-1000",
        "outlet": "1"
        }
```

Response:

```
{
    "deviceType": "ups",
    "deviceId": "GXT4-1000",
    "outlet": "1"
}
```

2.5.10 /serialPorts/<PORT#>/power/<ID>

Delete merged PDU outlets or UPS outlet groups. The ID is in the devicename~outletnumber format.

Methods

DELETE

Parameters

None

Response Body

JSON object

Response Codes

200	OK
201	Created
400	Bad Request

Examples

```
DELETE /serialPorts/1/power/MPX~5
```

2.6 Aux Port

2.6.1 /auxPorts[/<NAME>]

Get and set Auxiliary Port parameters. There is a single Auxiliary port when a built-in modem is present.

Methods

GET, PUT, PATCH

Parameters

PARAMETER	DESCRIPTION		
deviceName	Ex: ttyM1		
status	**enabled/disabled**		
profile	**unconfigured/dial_in/dial_out**		
phoneNumber	Dial-Out only.		
speed	AuxPort speed: integer **1200/2400/4800/9600/19200/38400/57600/115200**		
initChat	Dial-In Default: "" \d\d\d+++\d\d\dATZ OK Dial-Out Default:		
		ABORT ABORT ABORT ABORT ABORT " " OK CONNECT	BUSY VOICE "NO CARRIER" "NO DIAL TONE: "NO DIAL TONE" ATZ ATDT\T "" ""
pppAddressConfig	**none, local, remote** (Dial-Out profile only supports local)		

PARAMETER	DESCRIPTION
localIpv4Address	IPv4 address of the local side of the ppp connection when pppAddressConfig is local.
remoteIpv4Address	IPv4 address of the remote side of the ppp connection when pppAddressConfig is local.
localIpv6Address	IPv6 address of the local side of the ppp connection when pppAddressConfig is local.
remoteIpv6Address	IPv6 address of the remote side of the ppp connection when pppAddressConfig is local.
pppAuthentication	**none/appliance/remote**
pppAuthenticationProtocol	**pap/chap/eap** (eap not supported for remote authentication)
pppRemoteUsername	The username for remote authentication.
pppRemotePassphrase	The passphrase for remote authentication.
chapInterval	Integer number of seconds. Default: 0
chapMaxChallenge	Default: 10
chapRestart	Integer number of seconds. Default: 3
pppIdleTimeout	Integer number of seconds. Default: 0

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

GET /auxPorts

```
{
    "auxPorts": [
        {
            "deviceName”: "ttyM1",
            "status": "disabled",
            "profile": "unconfigured"
        }
    ]
}
```

GET /auxPorts

```
{
    "deviceName”: "ttyM1",
    "status": "disabled",
    "profile": "dial_in",
```

```
            "speed": 38400,
            "initChat": "\"\" \d\d\d+++\d\d\dATZ OK",
            "pppAddressConfig": "none",
            "pppAuthentication": "none",
            "chapInterval": 0,
            "chapMaxChallenge": 10,
            "chapRestart": 3,
            "pppIdleTimeout": 0
        }
```

```
PUT /auxPorts/ttyM1
        {
            "status": "enabled",
            "profile": "dial_out",
            "phoneNumber": "18001234567",
            "speed": 38400,
            "initChat": "ABORT\tBUSY\nABORT\tVOICE\n",
            "localIpv4Address": "10.20.30.40",
            "remoteIpv4Address": "10.20.30.41",
            "localIpv6Address": "",
            "remoteIpv6Address": "",
            "pppAuthentication": "remote",
            "pppAuthenticationProtocol": "pap",
            "pppRemoteUsername": "testuser",
            "pppRemotePassphrase": "mypassphrase",
            "chapInterval": 0,
            "chapMaxChallenge": 10,
            "chapRestart": 3,
            "pppIdleTimeout": 0
        }
```

2.7 Modems

2.7.1 /modems[/<NAME>]

Get and set Modem parameters. This provides a single resource to configure modem profiles of dial-in and dial-out for all such devices: auxiliary port built-in modem, pluggable USB modems and serial port modems.

Methods

GET, PUT, PATCH

Parameters

PARAMETER	DESCRIPTION
deviceName	Ex: ttyM1
status	**enabled/disabled**
profile	unconfigured/dial_in/dial_out
phoneNumber	Dial-Out only.
speed	AuxPort speed: integer **1200/2400/4800/9600/19200/38400/57600/115200**

PARAMETER	DESCRIPTION	
initChat	Dial-In Default: "" \d\d\d+++\d\d\dATZ OK Dial-Out Default:	
	ABORT ABORT ABORT ABORT ABORT " " OK CONNECT	BUSY VOICE "NO CARRIER" "NO DIAL TONE: "NO DIAL TONE" ATZ ATDT\T "" ""
pppAddressConfig	**none, local, remote** (Dial-Out profile only supports local)	
localIpv4Address		
remoteIpv4Address		
localIpv6Address		
remoteIpv6Address		
pppAuthentication	**none/appliance/remote**	
pppAuthenticationProtocol	**pap/chap/eap** (eap not supported for remote authentication)	
pppRemoteUsername		
pppRemotePassphrase		
chapInterval	Integer number of seconds. Default: 0	
chapMaxChallenge	Default: 10	
chapRestart	Integer number of seconds. Default: 3	
pppIdleTimeout	Integer number of seconds. Default: 0	

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

```
GET /modems
        {
            "modems": [
                {
                    "deviceName": "ttyM1",
```

```
                "status": "disabled",
                "profile": "unconfigured"
            },
            {
                "deviceName": "ttyACM0",
                "status": "disabled",
                "profile": "dial_in",
                …
            },
            {
                "deviceName": "ttyS18",
                "status": "disabled",
                "profile": "dial_out",
                …
            }
        ]
    }
```

GET /modems/ttyM1

```
    {
        "deviceName”: "ttyM1",
        "status": "disabled",
        "profile": "dial_in",
        "speed": 38400,
        "initChat": "\"\" \d\d\d+++\d\d\dATZ OK",
        "pppAddressConfig": "none",
        "pppAuthentication": "none",
        "chapInterval": 0,
        "chapMaxChallenge": 10,
        "chapRestart": 3,
        "pppIdleTimeout": 0
    }
```

PUT /modems/ttyM1

```
        {
            "status": "enabled",
            "profile": "dial_out",
            "phoneNumber": "18001234567",
            "speed": 38400,
            "initChat": "ABORT\tBUSY\nABORT\tVOICE\n",
            "localIpv4Address": "10.20.30.40",
            "remoteIpv4Address": "10.20.30.41",
            "localIpv6Address": "",
            "remoteIpv6Address": "",
            "pppAuthentication": "remote",
            "pppAuthenticationProtocol": "pap",
            "pppRemoteUsername": "testuser",
            "pppRemotePassphrase": "mypassphrase",
            "chapInterval": 0,
            "chapMaxChallenge": 10,
            "chapRestart": 3,
            "pppIdleTimeout": 0
        }
```

2.8 CAS Profile

2.8.1 /casProfile

Get and set the CAS Profile parameters.

Methods

GET, PUT, PATCH

Parameters

PARAMETER	DESCRIPTION
autoDiscoveryTimeout	Integer seconds
probeTimeout	Integer seconds
defaultDiscoverySpeed	Serial port speed: integer **1200/2400/4800/9600/19200/38400/57600/115200/230400**
probe1200	**enabled/disabled**
probe2400	**enabled/disabled**
probe4800	**enabled/disabled**
probe9600	**enabled/disabled**
probe19200	**enabled/disabled**
probe38400	**enabled/disabled**
probe57600	**enabled/disabled**
probe115200	**enabled/disabled**
probe230400	**enabled/disabled**

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

GET /casProfile

```
{
    "autoDiscoveryTimeout": 60,
    "probeTimeout": 3,
    "defaultDiscoverySpeed": 9600,
```

```
            "probe1200": "enabled",
            "probe2400": "disabled",
            "probe4800": "disabled",
            "probe9600": "enabled",
            "probe19200": "disabled",
            "probe38400": "disabled",
            "probe57600": "enabled",
            "probe115200": "enabled",
            "probe230400": "disabled"
        }
```

```
PUT /casProfile
        {
                "autoDiscoveryTimeout": 60,
                "probeTimeout": 3,
                "defaultDiscoverySpeed": 9600,
                "probe1200": "enabled",
                "probe2400": "disabled",
                "probe4800": "disabled",
                "probe9600": "enabled",
                "probe19200": "disabled",
                "probe38400": "disabled",
                "probe57600": "enabled",
                "probe115200": "enabled",
                "probe230400": "disabled"
            }
```

2.8.2 /casProfile/probeStrings

Add and remove probe strings used by auto discovery.

Methods

GET, POST, DELETE

Parameters

PARAMETERS	DESCRIPTION
index	Read-only numeric index
probeString	Example: "\r"

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
201	Created
400	Bad Request

Examples

GET /casProfile/probeStrings

```
{
    "probeStrings": [
        {
            "index": 0,
            "probeString": "\r"
        },
        {
            "index": 1,
            "probeString": "bob"
        }
        ...
    ]
}
```

GET /casProfile/probeStrings/0

```
{
        "index": 0,
        "probeString": "\r"
}
```

POST /casProfile/probeStrings

```
{
        "probeString": "test"
}
```

Response is:

```
{
        "index": 2,
        "probeString": "test"
}
```

DELETE /casProfile/probeStrings/2

2.8.3 /casProfile/matchStrings

Add and remove match strings used by auto discovery.

Methods

GET, POST, DELETE

Parameters

PARAMETERS	DESCRIPTION
index	Read-only numeric index
matchString	Example: "%H!login:"

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
201	Created
400	Bad Request

Examples

GET /casProfile/matchStrings

```
{
    "matchStrings": [
        {
            "index": 0,
            "matchString": "%H.*ogin:"
        },
        {
            "index": 1,
            "matchString": "%H!login:"
        },
        {
            "index": 2,
            "matchString": "%H#"
        },
        {
            "index": 3,
            "matchString": "%H>"
        }
    ]
}
```

GET /casProfile/matchStrings/1

```
{
        "index": 1,
```

```
        "matchString": "%H!login:"
    }
```

```
POST /casProfile/matchStrings
    {
        "matchString": "%H login:"
    }
```

Response is:

```
    {
        "index": 4,
        "matchString": "%H login:"
    }
```

```
DELETE /casProfile/matchStrings/4
```

2.8.4 /casProfile/autoAnswer

Add and remove auto answer strings pairs.

Methods

GET, POST, DELETE

Parameters

PARAMETERS	DESCRIPTION
index	Read-only numeric index
inputString	Input string to compare against incoming data
outputString	Output string to send in response to matching input data

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
201	Created
400	Bad Request

Examples

GET /casProfile/autoAnswer

```
{
    "autoAnswer": [
        {
            "index": 0,
            "inputString": "myinput",
            "outputString": "myoutput"
        },
        {
            "index": 1,
            "inputString": "yourinput",
            "outputString": "youroutput"
        }
        ...
    ]
}
```

GET /casProfile/autoAnswer/0

```
{
    "index": 0,
    "inputString": "myinput",
    "outputString": "myoutput"
}
```

POST /casProfile/autoAnswer

```
{
    "inputString": "testinput",
    "outputString": "testoutput"
}
```

Response is:

```
{
    "index": 2,
    "inputString": "testinput",
    "outputString": "testoutput"
}
```

DELETE /casProfile/autoAnswer/2

2.9 Dial-In Profile

2.9.1 /dialinProfile

This resource provides the ability to get and set the Dial-In Profile settings.

Methods

GET, PUT, PATCH

Parameters

PARAMETER	DESCRIPTION
loginToAppliance	Allow dial-in login to appliance. **enabled/disabled/callback**
otpLoginAuth	OTP login authentication. **enabled/disabled**
pppConnection	PPP connection. **enabled/disabled/callback**
pppPapAuth	PPP/PAP authentication: **dsview_down_local / dsview / dsview|local / kerberos_down_local / kerberos / kerberos|local / ldap_down_local / ldap / ldap|local / local / local|radius / local|tacacs+ / otp / otp|local / radius_down_local / radius / radius|local / tacacs+_down_local / tacacs+ / tacacs+|local**

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

GET /dialinProfile

```
{
    "loginToAppliance": "disabled",
    "otpLoginAuth": "disabled",
    "pppConnection": "disabled",
    "pppAuth": "local"
}
```

PUT /dialinProfile

```
{
    "loginToAppliance": "enabled",
}
```

2.9.2 /dialinProfile/callbackUsers[/<NAME>]

Add and remove callback users and their numbers from the Dial-In Profile.

Methods

GET, PUT, PATCH, POST, DELETE

Parameters

PARAMETER	DESCRIPTION
user	Example: "bob"
number	Example: "8001234567"

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
201	Created
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

```
GET /dialinProfile/callbackUsers
{
    "callbackUsers": [
        {
            "user": "bob",
            "number": "8001234567"
        },
        {
            "user": "joe",
            "number": "8004567890"
        }
        ...
    ]
}
GET /dialinProfile/callbackUsers/bob
{
    "user": "bob",
    "number": "8001234567"
}
PUT /dialinProfile/callbackUsers/bob
{
```

```
    "number": "8001234567"
}

POST /dialinProfile/callbackUsers
{
    "user": "test",
    "number": "8001238378"
}
Response is:
{
  "user": "test",
  "number": "8001238378"
}
DELETE /dialinProfile/callbackUsers/test
```

2.10 /dialinProfile/pppOtpUsers[/<NAME>]

Add and remove PPP OTP users from the Dial-In Profile.

Methods

GET, POST, DELETE

Parameters

PARAMETER	DESCRIPTION
user	PPP OTP username. Example: "bob"
passphrase	Write-only value. Example: "thisisaphrase"
seed	Read-only value. Example: "AC9491"

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
201	Created
400	Bad Request

Examples

```
GET /dialinProfile/pppOtpUsers
{
  "pppOtpUsers": [
```

```
        {
            "user": "bob",
            "seed": "AC1234"
        },
        {
            "user": "joe",
            "seed": "AC1240"
        }
        ...
    ]
}
GET /dialinProfile/pppOtpUsers/bob
{
    "user": "bob",
    "seed": "AC1234"
}
POST /dialinProfile/pppOtpUsers
{
     "user": "test",
     "passphrase": "thisisatest"
}
Response is:
{
    "user": "test",
    "seed": "AC6949"
}
DELETE /dialinProfile/pppOtpUsers/test
```

2.11 Pluggable Devices

2.11.1 /pluggableDevices[/<NAME>]

Read information about the attached pluggable devices (USB, SD card).

Methods

GET

Parameters

PARAMETER	DESCRIPTION
deviceName	Linux assigned name of the USB device. (For example, ttyACM0, ttyUSB0)
deviceType	Device's type. console/ethernet/modem/storage/wirelessModem
card	Physical device type: mmc SD / usb usbslot
devicePath	USB device path to uniquely identify where the device is in the USB tree.
deviceInfo	Device info available via USB device descriptors.
status	Current state of the device. inserted/unmounted/ejected
port	Serial port number assigned to this device if it is enabled as a console.

Query

Fields are supported for all parameters.

Response Codes

200	OK
400	Bad Request

Examples

GET /pluggableDevices

```
{
    "pluggableDevices": [
        {
            "deviceName": "ttyACM0",
            "deviceType": "Console",
            "card": "usb usbslot",
            "devicePath": "1-1.3",
            "deviceInfo": "…",
            "status": "ejected",
            "port": "49"
        },
        …
    ]
}
```

GET /pluggableDevices/ttyUSB0

```
{
    "deviceName": "ttyUSB0",
    "deviceType": "console",
    "card": "usb usbslot",
    "devicePath": "1-1.4",
    "deviceInfo": "…",
    "status": "inserted",
    "port": "49"
}
```

GET /pluggableDevices/mmcblk1p1

```
{
    "deviceName": "mmcblk1p1",
    "deviceType": "stroage",
    "card": "mmc SD",
    "devicePath": "",
    "deviceInfo": " ",
    "status": "inserted",
}
```

2.11.2 /pluggableDevices/<NAME>/setConsole

Sets up the specified pluggable device as a console port. This adds the device to the list of serial ports by adding a new serial port number. The serial port must then be configured appropriately. This can only be done for devices that show a deviceType of "console".

Methods

POST

Parameters

None

Query

None

Response Codes

200	OK
400	Bad Request

Request and Response Examples

```
POST /pluggableDevices/ttyUSB0/setConsole
    {
        "status": "success. ttyUSB0 set to console, port 49"
    }
```

2.11.3 /pluggableDevices/<NAME>/eject

Eject the specified pluggable device so that it can be physically removed without causing loss of data. For a storage device, this makes sure the device is not busy.

Methods

POST

Parameters

None

Query

None

Response Codes

200	OK
400	Bad Request

Examples

```
POST /pluggableDevices/sda1/eject
    {
        "status": "It is now safe to physically unplug the sda1 device"
    }
```

2.11.4 /pluggableDevices/<NAME>/delete

Deletes the specified pluggable device after it has been safely ejected and physically unplugged.

Methods

POST

Parameters

None

Query

None

Response Codes

200	OK
400	Bad Request

Examples

```
POST /pluggableDevices/ttyUSB1/delete
        {
            "status": "success. ttyUSB1 deleted"
        }
```

2.12 Authentication

2.12.1 /authentication

Get and set the various appliance authentication parameters.

Methods

GET, PUT, PATCH

Parameters

PARAMETER	DESCRIPTION
applianceAuthType	Type of authentication to use to grant access to the appliance. **dsview_down_local / dsview / dsview|local / ldap_down_local / ldap / ldap|local / local / local|radius / local|tacacs+ / otp / otp|local / radius_down_local / radius / radius|local / tacacs+_down_local / tacacs+ / tacacs+|local**
singleSignOn	Enables single sign-on. Uses the specified single sign-on authentication and no further authentication is needed when accessing a port. **enabled/disabled**
singleSignOnAuthType	Authentication to use for single sign-on. **unconfigured / dsview_down_local / dsview / dsview|local / ldap_down_local / ldap / ldap|local / local / local|radius / local|tacacs+ / otp / otp|local / radius_down_local / radius / radius|local / tacacs+_down_local / tacacs+ / tacacs+|local**

Query

Fields are supported for all parameters.

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

```
GET /authentication
        {
            "applianceAuthType": "local",
            "singleSignOn": "disabled",
            "singleSignOnAuthType": "unconfigured"
        }
PUT /authentication { "singleSignOn": "enabled" }
```

2.12.2 /authentication/dsview

Get and set DSView software authentication parameters.

Methods

GET, PUT, PATCH

Parameters

PARAMETER	DESCRIPTION
serverIpAddress1	IP Address of DSView Server #1
serverIpAddress2	IP Address of DSView Server #2
serverIpAddress3	IP Address of DSView Server #3
serverIpAddress4	IP Address of DSView Server #4

Query

Fields are supported for all parameters.

Response Codes

200	OK
204	OK No Content) for a PUT or PATCH
400	Bad Request

Examples

```
GET /authentication/dsview
        {
            "serverIpAddress1": "10.20.30.40",
```

```
            "serverIpAddress2": "0.0.0.0",
            "serverIpAddress3": "0.0.0.0",
            "serverIpAddress4": "0.0.0.0"
        }
PUT /authentication/dsview { "serverIpAddress1": "10.20.30.40" }
```

2.12.3 /authentication/kerberos

Get and set Kerberos authentication parameters.

Methods

GET, PUT, PATCH

Parameters

PARAMETER	DESCRIPTION
server	IP address (or realm) of the Kerberos server
realmDomainName	
domainName	

Query

Fields are supported for all parameters.

Response Codes

200	OK
204	OK No Content) for a PUT or PATCH
400	Bad Request

Examples

```
GET /authentication/kerberos
        {
            "server": "10.20.30.41",
            "realmDomainName": "avocent.com",
            "domainName": "avocent.com"
        }
PUT /authentication/kerberos { "server": "10.20.30.41" }
```

2.12.4 /authentication/ldap

Get and set LDAP authentication parameters.

Methods

GET, PUT, PATCH

Parameters

PARAMETER	DESCRIPTION
server	IP Address of the LDAP server
base	Base
secure	Secure mode: **on/off/start_tls**
userName	Database username
password	Database password for the username
attributes	Login attributes

Query

Fields are supported for all parameters.

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

```
GET /authentication/ldap
        {
            "server": "10.20.30.42",
            "base": "",
            "secure": "off",
            "userName": "myuser",
            "password": "mypassword",
            "attributes": ""
        }
PUT /authentication/ldap { "server":"10.20.30.42", "secure": "on" }
```

2.12.5 /authentication/radius

Get and set RADIUS authentication parameters.

Methods

GET, PUT, PATCH

Parameters

PARAMETER	DESCRIPTION
firstAuthenticationServer	IP address of the first authentication server
firstAccountingServer	IP address of the first accounting server
secondAuthenticationServer	IP address of the second authentication server

PARAMETER		DESCRIPTION
secondAccountingServer		IP address of the second accounting server
secret		Secret word or passphrase, applies to both sets of servers
timeout		Desired number of seconds for server timeout: integer
retries		Desired number of retries: integer
serviceType		Enables Service-Type attribute to specify the authorization group: **enabled/disabled**
serviceTypeGroups: (only valid if serviceType is enabled)		
	login	Authorization group name for Login
	framed	Authorization group name for Framed
	callbackLogin	Authorization group name for Callback Login
	callbackFramed	Authorization group name for Callback Framed
	outbound	Authorization group name for Outbound
	administrative	Authorization group name for Administrative

Query

Fields are supported for all parameters.

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

GET /authentication/radius


```
{
    "firstAuthenticationServer": "127.0.0.1",
    "firstAccountingServer": "127.0.0.1",
    "secondAuthenticationServer": "",
    "secondAccountingServer": "",
    "secret": "******",
    "timeout": 3,
    "retries": 2,
    "serviceType": "enabled",
    "serviceTypeGroups": {
        "login": "",
        "framed": "",
        "callbackLogin": "",
        "callbackFramed": "",
        "outbound": "",
        "administrative": ""
    }
}
```


PUT /authentication/radius {"firstAuthenticationServer": "10.20.30.45","timeout":120}

2.12.6 /authentication/tacacs

Get and set TACACS+ authentication parameters.

Methods

GET, PUT, PATCH

Parameters

PARAMETER		DESCRIPTION
firstAuthenticationServer		IP Address of the first authentication server
firstAccountingServer		IP Address of the first accounting server
secondAuthenticationServer		IP Address of the second authentication server
secondAccountingServer		IP Address of the second accounting server
service		Service: **ppp/raccess/shell**
secret		Secret word or passphrase, applies to both sets of servers
timeout		Desired number of seconds for server timeout
retries		Desired number of retries
version		Version: **v0 / v0_v1 / v1 / v1_v0**
userLevel		Enable User-Level attribute to specify the authorization group: **enabled/disabled**
userLevelGroups: (only valid if userLevel is enabled)		
	userLevel1	Authorization group name for User-Level 1
	...	...
	userLevel15	Authorization group name for User-Level 15

Query

Fields are supported for all parameters.

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

GET /authentication/tacacs


```
{
    "firstAuthenticationServer": "10.20.30.46",
    "firstAccountingServer": "10.20.30.47",
    "secondAuthenticationServer": "",
    "secondAccountingServer": "",
    "service": "ppp",
    "secret": "",
    "timeout": 10,
```

```
            "retries": 2,
            "version": "v1",
            "userLevel": "enabled",
            "userLevelGroups": {
                "userLevel1": "",
                ...
                "userLevel15": ""
            }
        }
PUT /authentication/tacacs {"firstAuthenticationServer": "10.20.30.46",
"firstAccountingServer": "10.20.30.47","version": "v0_v1"}
```

2.13 Users

2.13.1 /users[/<NAME>]

This command provides the ability to view and edit user settings as well as add new users and delete existing users.

Methods

GET, PUT, PATCH, POST, DELETE

NOTE: For methods PUT, PATCH, POST and DELETE, the Name is required. For example: DELETE /users/testuser.

Parameters

PARAMETER		DESCRIPTION
name		User name; admin and root exist by default
settings		
	password	Password for the user
	changePasswordNextLogin	Set to force the user to change the password the next time they log in; **enabled/disabled**
	userGroups	List of groups to which this user belongs: string array
passwordExpiration		
	minimumDays	Minimum number of days allowed between password changes; an integer
	maximumDays	Maximum number of days a password is valid; an integer
	inactiveDays	Number of inactive days after which a password is considered expired; an integer
	warningDays	Number of days a warning is issued to the user prior to expiration; an integer
accountExpiration		Account expiration date

Query

Fields are supported for all parameters.

Response Codes

200	OK
201	Created
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

GET /users

```
{
    "users": [
        {
            "name": "admin",
            "settings": {
                "password": "",
                ...
        },
        {
            "name": "root",
            ...
        }
    ]
}
```

GET /users/admin

```
{
    "name": "admin",
    "settings": {
        "password": "",
        "changePasswordNextLogin": "disabled",
        "userGroups": [
            "admin"
        ],
        "passwordExpiration": {
            "minimumDays": 0,
            "maximumDays": 99999,
            "inactiveDays": "",
            "warningDays": 7
        },
        "accountExp": ""
    }
}
```

POST /users

```
{
    "name": "bob",
    "settings": {
        "password": "password1234!",
        "changePasswordNextLogin": "enabled",
        ...
}
```

Response is:

```
{
    "name": "bob",
    "settings": {
        "password": "password1234!",
```

```
        ...
    }
```

2.13.2 /users/<NAME>/unlock

This resource provides the ability to unlock a locked user account.

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
POST /users/admin/unlock
Response is:
        {
            "status": "user account unlocked"
        }
```

2.13.3 /users/passwordRules

Configure the syslog parameters.

Methods

GET, PUT, PATCH

Parameters

PARAMETER		DESCRIPTION
checkComplexity		enabled/disabled
complexitySettings:		
	minimumDigits	Integer minimum number of digits
	minimumUppercase	Integer minimum number of uppercase characters
	minimumSpecial	Integer minimum number of special characters
	minimumSize	Integer minimum number of total characters
expirationSettings:		
	minimumDays	Integer minimum number of days; default 0
	maximumDays	Integer maximum number of days; default 99999
	warningDays	Integer number of warning days; default 7
lockoutSettings:		
	failedAttempts	Integer number of failed attempts allowed before locking account; default is 0, which is disabled
	lockoutDuration	Integer account lockout duration after each failed login in minutes; default is 0.
	unlockAfter	Integer unlock account after a specific number of minutes; default is 10. 0 means manual unlock

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

GET /users/passwordRules

```
{
    "checkComplexity": "enabled",
    "complexitySettings": {
        "minimumDigits": 0,
        "minimumUppercase": 0,
        "minimumSpecial": 0,
        "minimumSize": 8
    },
    "expirationSettings": {
        "minimumDays": 0,
```

```
            "maximumDays": 99999,
            "warningDays": 7,
        },
        "lockoutSettings": {
            "failedAttempts": 0,
            "lockoutDuration": 0,
            "unlockAfter": 10
        }
    }
PUT /users/passwordRules
        {
            "checkComplexity": "disabled",
            "lockoutSettings": {
                "failedAttempts":3,
                "lockoutDuration":1,
                "unlockAfter": 5
            }
        }
```

2.14 Groups

2.14.1 /groups[/<NAME>]

This resource provides the ability to view and edit group settings as well as add new users and delete existing users.

Methods

GET, PUT, PATCH, POST, DELETE

Parameters

PARAMETER		DESCRIPTION
name		Group name. admin, appliance-admin, shell-login-profile and user exist by default.
members		List of users which belong to this group. String array.
settings		
	sessionTimeout	**global/custom**
	customTimeout	Custom session timeout in seconds. Integer. Default 300.
	profile	**none/ts_menu/cli**
	tsMenuOptions	String.
	cliCmd	String. Example: shell.
	cliExitAfterExecuting	**enabled/disabled**

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
201	Created
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

GET /groups

```
{
    "groups": [
        {
            "name": "admin",
            "settings" : {
                "sessionTimeout": "global",
                "customTimeout": 300,
                "profile": "cli",
                "tsMenuOptions": "",
                "cliCmd": "shell",
                "cliExitAfterExecuting": "disabled"
            },
            "members" : [
                "root",
                "admin"
            ]
        },
        {
            "name": "appliance-admin",
            "settings" : {
                ...
        }
    ]
}
```

GET /groups/admin

```
{
    "name": "admin",
    "members" : [
        "admin", "root"
    ],
    "settings" : {
        "sessionTimeout": "global",
        "profile": "cli",
        "cliCmd": "shell",
        "cliExitAfterExecuting":"disabled"
    }
}
```

POST /groups {"name":"mygroup"}

PUT /groups/mygroup

```
{
    "settings" : {
        "sessionTimeout": "custom",
        "customTimeout": 100,
        "profile": "none"
    },
```

```
            "members": ["admin","root"]
        }

PUT /groups/mygroup
        {
            "settings" : {
                "sessionTimeout": "custom",
                "customTimeout": 150,
                "profile": "ts_menu",
                "tsMenuOptions": "my options"
            }
        }

PUT /groups/mygroup
        {
            "settings" : {
                "sessionTimeout": "custom",
                "customTimeout": 250,
                "profile": "cli",
                "cliCmd": "shell",
                "cliExitAfterExecuting": "disabled"
            }
        }
```

2.15 AccessRights

2.15.1 /accessRights/users/<NAME>/serialPorts[/<PORT>]

Configure the individual users access rights to serial ports. This is an array of access right structures.

Methods

GET, PUT, PATCH, POST, DELETE

Parameters

PARAMETER	DESCRIPTION
port	Serial port number
name	Serial port name
session	single/multi
accessMode	read_write/read_only
killMultiSession	enabled/disabled
sendMessageMultiSession	enabled/disabled
powerControl	enabled/disabled
dataBufferManagement	enabled/disabled
restfulMenu	enabled/disabled

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
201	Created
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

```
GET /accessRights/users/myuser/serialPorts
        {
            "accessRights": [
                {
                    "port": 1,
                    "name": "ttyS1",
                    "session": "single",
                    "accessMode": "read_write",
                    "killMultiSession": "enabled",
                    "sendMessageMultiSession": "disabled",
                    "powerControl": "enabled",
                    "dataBufferManagement": "disabled",

                },
                {
                    "port": 2,
                    "name": "ttyS2",
                    "session": "multi",
                    "accessMode": "read_only",
                    ...
                }
            ]
        }
GET /accessRights/users/myuser/serialPorts/1
        {
            "port": 1,
            "name": "ttyS1",
            "session": "single",
            ...
        }
PUT /accessRights/users/myuser/serialPorts/2
        {
            "session": "read_only"
        }
POST /accessRights/users/myuser/serialPorts
        {
            "port": 3,
            "name": "ttyS3",
            "session": "single"
        }
```

Response is:

```
{
        "port": 3,
        "name": "ttyS3",
        "session": "single",
        "accessMode": "read_write",
        "killMultiSession": "disabled ",
        "sendMessageMultiSession": "disabled",
        "powerControl": "disabled ",
        "dataBufferManagement": "disabled",
        "restfulMenu": "disabled",
}
```


DELETE /accessRights/users/myuser/serialPorts/2

2.15.2 /accessRights/users/<NAME>/appliance

Configure the individual users access rights to the appliance.

Methods

GET, PUT, PATCH

Parameters

PARAMETER	DESCRIPTION
view	enabled/disabled
disconnect	enabled/disabled
reboot	enabled/disabled
flashUpgrade	enabled/disabled
configureSettings	enabled/disabled
configureUsers	enabled/disabled
backupRestoreConfig	enabled/disabled
shellAccess	enabled/disabled
transferFiles	enabled/disabled

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

```
GET /accessRights/users/myuser/appliance
        {
            "view": "disabled",
            "disconnect": "disabled",
            "reboot": "disabled",
            "flashUpgrade": "disabled",
            "configureSettings": "disabled",
            "configureUsers": "disabled",
            "backupRestoreConfig": "disabled",
            "shellAccess": "disabled",
            "transferFiles": "disabled"
        }
PUT /accessRights/users/myuser/appliance
        {
            "shellAccess": "enabled"
        }
```

2.15.3 /accessRights/users/<NAME>/pdus

Configure the individual users access rights to PDU devices.

Methods

GET, POST, DELETE

Parameters

PARAMETER	DESCRIPTION
pduId	PDU name

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
201	Created
400	Bad Request

Examples

GET /accessRights/users/myuser/pdus
Response is:

```
{
    "accessRightsPdu": [
        {
            "pduId": "MyPDU1"
        },
        {
            "pduId": "MyPDU2"
        }
    ]
}
```

POST /accessRights/users/myuser/pdus

```
{
    "pduId": "newPdu"
}
```

DELETE /accessRights/users/myuser/pdus/newPdu

2.15.4 /accessRights/users/<NAME>/outlets

Configure the individual users access rights to individual PDU outlets

Methods

GET, POST, DELETE

Parameters

PARAMETER	DESCRIPTION
pduId	PDU name
outlets	List of outlets

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
201	Created
400	Bad Request

Examples

GET /accessRights/users/myuser/outlets
Response is:

```
{
    "accessRightsOutlets": [
        {
            "pduId": "MyPDU",
            "outlets": "13-15"
        },
        {
            "pduId": "MyPDU2",
            "outlets": "1-3,5"
        }
    ]
}
```

POST /accessRights/users/myuser/outlets

```
{
    "pduId": "newPdu",
    "outlets" : "7,8,10-12"
}
```

DELETE /accessRights/users/myuser/outlets/newPdu

2.15.5 /accessRights/users/<NAME>/ups

Configure the individual users access rights to UPS devices.

Methods

GET, POST, DELETE

Parameters

PARAMETER	DESCRIPTION
upsId	UPS name

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
201	Created
400	Bad Request

Examples

GET /accessRights/users/myuser/ups
Response is:

```
{
    "accessRightsUps": [
        {
            "upsId": "GXT4_1"
        },
        {
            "upsId": "GXT4_2"
        }
    ]
}
```

POST /accessRights/users/myuser/ups

```
{
    "upsId": "newUps"
}
```

DELETE /accessRights/users/myuser/ups/newUps

2.15.6 /accessRights/users/<NAME>/upsOutletGroups

Configure the individual users access rights to individual UPS outlet groups.

Methods

GET, POST, DELETE

Parameters

PARAMETER	DESCRIPTION
upsId	UPS name
outletGroups	List of outlet groups

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
201	Created
400	Bad Request

Examples

GET /accessRights/users/myuser/upsOutletGroups
Response is:

```
{
    "accessRightsOutlets": [
        {
            "upsId": "GXT4_1",
            "outletGroups": "1-2"
        },
        {
            "upsId": "GXT4_2",
            "outletGroups": "2"
        }
    ]
}
```

POST /accessRights/users/myuser/upsOutletGroups

```
{
    "upsId": "newUps",
    "outletGroups" : "1-2"
}
```

DELETE /accessRights/users/myuser/upsOutletGroups/newUps

2.15.7 /accessRights/groups/<NAME>/serialPorts[/<PORT>]

Refer to /accessRights/users/<NAME>/serial[/<PORT>].

2.15.8 /accessRights/groups/<NAME>/appliance

Refer to /accessRights/users/<NAME>/appliance.

2.15.9 /accessRights/groups/<NAME>/pdus

Refer to /accessRights/users/<NAME>/pdus.

2.15.10 /accessRights/groups/<NAME>/outlets

Refer to /accessRights/users/<NAME>/outlets.

2.15.11 /accessRights/groups/<NAME>/ups

Refer to /accessRights/users/<NAME>/ups.

2.15.12 /accessRights/groups/<NAME>/upsOutletGroups

Refer to /accessRights/users/<NAME>/upsOutletGroups.

2.15.13 /accessRights/dsview

This resource provides the ability to configure the dsview access rights.

Methods

GET, PUT, PATCH

Parameters

PARAMETER	DESCRIPTION
session	single/multi
accessMode	read_write/read_only
killMultiSession	enabled/disabled
sendMessageMultiSession	enabled/disabled

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

```
GET /accessRights/dsview
        {
            "session": "single",
            "accessMode": "read_write",
            "killMultiSession": "enabled",
            "sendMessageMultiSession": "disabled",
        }
PUT /accessRights/dsview
            {
                  "accessMode": "read_only"
            }
```

2.16 Events and Logs

2.16.1 /events[/<ID>]

List, enable and disable the various events and control how their notifications are sent.

Methods

GET, PUT, PATCH

Parameters

PARAMETER	DESCRIPTION
event	Event number. Integer. Read-only.
description	Event description. Read-only.
traps	Send a trap to one or more SNMP servers. **enabled/disabled**
syslog	Send a message to one or more Syslog servers. **enabled/disabled**
dsview	Send a message to one or more DSView servers. **enabled/disabled**
email	Send an email. **enabled/disabled**
sms	Send a text message. **enabled/disabled**

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

```
GET /events
        {
            "events":  [
                {
                    "event": 1,
                    "description": "Appliance Rebooting",
                    "traps": "disabled",
                    "syslog": "enabled",
                    "dsview": "disabled",
                    "email": "disabled",
                    "sms": "disabled"
                },
                {
                    "event": 2,
                    "description": "Appliance Daemon Started",
                    "traps": "disabled",
                    "syslog": "disabled",
                    "dsview": "disabled",
                    "email": "disabled",
```

```
                "sms": "disabled"
            },
            ...
        ]
    }
```

```
GET /events/513
    {
        "events":  513,
        "description": "CPU Temperature Max Threshold Exceeded",
        "traps": "disabled",
        "syslog": "enabled",
        "dsview": "disabled",
        "email": "enabled",
        "sms": "disabled"
    },
```

```
GET /events?syslog=enabled

PUT /events/513
    {
        "traps": "enabled",
        "syslog": "disabled"
    }
```

2.16.2 /events/syslog

Configure the syslog parameters.

Methods

GET, PUT, PATCH

Parameters

PARAMETER	DESCRIPTION
facility	Log Local facility. **log_local_0/log_local_1/log_local_2/log_local_3/log_local_4/log_local_5**
remoteServerIpv4	**enabled/disabled**
ipv4Server1	Remote server IP address.
ipv4Port1	Remote server UDP port number. Integer.
ipv4Server2	Remote server IP address.
ipv4Port2	Remote server UDP port number. Integer.
ipv4Server3	Remote server IP address.
ipv4Port3	Remote server UDP port number. Integer.
ipv4Server4	Remote server IP address.

PARAMETER	DESCRIPTION
ipv4Port4	Remote server UDP port number. Integer.
ipv4Server5	Remote server IP address.
ipv4Port5	Remote server UDP port number. Integer.
remoteServerIpv6	**enabled/disabled**
ipv6Server1	Remote server IP address.
ipv6Port1	Remote server UDP port number. Integer.
ipv6Server2	Remote server IP address.
ipv6Port2	Remote server UDP port number. Integer.
ipv6Server3	Remote server IP address.
ipv6Port3	Remote server UDP port number. Integer.
ipv6Server4	Remote server IP address.
ipv6Port4	Remote server UDP port number. Integer.
ipv6Server5	Remote server IP address.
ipv6Port5	Remote server UDP port number. Integer.
applianceConsole	Send syslog messages to the appliance console. **enabled/disabled**
rootSession	Send syslog messages to any root sessions. **enabled/disabled**

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

GET /events/syslog


```
{
    "facility": 2,
    "remoteServerIpv4": "enabled",
    "ipv4Server1": "10.20.30.40",
    "ipv4Port1": 514,
    "ipv4Server2": "",
    "ipv4Server3": "",
    "ipv4Server4": "",
    "ipv4Server5": "",
    "remoteServerIpv6": "disabled",
```

```
        "applianceConsole": "enabled",
        "rootSession": "disabled"
    }
```

```
PUT /events/syslog
        {
            "remoteServerIpv6": "enabled",
            "ipv6Server1": "10.20.30.41",
            "ipv6Port1": 514
        }
```

2.16.3 /events/snmp

Configure the event destination SNMP parameters.

Methods

GET, PUT, PATCH

Parameters

PARAMETER	DESCRIPTION
community	string
server1	SNMP server1 IP address
server2	SNMP server2 IP address
server3	SNMP server3 IP address
server4	SNMP server4 IP address
server5	SNMP server5 IP address

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

```
GET /events/snmp
        {
            "community": 2,
            "server1": 10.20.30.40",
            "server2": "",
            "server3": "",
            "server4": "",
            "server5": "",
        }
```

```
PUT /events/snmp
            {
                "server2": 10.20.30.42",
            }
```

2.16.4 /events/sms

Configure the sms (text message) event destination parameters.

Methods

GET, PUT, PATCH

Parameters

PARAMETER	DESCRIPTION
server	IP address of an SMS server
port	TCP port number on server
number	Number to send messages to

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

GET /events/sms

```
{
    "server": 10.20.30.40",
    "port": "",
    "number": "8001234567",
}
```

PUT /events/sms

```
{
    "number": "8001234567",
}
```

2.16.5 /events/email

Configure the email event destination parameters.

Methods

GET, PUT, PATCH

Parameters

PARAMETER	DESCRIPTION
server	IP address of an email server
port	TCP port number to use
number	Email address to recieve event messages to send messages to

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

```
GET /events/mail
        {
            "server": 10.20.30.40",
            "port": "25",
            "destination": "help@myemail.com"
        }
```

```
PUT /events/email
        {
            "destination": "help@myemail.com"
        }
```

2.16.6 /events/dsview

Configure the DSView event destination parameters.

Methods

GET, PUT, PATCH

Parameters

PARAMETER	DESCRIPTION
server	IP address of the DSView server.
syslogServerPort	Syslog server port. Default 4514.
sshServerPort	SSH server port. Default 4122.
sshUsername	SSH username.
sshIdleTimeout	SSH idle timeout in seconds. Default 15.
sshStartThreshold	SSH start threshold in bytes. Default 10000.
sshTunnelBufferSize bufferFullFirstWarning	SSH tunnel buffer size in bytes. Default 1000000.
bufferFullFirstWarning	Buffer full first warning in bytes. Default 500000.
bufferFullSecondWarning	Buffer full second warning in bytes. Default 700000.
bufferFullThirdWarning	Buffer full third warning in bytes. Default 1000000.

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

```
GET /events/dsview
        {
            "server": 10.20.30.40",
            "syslogServerPort": "4514,
            "sshServerPort": 4122,
            "sshUsername": "admin",
            "sshIdleTimeout": 15,
            "sshStartThreshold": 10000,
            "sshTunnelBufferSize": 1000000,
            "bufferFullFirstWarning": 500000,
            "bufferFullSecondWarning": 700000,
            "bufferFullThirdWarning": 1000000
        }
```

```
PUT /events/dsview
            {
                "server": "10.20.30.40",
                "sshUsername": "admin",
            }
```

2.16.7 /events/trapForward[/<ID>]

Add and delete table entries consisting of:

- Index (read-only)
- serverIPAddress
- serverUDPPort
- OID for filter

Methods

GET,PUT,PATCH,POST,DELETE

Parameters

PARAMETER	DESCRIPTION
index	Read-only integer id.
serverIPAddress	IP address of the server to send traps to.
serverUDPPort	UDP port number to send traps to. Integer.
oid	OID for filter. Default: "default".

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
201	Created
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

GET /events/trapForward

```
{
    "trapForward": [
        {
            "index": 0,
            "serverIpAddress": "10.20.30.40",
            "serverUdpPort": 100,
            "oid": "default"
        },
        {
            "index": 1,
            "serverIpAddress": "10.20.30.41",
            "serverUdpPort": 101,
            "oid": ".1.1.1"
        },
        ...
    ]
}
```

GET /events/trapForward/1

```
{
    "index": 1,
    "serverIPAddress": "10.20.30.41",
    "serverUDPPort": 101,
```

```
        "oid": ".1.1.1"
    }
```

```
POST /events/trapForward
        {
            "serverIPAddress": "10.20.30.43",
            "oid": "default"
        }
```

Response is:

```
    {
        "index": 3,
        "serverIPAddress": "10.20.30.43",
        "serverUDPPort": 162,
        "oid": "default"
    }
```

```
DELETE /events/trapForward/1
```

2.16.8 /events/dataBuffering

Configure data buffer settings.

Methods

GET, PUT, PATCH

Parameters

PARAMETER	DESCRIPTION
segmentSize	Local file segment size in Kbytes. 0=unlimited. Default 256.
spareSegments	Number of local spare file segments. Default 1.
nfsServer	IP Address of the NFS server to write data.
nfsPath	File path on NFS server to write data.
nfsSegmentSize	NFS file segment size in Kbytes. 0=unlimited. Default 1024.
nfsSpareSegments	Number of nfs spare file segments. Default 10.
closeByTime	Time in HH:MM format when the current log file should be closed and a new one opened.
syslogFacility	Syslog Log Local facility to use for data buffering. **log_local_0/log_local_1/log_local_2/log_local_3/log_local_4/log_local_5**

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

```
GET /events/dataBuffering
        {
            "segmentSize": 256,
            "spareSegments": 1,
            "nfsServer": "10.20.30.40",
            "nfsPath": "/mylogging",
            "nfsSegmentSize": 1024,
            "nfsSpareSegments": 10,
            "closeByTime": "01:00",
            "syslogFacility": "log_local_3"
        }
```

```
PUT /events/dataBuffering
            {
                "segmentSize": 64,
                "spareSegments": 5,
                "closeByTime": "23:30",
                "syslogFacility": "log_local_4"
            }
```

2.16.9 /events/applianceLogging

Configure appliance logging settings.

Methods

GET, PUT, PATCH

Parameters

PARAMETER	DESCRIPTION
sessionLogging	Appliance session logging. enabled/disabled
destination	Destination for writing log file. local/nfs/syslog/dsview
localDestination	Local filesystem for logging. Default mmcblk0.
timestamp	Include timestamp in log entries. enabled/disabled
alerts	Appliance session data logging alerts. enabled/disabled

PARAMETER	DESCRIPTION
alertString1	
alertString2	
alertString3	
alertString4	
alertString5	
alertString6	
alertString7	
alertString8	
alertString9	
alertString10	

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

```
GET /events/applianceLogging
        {
            "sessionLogging": "enabled",
            "destination": "local",
            "localDestination": "mmcblk0",
            "timestamp": "enabled",
            "alerts": "enabled",
            "alertString1": "login",
            "alertString2": "logout",
            ...
            "alertString10": "error",
        }
```

```
PUT /events/applianceLogging
        {
            "sessionLogging": "enabled",
            "destination": "nfs"
        }
```

2.17 Power Management

2.17.1 /power/pdus[/<NAME>]

This resource provides the ability to view information about all attached PDUs and to configure PDU settings.

Methods

GET, PATCH, PUT

Parameters

PARAMETER		DESCRIPTION
pduID		Name assigned to this specific PDU
vendor		Read only name of the vendor of the PDU
model		Read only model number of the PDU
device		Read only ttyS# form device name, or an IP address for a networked PDU
Number		Read only Position of the PDU in the chain
firmwareVersion		Read only firmware version of the PDU
maxCurrent		Read only current rating of the PDU in Amps
numberOfBanks		Read only number of banks (circuits) in the PDU
numberOfPhases		Read only number of phases in the PDU
pduIpAddress		Read only IP Address of PDU, if available
numberOfOutlets		Read only number of outlets
numberOfOutletsOn		Read only number of outlets which are powered on
current		
	currentValue	Read only current draw in Amps
	currentMax	Read only maximum observed current in amps
	currentMin	Read only minimum observed current in amps
	currentAverage	Read only average observed current in amps
	alarm	Read only current alarm state of the PDU: **normal/low_critical/low_warning/high_critical/high_warning/not_available**
	highCritical	High critical current threshold in amps
	highWarning	High warning current threshold in amps
	lowWarning	Low warning current threshold in amps
	lowCritical	Low critical current threshold in amps
voltage		
	voltageValue	Read only voltage in volts
	voltageMax	Read only maximum observed voltage in volts
	voltageMin	Read only minimum observed voltage in volts
	voltageAverage	Read only average observed voltage in volts
	voltageType	Read only type: **measured, estimated**
powerConsumption		

PARAMETER		DESCRIPTION
	powerValue	Read only power draw in watts
	powerMax	Read only maximum observed power draw in watts
	powerMin	Read only minimum observed power draw in watts
	powerAvg	Read only average observed power draw in watts
	powerType	Read only type, measured, estimated
powerFactor		Read only decimal ratio of the real power to the apparent power
energyConsumption		
	energyValue	Read only accumulated energy in kWh
	startTime	Read only timestamp for beginning of energy accumulation

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

GET /power/pdus/MPX
Response is:


```
{
    "pduId": "MPX",
    "vendor": "Vertiv",
    "model": "MPXPEM-NVAXXAXX",
    "device": "ttyS5",
    "number": "1",
    "firmwareVersion": "14.0.0.2",
    "maxCurrent": 30,
    "numberOfBanks": 3,
    "numberOfPhases": 0,
    "pduIpAddress": "10.207.24.75",
    "numberOfOutlets": 18,
    "numberOfOutletsOn": 18,
    "current": {
        "currentValue": "0.0",
        "currentMax": "0.0",
        "currentMin": "0.0",
        "currentAvg": "0.0",
        "alarm": "low_critical",
        "thresholds": {
```

```
            "highCritical": "27.0",
            "highWarning": "15.0",
            "lowWarning": "",
            "lowCritical": "1.8"
            }
        },
        "voltage": {
        "voltageValue": "115",
        "voltageMax": "116",
        "voltageMin": "114",
        "voltageAvg": "114",
        "voltageType": "measured"
        },
        "powerConsumption": {
            "powerValue": "0.0",
            "powerMax": "0.0",
            "powerMin": "0.0",
            "powerAvg": "0.0",
            "powerType": "measured"
        },
        "powerFactor": "0.00",
        "energyConsumption": {
            "energyValue": "0.700",
            "startTime": ""
        }
}
PUT /power/pdus/MPX
{
     "current": { "thresholds" : { "highCritical": "20.0",
                                   "highWarning": "15",
                                    "lowCritical": "0",
                                    "lowWarning": "0" } }
}
```


2.17.2 /power/pdus/<NAME>/off

Turn all outlets on the PDU off.

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

POST /power/pdus/MPX/off

Response is:

```
{
    "status": "success. all outlets turned off."
}
```

2.17.3 /power/pdus/<NAME>/on

Turn all outlets on the PDU on.

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

POST /power/pdus/MPX/on

Response is:

```
{
    "status": "success. all outlets turned on."
}
```

2.17.4 /power/pdus/<NAME>/cycle

Power cycle all outlets on a PDU.

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

POST /power/pdus/MPX/cycle

Response is:

```
{
    "status": "success. all outlets cycled."
}
```

2.17.5 /power/pdus/<NAME>/rename

Rename a PDU.

Methods

POST

Parameters

PARAMETER	DESCRIPTION
pduId	New name for the PDU

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
POST /power/pdus/MPX/rename
        {
            "pduId": "newname"
        }
Response is:
        {
            "status": "success. PDU renamed newname."
        }
```

2.17.6 /power/pdus/<NAME>/resetValues

Reset the min/max/avg electrical monitoring values for a PDU.

Methods

POST

Parameters

PARAMETER	DESCRIPTION
valuesType	Set to one of these values: **current, voltage, power, energy**

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
POST /power/pdus/MPX/resetValues
            {
                "valuesType":"current"
            }
Response is:
```

```
        {
            "status": "success. reset current for PDU."
        }
POST /power/pdus/MPX/resetValues
            {
                "valuesType":"voltage"
            }
Response is:
        {
            "status": "success. reset voltage for PDU."
        }
```

2.17.7 /power/pdus/<NAME>/reboot

Reboot the PDU controller. This does not affect outlet status.

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
POST /power/pdus/MPX/reboot

Response is:
        {
            "status": "success. reboot initiated."
        }
```

2.17.8 /power/pdus/<NAME>/factoryDefaults

Restore factory defaults on a PDU

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

POST /power/pdus/MPX/factoryDefaults

Response is:

```
{
    "status": "success. Factory defaults initiated"
}
```

2.17.9 /power/pdus/<NAME>/firmwareDownload

This action causes the appliance to download the firmware file specified, using ftp, in preparation for the subsequent firmware update of the PDU. The action does not return until the file download has completed or fails. Depending on the network speed, this could take a couple of minutes.

NOTE: Not all PDUs support firmware download.

Methods

POST

Parameters

PARAMETER	DESCRIPTION
ipAddress	IP address of the remote server from which to download the file
username	Username to access the remote server
password	Password to access the remote server
directory	Directory path on the remote server, typically relative to the ftp root directory
filename	Filename of the firmware file on the remote server

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
POST /power/pdus/MPX/firmwareDownload
        {
            "ipAddress": "10.20.30.80",
            "username": "anonymous",
            "password": "anonymous",
            "directory": "pub/firmware/",
            "filename": "pdu-firmware.fl"
        }
Response is:
    {
        "status": "download successful",
        "firmware": {
            "version": "14.0.0.2"
        }
    }
```

2.17.10 /power/pdus/<NAME>/firmwareInstall

This action causes the appliance to install a previously downloaded firmware image into the PDU.

This request does not return until the file has been transferred to the PDU and the upgrade has been initiated, which may take up to two minutes.

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

POST /power/pdus/MPX/firmwareInstall
Response is:

```
{
    "status": "install successful",
    "firmware": {
        "version": "14.0.0.2",
    }
}
```

2.17.11 /power/pdus/<NAME>/outlets[/<ID>]

This resource provides the ability to view information about all the outlets on the PDU, to control the outlets and configure the outlet settings.

Methods

GET, PUT, PATCH

Parameters

All parameters are read-only unless they are marked read-write.

PARAMETER		DESCRIPTION
outlet		Integer index.
name		Outlet name.
status		Outlet status: **on/off/on_fixed/on_locked/on_unlocked/on_pend_off/off_pend_on/on_pend_cycle/off_cycle/not_set**
bank		Bank number, if applicable. (Not present on all pdus)
current		
	currentValue	Current draw in Amps
	currentMax	Maximum observed current in amps.

PARAMETER		DESCRIPTION
	currentMin	Minimum observed current in amps.
	currentAverage	Average observed current in amps.
	Alarm	Current alarm state of the PDU: **normal/low_critical/low_warning/high_critical/high_warning/blown_fuse/not_available**
	highCritical	Read-write high critical current threshold in amps
	highWarning	Read-write high warning current threshold in amps
	lowWarning	Read-write low warning current threshold in amps
	lowCritical	Read-write low critical current threshold in amps
voltage		
	voltageValue	Voltage in volts
	voltageMax	Maximum observed voltage in volts
	voltageMin	Minimum observed voltage in volts
	voltageAverage	Average observed voltage in volts
	voltageType	measured or estimated
powerConsumption		
	powerValue	Power draw in watts
	powerMax	Maximum observed power draw in watts
	powerMin	Minimum observed power draw in watts
	powerAvg	Average observed power draw in watts
	powerType	measured or estimated
powerFactor		Decimal ratio of the real power to the apparent power
energyConsumption		
	energyValue	Accumulated energy in kWh
	startTime	Timestamp for beginning of energy accumulation
wakeUpState		Read-write wake state: **last, on, off**
postOnDelay		Read-write value in seconds
postOffDelay		Read-write value in seconds

Examples

GET /power/pdus/MPX/outlets/3

Response is:


```
{
    "number": "3",
    "name": "RCP_A-3_[417271G713C2013AUG01040",
    "bank": "A",
    "status": "on",
    "current": {
        "currentValue": "0.0",
        "currentMax": "0.0",
        "currentMin": "0.0",
        "currentAvg": "0.0",
```

```
        "alarm": "normal",
        "thresholds": {
            "highCritical": "19.0",
            "highWarning": "18.0",
            "lowWarning": "",
            "lowCritical": "0.0"
        }
    },
    "voltage": {
        "voltageValue": "116",
        "voltageMax": "117",
        "voltageMin": "114",
        "voltageAvg": "114",
        "voltageType": "measured"
    },
    "powerConsumption": {
        "powerValue": "0.0",
        "powerMax": "0.0",
        "powerMin": "0.0",
        "powerAvg": "0.0",
        "powerType": "measured"
    },
    "powerFactor": "0.00",
    "energyConsumption": {
        "energyValue": "0.000",
        "startTime": ""
    },
            "wakeUpState": "",
            "postOnDelay": "0.0",
            "postOffDelay": "0.0"
        }
PUT /power/pdus/MPX/outlets/3
            {
                "name": "newname"
                "postOnDelay": "5"
                "postOffDelay": "5"
                "current": { "thresholds" : { "highCritical": "20.0",
                                              "highWarning": "15",
                                              "lowCritical": "0",
                                              "lowWarning": "0" } }
            }
```

2.17.12 /power/pdus/<NAME>/outlets/<ID>/on

Turn a PDU outlet on.

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
POST /power/pdus/MPX/outlets/5/on
        {
            "status": "success. Outlet 5 turned on"
        }
```

2.17.13 /power/pdus/<NAME>/outlets/<ID>/off

Turn a PDU outlet off.

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
POST /power/pdus/MPX/outlets/5/off
        {
            "status": "success. Outlet 5 turned off."
        }
```

2.17.14 /power/pdus/<NAME>/outlets/<ID>/cycle

Power cycle an outlet. When the operation is completed the outlet will be in the “on” state.

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
POST /power/pdus/MPX/outlets/5/cycle
        {
            "status": "success. Outlet 5 cycled."
        }
```

2.17.15 /power/pdus/<NAME>/outlets/<ID>/lock

Lock an outlet into its current power state (if supported by the PDU).

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
POST /power/pdus/MPX/outlets/5/lock
        {
            "status": "success. outlet 5 locked."
        }
```

2.17.16 /power/pdus/<NAME>/outlets/<ID>/unlock

Unlock an outlet (if supported by the PDU).

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
POST /power/pdus/MPX/outlets/5/unlock
        {
            "status": "success. outlet 5 unlocked."
        }
```

2.17.17 /power/pdus/<NAME>/outlets/<ID>/resetValues

Reset the min/max/avg electrical monitoring values for an outlet.

Methods

POST

Parameters

PARAMETER	DESCRIPTION
valuesType	Set to one of these values: **current, voltage, power, energy**

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
POST /power/pdus/MPX/outlets/1/resetValues
        {
                "valuesType":"current"
            }
Response is:
        {
            "status": "success. reset current for outlet."
        }
```

2.17.18 /power/pdus/<NAME>/banks[/<ID>]

This resource provides the ability to view information about all the banks on the PDU.

Methods

GET, PUT, PATCH

Parameters

All parameters are read-only unless they are marked as read-write.

PARAMETER		DESCRIPTION
Number		Number of the bank
Ref		Reference name for the bank
names		Read only name for the bank
bank		Bank number, if applicable
Maxcurrent		Current rating for the bank

PARAMETER		DESCRIPTION
current		
	currentValue	Current draw in Amps
	currentMax	Maximum observed current in amps
	currentMin	Minimum observed current in amps
	currentAverage	Average observed current in amps
	Alarm	Current alarm state of the PDU: **normal/low_critical/low_warning/high_critical/high_warning/blown_fuse/not_available**
	highCritical	Read-write high critical current threshold in amps
	highWarning	Read-write high warning current threshold in amps
	lowWarning	Read-write low warning current threshold in amps
	lowCritical	Read-write low critical current threshold in amps
voltage		
	voltageValue	Voltage in volts
	voltageMax	Maximum observed voltage in volts
	voltageMin	Minimum observed voltage in volts
	voltageAverage	Average observed voltage in volts
	voltageType	measured or estimated
powerConsumption		
	powerValue	Power draw in watts
	powerMax	Maximum observed power draw in watts
	powerMin	Minimum observed power draw in watts
	powerAvg	Average observed power draw in watts
	powerType	measured or estimated
powerFactor		Decimal ratio of the real power to the apparent power
energyConsumption		
	energyValue	Accumulated energy in kWh
	startTime	Timestamp for beginning of energy accumulation

Examples

GET /power/pdus/MPX/banks/1


```
{
    "number": "1",
    "name": "BR_A_[417271G713C2013AUG010403]",
    "ref": "A",
     "maxCurrent": "20.0",
    "current": {
        "currentValue": "0.0",
        "currentMax": "0.0",
        "currentMin": "0.0",
        "currentAvg": "0.0",
        "alarm": "low_critical",
```

```
            "thresholds": {
                "highCritical": "19.0",
                "highWarning": "18.0",
                "lowWarning": "",
                "lowCritical": "2.0"
            }
        },
        "voltage": {
            "voltageValue": "116",
            "voltageMax": "117",
            "voltageMin": "114",
            "voltageAvg": "114",
            "voltageType": "measured"
        },
        "powerConsumption": {
            "powerValue": "0.0",
            "powerMax": "0.0",
            "powerMin": "0.0",
            "powerAvg": "0.0",
            "powerType": "measured"
        },
        "powerFactor": "0.00",
        "energyConsumption": {
            "energyValue": "0.000",
            "startTime": ""
          }
        }
PUT /power/pdus/MPX/banks/3
            {
                  "current": { "thresholds" : { "highCritical": "20.0",
                                                "highWarning": "15",
                                                "lowCritical": "0",
                                                "lowWarning": "0" } }
            }
```

2.17.19 /power/pdus/<NAME>/banks/<ID>/resetValues

Reset the min/max/avg electrical monitoring values for a bank.

Methods

POST

Parameters

PARAMETER	DESCRIPTION
valuesType	Set to one of these values: **current, voltage, power, energy**

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
POST /power/pdus/MPX/banks/1/resetValues
        {
            "valuesType":"current"
        }
Response is:
    {
        "status": "success. reset current for bank."
    }
```

2.17.20 /power/pdus/<NAME>/phases[/<ID>]

This resource provides the ability to view information about all the phases on the PDU.

Methods

GET, PUT, PATCH

Parameters

All parameters are read-only unless they are marked as read-write.

PARAMETER		DESCRIPTION
Number		Number of the phase
Ref		Reference name for the phase
names		Read only name for the phase
bank		Phase number, if applicable
Maxcurrent		Current rating for the phase
current		
	currentValue	Current draw in Amps
	currentMax	Maximum observed current in amps
	currentMin	Minimum observed current in amps
	currentAverage	Average observed current in amps
	alarm	Current alarm state of the PDU: **normal/low_critical/low_warning/high_critical/high_warning/blown_fuse/not_available**
	highCritical	Read-write high critical current threshold in amps
	highWarning	Read-write high warning current threshold in amps
	lowWarning	Read-write low warning current threshold in amps
	lowCritical	Read-write low critical current threshold in amps
voltage		

PARAMETER		DESCRIPTION
	voltageValue	Voltage in volts
	voltageMax	Maximum observed voltage in volts
	voltageMin	Minimum observed voltage in volts
	voltageAverage	Average observed voltage in volts
	voltageType	measured or estimated
powerConsumption		
	powerValue	Power draw in watts
	powerMax	Maximum observed power draw in watts
	powerMin	Minimum observed power draw in watts
	powerAvg	Average observed power draw in watts
	powerType	measured or estimated
powerFactor		Decimal ratio of the real power to the apparent power
energyConsumption		
	energyValue	Accumulated energy in kWh
	startTime	Timestamp for beginning of energy accumulation

Examples

GET /power/pdus/MPH2/phases

Response is:


```
{
    "number": "1",
    "name": "MPH2",
    "ref": "L1",
    "maxCurrent": "48.0",
    "current": {
        "currentValue": "0.0",
        "currentMax": "0.0",
        "currentMin": "0.0",
        "currentAvg": "0.0",
        "alarm": "normal",
        "thresholds": {
            "highCritical": "43.2",
            "highWarning": "19.2",
            "lowWarning": "",
            "lowCritical": "0.0"
        }
    },
    "voltage": {
        "voltageValue": "208",
        "voltageMax": "208",
        "voltageMin": "207",
        "voltageAvg": "207",
        "voltageType": "measured"
    },
    "powerConsumption": {
        "powerValue": "0.0",
        "powerMax": "0.0",
```

```
            "powerMin": "0.0",
            "powerAvg": "0.0",
            "powerType": "measured"
        },
        "powerFactor": "0.00",
            "energyConsumption": {
            "energyValue": "",
            "startTime": ""
        }
}
PUT /power/pdus/MPH2/phases/3
        {
            "current": { "thresholds" : { "highCritical": "20.0",
                                          "highWarning": "15",
                                          "lowCritical": "0",
                                          "lowWarning": "0" } }
        }
```

2.17.21 /power/pdus/<NAME>/phases/<ID>/resetValues

Reset the min/max/avg electrical monitoring values for a phase.

Methods

POST

Parameters

PARAMETER	DESCRIPTION
valuesType	Set to one of these values: **current, voltage, power, energy**

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
POST /power/pdus/MPH2/phases/1/resetValues
        {
            "valuesType":"current"
        }
Response is:
```

```
{
    "status": "success. reset current for phase."
}
```

2.17.22 /power/pdus/<NAME>/sensors[/<ID>]

This resource provides the ability to view information about all the sensors on the PDU.

Methods

GET, PUT, PATCH

Parameters

All values are read-only unless they are marked as read-write.

PARAMETER	DESCRIPTION
number	Number of the sensor
ref	Reference name for the sensor
name	Read only name for the sensor
unit	Unit value; C,F for temperature sensors, % for humidity sensor
sensorValue	Current draw in Amps
sensorMax	Maximum observed current in amps
sensorMin	Minimum observed current in amps
sensorAverage	Average observed current in amps
highCritical	Read-write high critical current threshold in amps
highWarning	Read-write high warning current threshold in amps
lowWarning	Read-write low warning current threshold in amps
lowCritical	Read-write low critical current threshold in amps

Examples

GET /power/pdus/MPX/sensors

```
{
    "number": "1",
    "name": "T-SNSR-P5.1.1",
    "ref": "T1.1",
    "type": "external-temperature",
    "unit": "C",
    "sensorValue": "24.5",
    "sensorMax": "25.0",
    "sensorMin": "24.5",
    "sensorAvg": "24.5",
    "thresholds": {
        "highCritical": "37.0",
        "highWarning": "35.0",
        "lowWarning": "",
```

```
            "lowCritical": ""
        }
    }

PUT /power/pdus/MPX/sensors/1
{
  "thresholds" : { "highCritical": "37.0",
                                    "highWarning": "35",
                                    "lowCritical": "0",
                                    "lowWarning": "0" } }
}
```

2.17.23 /power/pdus/<NAME>/sensors/<ID>/resetValues

Reset the min/max/avg values for a sensor.

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
POST /power/pdus/MPX/sensors/1/resetValues
Response is:
        {
            "status": "success. reset values for sensor. "
        }
```

2.17.24 /power/ups[/<NAME>]

This resource provides the ability to view information about all attached UPS devices, configure UPS settings and control UPS output and outlet groups.

Methods

GET, PATCH, PUT

Parameters

All parameters are read-only unless they are marked as read-write.

PARAMETER		DESCRIPTION
upsID		Name assigned to this specific UPS.
vendor		Name of the vendor of the UPS.
model		Model number of the UPS.
device		ttyS# form device name, or an IP address for a networked UPS.
input		
	inputRmsL1N	System input RMS voltage between line 1 and neutral in VAC.
	inputRmsL2N	System input RMS voltage between line 2 and neutral in VAC.
	inputRmsL1L2	System input RMS voltage between line 1 and line 2 in VAC.
	inputRmsCurrentL1	System input RMS current for line 1 in A AC.
	inputRmsCurrentL2	System input RMS current for line 2 in A AC.
	inputFrequency	System input frequency in Hz.
	inputMaxVoltageL1N	Maximum system input voltage measurement for Line 1-N since the last reset.
	inputMinVoltageL1N	Minimum system input voltage measurement for Line 1-N since the last reset.
	inputMaxVoltageL2N	Maximum system input voltage measurement for Line 2-N since the last reset.
	inputMinVoltageL2N	Minimum system input voltage measurement for Line 2-N since the last reset.
	inputNomVoltage	Nominal (or rated) system input voltage in VAC.
	inputNomCurrent	Nominal (or rated) system input current in A AC.
	inputNomFrequency	Nominal (or rated) system input frequency in Hz.
bypass		
	bypassInputVoltageL1N	Bypass input RMS voltage between line 1 and neutral in VAC.
	bypassInputVoltageL1L2	Bypass input RMS voltage between line 1 and line 2 in VAC.
	bypassInputVoltageL2N	Bypass input RMS voltage between line 2 and neutral in VAC.
	bypassInputRmsCurrentL1	Bypass input RMS current for line 1 in A AC.
	bypassInputRmsCurrentL2	Bypass input RMS current for line 2 in A AC.
	bypassInputFrequency	Bypass input frequency in Hz.
	bypassNomVoltage	Bypass nominal (or rated) voltage in VAC.
battery		
	batteryStatus	UPS battery status: **normal/low/depleted/unknown**.
	timeRemaining	Calculated time available on battery in minutes.
	percentCharge	Percentage of battery charge.
	chargeStatus	Battery charge status: **fully_charged/not_charging/charging/discharging**.
	dcBusVoltage	Voltage between the positive and negative terminals of the DC bus at the battery input, in VDC.

PARAMETER		DESCRIPTION
	dcBusNomVoltage	Nominal (or rated) voltage between the positive and negative terminals of the DC bus at the battery input, in VDC.
	cabinetType	Type of extended battery cabinets: **internal/external/Irt**.
	rating	Total rating of all parallel strings in the battery in AH.
	numOfEbcInstalled	Total number of extended battery cabinets installed.
	chargeCompensating	Battery charge algorithm changed due to battery temperature: **true/false**.
	chargerState	Current state of the battery charger: **on/off**.
	nomBatteryCapacity	Nominal (or rated) battery capacity time at full load, in minutes.
	dischargeTime	Time on battery operation for this discharge.
	floatVoltage	Cell voltage of the battery at float recharging.
	testResult	Outcome of the previous battery test: **unknown/passed/failed/system_failure/in_progress/unallowed**.
	lowWarningTime	Read-write time in seconds. When battery time remaining falls to, or below, this value, the low battery alarm is activated.
output		
	outputVoltageRmsL1N	System output RMS voltage between Line 1 and Neutral in VAC.
	outputVoltageRmsL1L2	System output RMS voltage between Line 1 and Line 2 in VAC.
	outputRmsCurrentL1	System output RMS current for Line 1 in A AC.
	outputVoltageRmsL2N	System output RMS voltage between Line 2 and Neutral in VAC.
	outputRmsCurrentL2	System output RMS current for Line 2 in A AC.
	outputFrequency	System output frequency in Hz.
	outputMaxVoltageL1N	Maximum system output voltage measurement for Line 1-neutral since last reset.
	outputMinVoltageL1N	Minimum system output voltage measurement for Line 1-neutral since last reset.
	outputMaxVoltageL2N	Maximum system output voltage measurement for Line 2-neutral since last reset.
	outputMinVoltageL2N	Minimum system output voltage measurement for Line 2-neutral since last reset.
	outputPower	Sum total of all system output phases.
	outputPowerL1	System output power on Line 1.
	outputPowerL2	System output power on Line 2.
	outputPercentPower	System output power as a percentage of the rated capacity.
	outputPercentPowerL1	System output power on Line 1 as a percentage of the rated capacity.
	outputPercentPowerL2	System output power on Line 2 as a percentage of the rated capacity.
	outputApparentPower	Sum total apparent power of all system output phases.
	outputApparentPowerL1	System apparent power on Line 1.
	outputApparentPowerL2	System apparent power on Line 2.
	outputNomVoltage	Nominal (or rated) system output voltage.
	outputApparentPowerRating	Output apparent power rating.
	outputNomFrequency	Nominal (or rated) system output frequency.
	outputSource	UPS output source: **other/normal/off/bypass/battery/booster/reducer**.

PARAMETER		DESCRIPTION
	powerFactorCorrection	State of the power factor correction circuitry of the system: on/off.
	nomPowerFactor	Nominal (or rated) system power factor.
	outputOffDelay	Time in seconds to delay before the output shuts off.
	outputOnDelay	Time in seconds to delay before the system turns on.
	outputCycleDelay	Time in seconds to remain off when the output is cycled.
ecoMode		
	ecoModeStatus	Current ECO mode status: **enabled/disabled**.
	ecoModeState	Read-write value to enable/disable ECO mode: **enabled/disabled**.
systemInfo		
	systemStatus	Operating status for the system: **normal_operation/startup/normal_with_warning/normal_with_alarm/abnormal_operation**.
	systemModel	System model identifier.
	firmwareVersion	Firmware version identifier.
	serialNumber	System serial number.
	manufactureDate	Manufacturing data of the system.
	inputBlackoutCount	Number of occurrence, since the last reset, where the input was not qualified to provide power to the system.
	inputBrownoutCount	Number of occurrence, since the last reset, where the system input voltage has fallen below a pre-determined threshold for a specified amount of time.
	inverterState	Inverter state: **on/off**.
	inletAirTemp	Temperature of the inlet air in degrees Celsius.
	shutdownReason	Reason for the most recent shutdown: **none/over_temperature/overload/dc_bus_overload/line_swap/low_battery/remote_command/input_inder_voltage/power_factor_correction_fail/external_signal_command**.
	DcConverterStatus	Operating state of the dc converter.
	upsTopology	UPS topology: **online/offline/line_interactive**.
	bypassInverterInputCfg	Input source configuration for the bypass and inverter: **single_combined_source/dual_separate_sources**.
	autoRestart	Read-write value: **enabled/disabled**. When enabled the UPS will automatically restart the load when utility power is restored after a battery discharge.
	autoRestartDelay	Read-write value in seconds. If power is lost, the control will delay this amount of time in seconds after power is restored before restarting the unit.
	audibleAlarmControl	Read-write audible alarm control: **on/off**.
Events		Value will be **active** if the event is active, otherwise it will be **normal**.
	batteryLow	Calculated battery time remaining has reached the low battery threshold.
	inputProblem	A problem with the input has been detected.
	outputOff	System output is off.
	batteryDischarging	Battery is discharging.
	replaceBattery	Battery is due for replacement.
	batteryTestFail	Battery test failed.
	batteryTestInProgress	Battery self test is in progress.

PARAMETER		DESCRIPTION
	batteryUndervoltage	Battery voltage is too low.
	batteryOvervoltage	Battery voltage has exceeded a predetermined limit.
	loadOnBypass	Output power is supplied by the bypass.
	badBypass	A problem with the bypass has been detected.
	inputUndervoltage	One or more of the input phase voltages had dropped below the limit.
	shutdownPending	One or more of the input phase voltages had exceeded the limit.
	overTemperature	Equipment over temperature summary event.
	outputUndervoltage	One or more of the output phase voltages had dropped below the limit.
	outputOvervoltage	One or more of the output phase voltages had exceeded the limit.
	outputOverload	An overload exists on the output.
	inputBadFrequency	Input frequency is outside the normal range.
	chargerFailure	Charger is off.
	lossOfRedundancy	Multi-mode collection doesn't have enough modules to satisfy the redundancy configuration.
	rectifierFailure	Rectifier is off.
	inverterFailure	Inverter output is off.
	dctoDcConverterFault	Failure has occurred in the battery discharge circuit.
	parallelICommWarning	Parallel communication bus warning.
	maintBypassBreaker	Maintenance bypass breaker is closed (if available).
	fanFailure	One or more fans have failed (if available).
	epoLatched	System output is off - Emergency Power Off latched requires manual reset (if available).
	inputWiringFault	Neutral/ground conductors on the input wiring are not properly bonded, or the line/neutral conductors have been swapped (if available).

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

GET /power/ups/GXT4
Response is:
{

```
    "upsId": "GXT4",
    "vendor": "Net-Liebert",
    "model": "MinValidation",
    "device": "10.207.24.67",
    "input": {
        "inputRmsL1N": "113.0",
        "inputRmsCurrentL1": "0.0",
        "inputFrequency": "59.9",
        "inputMaxVoltageL1N": "118.0",
        "inputMinVoltageL1N": "112.0",
        "inputNomVoltage": "120",
        "inputNomCurrent": "8",
        "inputNomFrequency": "60"
    },
    "bypass": {
        "bypassInputVoltageL1N": "113.0",
        "bypassInputRmsCurrentL1": "0.0",
        "bypassInputFrequency": "59.9",
        "bypassNomVoltage": "120"
    },
    "battery": {
        "batteryStatus": "normal",
        "timeRemaining": "178",
        "percentCharge": "100",
        "chargeStatus": "fully_charged",
        "dcBusVoltage": "54",
        "dcBusNomVoltage": "48",
        "cabinetType": "internal",
        "rating": "9",
        "numOfEbcInstalled": "0",
        "chargeCompensating": "false",
        "chargerState": "on",
        "nomBatteryCapacity": "7",
        "dischargeTime": "0",
        "floatVoltage": "54",
        "testResult": "unknown",
        "lowWarningTime": 28
     },
    "output": {
        "outputVoltageRmsL1N": "0.0",
        "outputRmsCurrentL1": "0.0",
        "outputFrequency": "0.0",
        "outputMaxVoltageL1N": "117.0",
        "outputMinVoltageL1N": "0.0",
        "outputPower": "0.0",
        "outputPercentPower": "0",
        "outputApparentPower": "0.0",
        "outputNomVoltage": "120",
        "outputApparentPowerRating": "1000",
        "outputNomFrequency": "60",
        "outputSource": "off",
        "powerFactorCorrection": "off",
        "nomPowerFactor": "0.9",
        "outputOffDelay": 10,
        "outputOnDelay": 10,
        "outputCycleDelay": 10
    },
    "ecoMode": {
```

```
    "ecoModeStatus": "off",
    "ecoModeState": "enabled"
  },
  "systemInfo": {
    "systemStatus": "normal_with_alarm",
    "systemModel": "MinValidation",
    "firmwareVersion": "U140D170",
    "serialNumber": "1521500545AFB93",
    "manufactureDate": "04AUG15",
    "inputBlackoutCount": "0",
    "inputBrownoutCount": "0",
    "inverterState": "off",
    "inletAirTemp": "20.0",
    "shutdownReason": "remote_command",
    "dcConverterStatus": "off",
    "upsTopology": "online",
    "bypassInverterInputCfg": "single_combined_source",
    "autoRestart": "enabled",
    "autoRestartDelay": 10,
    "audibleAlarmControl": "on"
  },
  "events": {
    "batteryLow": "normal",
    "inputProblem": "normal",
    "outputOff": "active",
    "batteryDischarging": "normal",
    "replaceBattery": "normal",
    "batteryTestFail": "normal",
    "batteryTestInProgress": "normal",
    "batteryUndervoltage": "normal",
    "batteryOvervoltage": "normal",
    "loadOnBypass": "normal",
    "badBypass": "normal",
    "inputUndervoltage": "normal",
    "shutdownPending": "normal",
    "overTemperature": "normal",
    "outputUndervoltage": "normal",
    "outputOvervoltage": "normal",
    "outputOverload": "normal",
    "inputBadFrequency": "normal",
    "chargerFailure": "normal",
    "lossOfRedundancy": "normal",
    "rectifierFailure": "normal",
    "inverterFailure": "normal",
    "dctoDcConverterFault": "normal",
    "parallellCommWarning": "normal",
    "maintBypassBreaker": "normal",
    "fanFailure": "normal",
    "epoLatched": "normal",
    "inputWiringFault": "normal"
  }
}
```


PUT /power/ups/GXT4


```
{
      "battery": {"lowWarningTime" : 30},
      "ecoMode": {"ecoModestate" : "enabled"}
}
```

```
PUT /power/pdus/GXT4
{
      "systemInfo": {"autoRestart" : "enabled"},
                                    {"autoRestartDelay" : 10"},
                                    {"audibleAlarmControl" : "off"}
}
```

2.17.25 /power/ups/<NAME>/rename

Rename the UPS

Methods

POST

Parameters

PARAMETER	DESCRIPTION
upsId	New name for the UPS

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
POST /power/ups/GXT4/rename
            {
                  "upsId" : "newname"}
            }
Response is:
        {
            "status": "success. ups renamed."
        }
```

2.17.26 /power/ups/<NAME>/restoreName

Restore the default name for a serially-connected UPS. Does not apply to Net-UPS.

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

POST /power/ups/GXT4/restoreName

Response is:

```
{
    "status": "success. restored ups name."
}
```

2.17.27 /power/ups/<NAME>/outputOn

Turns the UPS output on.

Methods

POST

Parameters

Parameter is optional. If the parameter is omitted, the delay will default to 10 seconds.

PARAMETER	DESCRIPTION
outputOnDelay	Delay time in seconds

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
POST /power/ups/GXT4/outputOn
        {
            "output": {"outputOnDelay" : 10}
        }
Response is:
    {
        "status": "success. output on ups GXT3 will be turned on, delay 10 seconds."
    }
```

2.17.28 /power/ups/<NAME>/outputOff

Turns the UPS output off

Methods

POST

Parameters

Parameter is optional. If the parameter is omitted, the delay will default to 10 seconds.

PARAMETER	DESCRIPTION
outputOffDelay	Delay time in seconds

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
POST /power/ups/GXT4/outputOff
        {
            "output": {"outputOffDelay" : 10}
        }
```

```
Response is:
        {
            "status": "success. output on ups GXT3 will be turned off, delay 10 seconds."
        }
```

2.17.29 /power/ups/<NAME>/outputCycle

Cycle the UPS output off and back on again

Methods

POST

Parameters

Parameter is optional. If the parameter is omitted, the delay will default to 10 seconds.

PARAMETER	DESCRIPTION
outputCycleDelay	Time in seconds that the output will remain off

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
POST /power/ups/GXT4/outputCycle
            {
                "output": {"outputCycleDelay" : 10}
            }
Response is:
        {
            "status": "success. Output on ups GXT4 cycle initiated, delay 10 seconds."
        }
```

2.17.30 /power/ups/<NAME>/testBattery

Initiate battery self test.

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

POST /power/ups/GXT4/testBattery
Response is:

```
{
    "status": "success. Started battery test on ups GXT4"
}
```

2.17.31 /power/ups/<NAME>/silenceAlarm

Silence the audible alarm.

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

POST /power/ups/GXT4/silenceAlarm
Response is:

```
{
    "status": "success. alarm silenced on ups GXT4"
}
```

2.17.32 /power/ups/<NAME>/resetPowerStats

Reset the power statistics.

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

POST /power/ups/GXT4/resetPowerStats
Response is:

```
{
    "status": "success. Power statistics reset on ups GXT4"
}
```

2.17.33 /power/ups/<NAME>/outletGroups[/<ID>]

This resource provides the ability to view and control the controllable outlet groups on the UPS. Not all UPS devices have controllable outlet groups.

Methods

GET, PUT, PATCH

Parameters

PARAMETER	DESCRIPTION
outlet	Read only integer index.
name	Read only UPS outlet group name.
status	Read only UPS outlet group status: **on/off/invalid/pending_on/pending_off/pending_cycle/pending**

Examples

GET /power/ups/GXT4/outletGroups

Response is:

```
{
    "outletGroups": [
        {
            "number": "1",
            "name": "GXT4_1",
            "status": "off"
        },
        {
            "number": "2",
            "name": "GXT4_2",
            "status": "off"
        }
    ]
}
```

2.17.34 /power/login

This resource allows the user to configure passwords for the various brands of PDUs.

Methods

GET, PUT, PATCH

Parameters

PARAMETER	DESCRIPTION
avocentPduUsername	Read only username used to access Avocent-Cyclades PDUs. "admin"
avocentPduPassword	Password used to access Avocent-Cyclades PDUs. Default "pm8".
spcPduUsername	Read only username used to access SPC PDUs. "admn"
spcPduPassword	Password used to access SPC PDUs. Default "admn".
serverTechPduUsername	Read only username used to access ServerTech PDUs. "admn"
serverTechPduPassword	Password used to access ServerTech PDUs. Default "admn".

PARAMETER	DESCRIPTION
raritanPduUsername	Read only username used to access Raritan PDUs. "admin"
raritanPduPassword	Password used to access Raritan PDUs. Default "admin".
eatonPduUsername	Read only username used to access Eaton PDUs. "admin"
eatonPduPassword	Password used to access Eaton PDUs. Default "admin".
apcPduUsername	Read only username used to access APC PDUs. "apc"
apcPduPassword	Password used to access APC PDUs. Default "apc".
vertivPduUsername	Read only username used to access Vertiv PDUs. "admin"
vertivPduPassword	Password used to access Vertiv PDUs. Default "admin"
geistPduUsername	Username used to access Geist PDUs. There is no default value.
geistPduPassword	Password used to access Geist PDUs. There is no default value.

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

GET /power/login


```
{
    "avocentCycladesPduUsername": "admin",
    "avocentCycladesPduPassword": "",
    "spcPduUsername": "admn",
    "spcPduPassword": "",
    "serverTechPduUsername": "admn",
    "serverTechPduPassword": "",
    "raritanPduUsername": "admn",
    "raritanPduPassword": "",
    "eatonPduUsername": "admn",
    "eatonPduPassword": "",
    "apcPduUsername": "admn",
    "apcPduPassword": "",
    "vertivPduUsername": "admn",
    "vertivPduPassword": "",
    "geistPduUsername": "admn",
    "geistPduPassword": ""
}
```


PUT /power/login


```
{
```

```
            "avocentCycladesPduPassword": "password",
            "geistPduUsername": "test",
            "geistPduPassword": "testpassword"
        }
```

2.17.35 /power/outletGroups[/<NAME>]

This resource provides the ability to view information about existing Outlet Groups, to create new Outlet Groups or delete existing Outlet Groups.

Methods

GET, POST, DELETE

Parameters

PARAMETER	DESCRIPTION
name	Read-only name of the outlet group
status	Read-only outlet group status: **on/off/on_fixed/on_locked/on_unlocked/on_pend_off/off_pend_on/on_pend_cycle/off_cycle/not_set/unconfigured**
outlets	Read-only list of outlets in the group
powerConsumption	Read-only power draw in watts for the group

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

GET /power/outletGroups
Response is:

```
{
    "outletGroups": [
        {
            "name": "group1",
            "status": "on",
            "outlets": "MPX[17-18], PX2[5,7,19-20]",
            "powerConsumption": "0.0"
        },
        {
```

```
            "name": "group2",
            "status": "unconfigured",
            "powerConsumption": "N/A"
        },
    ]
}

To create a group:
            POST /power/outletGroups
            {
            "name" : "newGroup"
            }
        Response is:
        {
        "name": "newGroup"
        }

To delete a group:
DELETE /power/outletGroups/newGroup
```

2.17.36 /power/outletGroups/<NAME>/on

Turns all the outlets in the outlet group on.

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
POST /power/outletGroups/group1/on
Response is:
{
    "status": "success. outletgroup group1 turned on."
}
```

2.17.37 /power/outletGroups/<NAME>/off

Turns all the outlets in the outlet group off.

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
POST /power/outletGroups/group1/off
Response is:
{
    "status": "success. outletgroup group1 turned off."
}
```

2.17.38 /power/outletGroups/<NAME>/cycle

Cycles all the outlets in the outlet group off and back on. All outlets will be left in the on state.

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

POST /power/outletGroups/group1/cycle
Response is:

```
{
    "status": "success. outletgroup group1 cycled."
}
```

2.17.39 /power/outletGroups/<NAME>/outlets

This resource provides the ability to view detailed information about the outlets in an Outlet Group, to add outlets to the group and to remove existing outlets.

Methods

GET, POST

Parameters

PARAMETER	DESCRIPTION
id	Id for the outlet consisting of the PDU name and the outlet number
pduId	Name of the PDU for this outlet
outlet	Outlet number
status	Outlet status: **on/off/on_fixed/on_locked/on_unlocked/on_pend_off/off_pend_on/on_pend_cycle/off_cycle/not_set**
powerConsumption	Read-only power draw in watts for the group

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

GET /power/outletGroups/group1/outlets
Response is:

```
{
    "outlets": [
     {
        "id"  : "PX2~5",
        "pduId": "PX2",
        "outlet": "5",
        "status": "on",
        "powerConsumption": "0.0"
     },
     {
        "id"   : "MPX~5"
        "pduId" : "MPX",
        "outlet": "5",
        "status": "on",
        "powerConsumption": "0.0"
     }
   ]
}
```

To add an outlet (or a range of outlets):

POST /power/outletGroups/group1/outlets

```
{
      "pduId"  : "PX2",
      "outlet" : "7,12-15"
}
```

Response is:

```
{
    "pduId" : "PX2",
    "outlet": "7,12-15"
}
```

2.17.40 /power/outletGroups/<NAME>/outlets/<ID>

This resource provides the ability to delete an outlet from an existing outlet group. The ID should consist of the PDU name and the outlet number, in this format: "<pduname>~<outletnumber>".

Methods

DELETE

Parameters

None

Response Body

None

Response Codes

200	OK
400	Bad Request

Examples

```
DELETE /power/outletGroups/group1/outlets/PX2~5
```

2.17.41 /power/networkPdus[/<ADDR>]

This resource provides the ability to view information about all known Network PDUs, to add new Network PDUs, and to modify existing entries.

Methods

GET, PUT, PATCH, POST, DELETE

Parameters

PARAMETER	DESCRIPTION
ipAddress	IP Address of the PDU. Writeable only on a POST when the entry is created
pduType	Type of the network PDU: **net-mph-mpx / net-avocent-pmhd / net-geist / net-servertech / net-servertech-pro2 / net-eaton / net-raritan / net-apc-rpdu / net-apc-rpdu2**
snmpVersion	SNMP version: v1 / v2c
snmpCommunityName	SNMP community name. Example: "public"
pollingRate	Polling rate of the network PDU in seconds

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

```
GET /power/networkPdus
{
        "pdus": [
```

```
            {
              "ipAddress": "10.207.24.168",
              "pduType": "net-geist",
              "snmpVersion": "v2c",
              "snmpCommunityName": "private",
              "pollingRate": 60
            },
            {
              "ipAddress": "10.207.24.85",
              "pduType": "net-mph",
              "snmpVersion": "v2c",
              "snmpCommunityName": "VertivRackPDU",
              "pollingRate": 60
            }
          ]
        }
```

PUT /power/networkPdus/10.207.24.63

```
        {
            "snmpVersion": "v2c",
            "snmpCommunityName": "private",
            "pollingRate": 30
        }
```

To add a Net-PDU:

POST /power/networkPdus

```
        {
            "ipAddress": "10.20.30.42",
            "pduType": "net-geist",
            "snmpVersion": "2c",
            "snmpCommunityName": "public",
            "pollingRate": 60
        }
```

Response is:

```
        {
            "ipAddress": "10.20.30.40",
            "pduType": "net-geist",
            "snmpVersion": "2c",
            "snmpCommunityName": "public",
            "pollingRate": 60
        }
```

To remove a Net-PDU:

DELETE /power/networkPdus/10.207.24.63

2.17.42 /power/networkUps[/<ADDR>]

This resource provides the ability to view information about all known Network UPS devices, to add new Network UPS device, and to modify existing entries.

Methods

GET, PUT, PATCH, POST, DELETE

Parameters

PARAMETER	DESCRIPTION
ipAddress	IP Address of the UPS. Writeable only on a POST when the entry is created
pduType	Type of the network PDU: **net-liebert**
snmpVersion	SNMP version: **v1 / v2c**
snmpCommunityName	SNMP community name. Example: "public"
pollingRate	Polling rate of the network UPS in seconds

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

GET /power/networkUps

```
{
    "ups": [
      {
        "ipAddress": "10.207.24.67",
        "upsType": "net-liebert",
        "snmpVersion": "v2c",
        "snmpCommunityName": "public",
        "pollingRate": 30
      },
      {
        "ipAddress": "10.207.24.81",
        "upsType": "net-liebert",
        "snmpVersion": "v2c",
        "snmpCommunityName": "public",
        "pollingRate": 60
      }
    ]
  }
```

PUT /power/networkUps/10.207.24.673

```
{
        "snmpVersion": "v2c",
        "snmpCommunityName": "private",
        "pollingRate": 30
    }
```

To add a Net-PDU:

POST /power/networkUps

```
{
        "ipAddress": "10.207.24.81",
        "pduType": "net-liebert",
        "snmpVersion": "2c",
        "snmpCommunityName": "public",
        "pollingRate": 60
}
```

Response is:

```
{
    "ipAddress": "10.20.24.81",
    "pduType": "net-liebert",
    "snmpVersion": "2c",
    "snmpCommunityName": "public",
    "pollingRate": 60
}
```

To remove a Net-UPS:

DELETE /power/networkUps/10.207.24.81

2.18 Sensors

2.18.1 /sensors/internal

This resource provides access to the internal temperature sensors and voltages of the appliance. Temperatures and voltages are read-only fields, but the minimum, maximum and threshold values can be set for triggering alerts.

2.18.2 Methods

GET, PUT, PATCH

2.18.3 Parameters

PARAMETER	DESCRIPTION
currentCPUTemperature	Current temperature of the CPU in degrees Celsius. Read-only.
maximumCPUTemperature	Maximum CPU Temperature in degrees Celsius.
maximumCPUTemperatureThreshold	Maximum CPU Temperature Threshold in degrees Celsius.
minimumCPUTemperature	Minimum CPU Temperature in degrees Celsius.
minimumCPUTemperatureThreshold	Minimum CPU Temperature Threshold in degrees Celsius.
currentBoardTemperature	Current temperature of the board in degrees Celsius. Read-only.
maximumBoardTemperature	Maximum Board Temperature in degrees Celsius.
maximumBoardTemperatureThreshold	Maximum Board Temperature Threshold in degrees Celsius.
minimumBoardTemperature	Minimum Board Temperature in degrees Celsius.
minimumBoardTemperatureThreshold	Minimum Board Temperature Threshold in degrees Celsius.
voltagePSInternal	PS Internal Supply [0.95v ~ 1.05v]. Float. Read-only.
voltagePLInternal	PL Internal Supply [0.95v ~ 1.05v]. Float. Read-only.
voltagePSAuxiliary	PS Auxiliary Supply [1.71v ~ 1.89v]. Float. Read-only.

PARAMETER	DESCRIPTION
voltagePLAuxiliary	PL Auxiliary Supply [1.71v ~ 1.89v]. Float. Read-only.
voltagePSDDR3	PS DDR3 Supply [1.31v ~ 1.39v]. Float. Read-only.
voltagePLBlockRam	PL Block RAM Supply [0.95v ~ 1.05v]. Float. Read-only.
voltagePowerSupply1	Power Supply 1 voltage [11.06v ~ 12.98v]. Float. Read-only.
voltagePowerSupply2	Power Supply 2 voltage [11.06v ~ 12.98v]. Float. Read-only (if present).

2.18.4 Query

Fields are supported for all parameters.

2.18.5 Response Body

JSON object

2.18.6 Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

2.18.7 Examples

```
GET /sensors/internal
   {
       "currentCPUTemperature": 52,
       "maximumCPUTemperature": 0,
       "maximumCPUTemperatureThreshold": 0,
       "minimumCPUTemperature": 0,
       "minimumCPUTemperatureThreshold": 0
       "currentBoardTemperature": 32,
       "maximumBoardTemperature": 0,
       "maximumBoardTemperatureThreshold": 0,
       "minimumBoardTemperature": 0,
       "minimumBoardTemperatureThreshold": 0
       "voltagePSInternal": 0.97,
       "voltagePLInternal": 0.97,
       "voltagePSAuxiliary": 1.79,
       "voltagePLAuxiliary": 1.79,
       "voltagePSDDR3": 1.34,
       "voltagePLBlockRAM": 0.97,
       "voltagePowerSupply1": 11.66,
       "voltagePowerSupply2": 11.64
   }
PUT /sensors/internal
   {
          "maximumCPUTemperature": 54,
          "minimumCPUTemperature": 45
     }
```

2.18.8 /sensors/1Wire[/<ADDRESS>]

This resource provides access to the attached 1-wire sensors.

2.18.9 Methods

GET, PUT, PATCH, POST, DELETE

2.18.10 Parameters

PARAMETER	DESCRIPTION
address	Physical 1-wire address. Read-only string.
name	User assignable name of the sensor.
location	User assignable location of the sensor.
type	Type of sensor. Read-only string.
value1	Read-only value of sensor element.
value2	Read-only value of sensor element. Contact sensors only.
value3	Read-only value of sensor element. Contact3 sensors only.
maximum	Maximum observed value. Analog sensors only. Read-only (if present).
minimum	Minimum observed value. Analog sensors only. Read-only (if present).
average	Average observed value. Analog sensors only. Read-only (if present).
unit	Sensor units, such as F or C for temperature. Analog sensors only.
thresholdLowWarning	Analog sensors only. Blank for none.
thresholdLowCritical	Analog sensors only. Blank for none.
thresholdHighWarning	Analog sensors only. Blank for none.
thresholdHighCritical	Analog sensors only. Blank for none.
alarm	Analog sensors only. **enabled/disabled**
alarm1	Contact sensors only. **disabled/alarm_when_open/alarm_when_closed**.
alarm2	Contact sensors only. **disabled/alarm_when_open/alarm_when_closed**.
alarm3	Contact sensors only. **disabled/alarm_when_open/alarm_when_closed**.
filterTime	Leak sensors only. Integer 1 or a multiple of 15 up to 135.
leakAlarm	Leak sensors only. **enabled/disabled**.
cableFailAlarm	Leak sensors only. **enabled/disabled**.

2.18.11 Query

Fields are supported for all parameters.

2.18.12 Response Body

JSON object

2.18.13 Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

2.18.14 Examples

```
GET /sensors/1Wire
      {
        "1wire": [
            {
                "address": "72390C000000",
                "name": "20.72390C000000",
                "location": "",
                "type": "Dry Contact SN-3C",
                "value1": "sensor_open",
                "value2": "sensor_open",
                "value3": "sensor_closed",
                "alarm1": "alarm_when_closed",
                "alarm2": "alarm_when_open",
                "alarm3": "disabled"
            },
            {
                "address": "E38BB0000000",
                "name": "outside_humidity",
                "location": "outside",
                "type": "External Humidity",
                "value1": "39.03 %",
                "maximum": 43.54,
                "minimum": 38.70,
                "average": 38.94,
                "unit": "%",
                "thresholdLowWarning": "",
                "thresholdLowCritical": "10.5",
                "thresholdHighWarning": "",
                "thresholdHighCritical": "85.0",
                "alarm": "disabled"
            },
            {
                "address": "730000003000",
                "name": "leak sensor",
                "location": "closet",
                "type": "Leak SN-L",
                "value1": "normal",
                "filterTime": 30,
                "leakAlarm": "enabled",
                "cableFailAlarm": "disabled"
            },
            ...
        ]
  }
PUT /sensors/1Wire/72390C000000
    {
              "name": "mydoor",
              "alarm1": "alarm_when_open",
    }
```

```
POST /sensors/1Wire/refresh
  {
        "status": "initiated 1-Wire update"
  }
```

This refreshes the list of sensors by purging the list and redetecting attached 1-wire devices.

DELETE /sensors/1Wire/72390C000000

This resets the values, if applicable, of the specified sensor.

2.18.15 /sensors/digitalIn[/<ID>]

This resource provides access to the attached Digital In sensors.

2.18.16 Methods

GET, PUT, PATCH

2.18.17 Parameters

PARAMETER	DESCRIPTION
position	Digital In position, integer value 1-4. Read-only.
name	User assignable name of the digital in sensor.
location	User assignable location of the sensor.
type	Type of sensor.
alarm	Alarm enable. **disabled/alarm_when_open/alarm_when_closed**
value	Read-only value of sensor element.

2.18.18 Query

Fields are supported for all parameters.

2.18.19 Response Body

JSON object

2.18.20 Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

2.18.21 Examples

```
GET /sensors/digitalIn
      {
          "digitalIn": [
              {
```

```
                "position": 1,
                "name": "1",
                "location": "",
                "type": "vibration",
                "alarm": "disabled",
                "value": "sensor_open"
            },
                "position": 2,
                "name": "1",
                "location": "",
                "type": "vibration",
                "alarm": "disabled",
                "value": "sensor_open"
            },
                "position": 3,
                "name": "1",
                "location": "",
                "type": "vibration",
                "alarm": "disabled",
                "value": "sensor_open"
            },
                "position": 4,
                "name": "mymotionsensor",
                "location": "myoffice",
                "type": "motion_ad_im",
                "alarm": "enabled_when_open",
                "value": "sensor_closed"
            }
        ]
    }
GET /sensors/digitalIn/1
    {
        "position": 1,
        "name": "1",
        "location": "",
        "type": "vibration",
        "alarm": "disabled",
        "value": "sensor_open"
    }
PUT /sensors/digitalIn/2
        {
            "name": "mydoor",
            "location": "myoffice",
            "alarm": "alarm_when_open"
        }
```


2.18.22 /sensors/pdu

This resource provides access to the sensors attached to PDUs.

Methods

GET, POST

Parameters

PARAMETER	DESCRIPTION
name	Read-only name of the sensor.
pdu	Read-only name of the PDU to which the sensor is connected.
type	Type of sensor.
value	Read-only value of sensor element.
maximum	Maximum observed value. Analog sensors only. Read-only (if present).
minimum	Minimum observed value. Analog sensors only. Read-only (if present).
average	Average observed value. Analog sensors only. Read-only (if present).

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
GET /sensors/pdu
        {
            "sensors": [
                {
                    "name": "T-SNSR-RCU-OD.1.1",
                    "pdu": "RCU-OD",
                    "type": "External Temperature",
                    "value": "27.0C (80.0F)",
                    "maximum": "27.0C (80.0F)",
                    "minimum": "27.0C (80.0F)",
                    "average": "27.0C (80.0F)"
                },
                {
                    "name": "H-SNSR-RCU-OD.1.2",
                    "pdu": "RCU-OD",
                    "type": "External Humidity",
                    "value": "51.0%",
                    "maximum": "51.0%",
                    "minimum": "51.0%",
                    "average": "51.0%"
                },
                {
                    "name": "A-SNSR-RCU-OD.1.3",
                    "pdu": "RCU-OD",
                    "type": "Air Flow",
```

```
                "value": "46",
                "maximum": "46",
                "minimum": "46",
                "average": "46"
            },
            {
                "name": "W-SNSR-RCU-OD.1.4",
                "pdu": "RCU-OD",
                "type": "Dewpoint",
                "value": "16.0C (60.5F)",
                "maximum": "16.0C (60.5F)",
                "minimum": "16.0C (60.5F)",
                "average": "16.0C (60.5F)"
            }
        ]
    }
```


2.18.23 sensors/pdu/<NAME>/reset

This resource provides the ability to reset the minimum/maximum/average values for a sensor.

Methods

GET, POST

Parameters

None

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

POST /sensors/pdu/A-SNSR-RCU-OD.1.3/reset
Response is:


```
    {
        "status": "success. sensor reset."
    }
```

2.19 Digital Out

2.19.1 /digitalOut

This resource provides access to the digital out ports, if present, on the appliance. It allows for naming the ports, reading their current state, or changing the state.

Methods

GET, PUT, PATCH

Parameters

PARAMETER	DESCRIPTION
position	Integer position of the output, either 1 or 2. Read-only.
name	Name assigned to the output.
state	State of the output: **on / off**

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
204	OK No Content (for a PUT or PATCH)
400	Bad Request

Examples

```
GET /digitalOut
        {
            "digitalOut": [
                {
                    "position": 1,
                    "name": "lights",
                    "state": "on"
                },
                {
                    "position": 2,
                    "name": "alarm",
                    "state": "off"
                }
            ]
        }
GET /digitalOut/1
        {
```

```
            "position": 1,
            "name": "lights",
            "state": "on"
        }
PUT /digitalOut/2           {
                    "state": "on"
            }
```

2.20 Monitoring

2.20.1 /monitoring/network/devices[/<INT>]

This resource provides information about the network devices and their current IP addresses.

Methods

GET

Parameters

PARAMETER	DESCRIPTION
deviceName	Device name: **eth#, bond#**
status	Current device status. **enabled/disabled/plugged/unplugged**
ipv4Address	Current device IPv4 Address.
ipv4Mask	Current device IPv4 Mask.
ipv6Address	Current device IPv6 Address.

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

```
GET /monitoring/network/devices
            {
                "devices": [
                    {
                        "deviceName": "eth0",
                        "status": "enabled",
```

```
                    "ipv4Address": "10.20.30.40",
                    "ipv4Mask": "255.255.252.0",
                    "ipv6Address": "2001:cdba:0000:0000:0000:0000:5678:1234"
                },                    ...
            ]
        }
GET /monitoring/network/devices/eth2
        {
            "deviceName": "eth2",
            "ipv4Address": "10.20.30.42",
            "ipv4Mask": "255.255.255.0",
            "ipv6Address": ""
        }
```

2.20.2 /monitoring/network/routingTables/ipv4

This resource returns the ipv4 routing tables.

Methods

GET

Parameters

PARAMETER	DESCRIPTION
table	Multiple routing only.
from	Multiple routing only.
destination	
gateway	
genmask	Not for multiple routing.
flags	Not for multiple routing.
metric	Not for multiple routing.
ref	Not for multiple routing.
use	Not for multiple routing.
source	Multiple routing only.
iface	

Query

Fields and filtering are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

GET /monitoring/network/routingTables/ipv4

```
{
"destination": "default",
"gateway": "*",
"genmask": "255.255.252.0",
"flags": "UG",
"metric": 0,
"ref": 0,
"use": 0,
"iface": "eth0"
}
```

When multiple routing tables are enabled (advanced routing):

```
GET /monitoring/network/routingTables/ipv4
{
"table": "default",
"from": "*",
"destination": "default",
"gateway": "*",
"source": "192.168.161.10",
"iface": "eth0"
}
```

2.20.3 /monitoring/network/routingTables/ipv6

This resource returns the ipv6 routing tables.

Methods

GET

Parameters

PARAMETER	DESCRIPTION
destination	
nexthop	
flags	
metric	
ref	
use	
ifac	

Query

Fields and filtering are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

GET /monitoring/network/routingTables/ipv6

```
{
"destination": "::/0",
"nexthop": "::",
"flags": "UGDAe"
"metric": 1024,
"ref": 0,
"use": 0,
"iface": "eth0"
}
```

2.20.4 /monitoring/serialPorts[/<NUM>]

This resource provides information about the current state of the serial ports including transmission counts.

The DELETE method is used on an individual serial port to clear the counters to zero.

Methods

GET, DELETE

Parameters

PARAMETER	DESCRIPTION
port	Serial port number. Integer. Read only.
deviceName	Serial port device name: Example: ttyS2
profile	Port profile: cas/power/dial_in/dial_out/socket_client/unconfigured
settings	Physical port settings including speed and parity. Example: 9600 8N1.
signals	Current serial signals potentially including: RTS, CTS, DTR, DSR, CD, RI
txBytes	Number of transmitted bytes. Integer.
rxBytes	Number of received bytes. Integer.
frameErrors	Number of frame errors. Integer.
parityErrors	Number of parity errors. Integer.
break	Number of breaks. Integer.
overrun	Number of overruns. Integer.

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

GET /monitoring/serialPorts


```
{
"serialPorts": [
{
"port": "1",
"deviceName": "ttyS1",
"profile": "cas",
"settings": "9600 8N1",
"signals": "CTS|DSR|CD|RI",
"txBytes": 0,
"rxBytes": 967,
"frameErrors": 0,
"parityErrors": 0,
"break": 0,
"overrun": 0
},
{
"port": "2",
```

```
"deviceName": "ttyS2",
"profile": "power",
"settings": "9600 8N1",
"signals": "RTS|CTS|DTR|DSR|CD|RI",
"txBytes": 21,
"rxBytes": 510,
"frameErrors": 0,
"parityErrors": 0,
"break": 0,
"overrun": 0
},
...
]
}
```


GET /monitoring/serialPorts/1


```
{
"port": "1",
"deviceName": "ttyS1",
"profile": "cas",
"settings": "9600 8N1",
"signals": "CTS|DSR|CD|RI",
"txBytes": 0,
"rxBytes": 967,
"frameErrors": 0,
"parityErrors": 0,
"break": 0,
"overrun": 0
}
```


DELETE /monitoring/serialPorts/2

2.20.5 /monitoring/ipsec[/<NAME>]

This resource provides information about the status of the IPsec connections and tunnel status.

Methods

GET

Parameters

PARAMETER	DESCRIPTION
connectionName	IPSec connection name.
status	Connection status: down/connecting/established.
remoteIpAddress	
lifetime	
establishedTime	

PARAMETER	DESCRIPTION
phase1Algorithm	
phase2Algorithm	
certificateName	

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Examples

GET /monitoring/ipsec

```
{
"ipsecConnections": [
{
"connectionName": "testConnection",
"status": "DOWN",
"remoteIpAddress": "10.20.30.40",
"lifetime": "24h",
"establishedTime": "",
"phase1Algorithm": "",
"phase2Algorithm": "",
"certificateName": ""
},
...
]
}
```

GET /monitoring/ipsec/testConnection

```
{
"connectionName": "testConnection",
"status": "DOWN",
"remoteIpAddress": "10.20.30.40",
"lifetime": "24h",
"establishedTime": "",
"phase1Algorithm": "",
"phase2Algorithm": "",
"certificateName": ""
}
```

2.21 Access

2.21.1 /access/serialPorts[/PORT#]

Returns a list of all serial ports which have a profile setting of CAS.

Methods

GET

Parameters

PARAMETER	DESCRIPTION
port	Read-only port number
name	Read-only name of the serial port
status	Status of the serial port: **idle/in_use**

Query

Fields are supported for all parameters.

Response Body

JSON object

Response Codes

200	OK
401	Not Authorized
40x	Failure

Example

GET /access/serialPorts/1
Response is:

```
{
    "port": "1",
    "name": "port1",
    "status": "idle"
}
```

2.21.2 /access/serialPorts/<PORT#>/on

Turns on the power outlets merged with this port.

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Example

POST /access/serialPorts/1/on
Response is:

```
{
    "status": "success. target powered on"
}
```

2.21.3 /access/serialPorts/<PORT#>/off

Turns off the power outlets merged with this port.

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Example

```
POST /access/serialPorts/1/off
Response is:
        {
            "status": "success. target powered off"
        }
```

2.21.4 /access/serialPorts/<PORT#>/cycle

Cycle power on the power outlets merged with this port.

Methods

POST

Parameters

None

Query

None

Response Body

JSON object

Response Codes

200	OK
400	Bad Request

Example

```
POST /access/serialPorts/1/cycleResponse is:
        {
            "status": "success. target power cycled"
        }
```

2.22 Miscellaneous Resources

2.22.1 /resources

This command provides a list of the available resources and methods.

Methods

GET

Parameters

None

Query

None

Response Codes

200	OK
40x	Failure

Examples

GET /resources

```
{
    "resources": [
        "GET    /access/serialPorts",
        "GET    /access/serialPorts/:PORT",
        "POST   /access/serialPorts/:PORT/cycle",
        "POST   /access/serialPorts/:PORT/off",
        "POST   /access/serialPorts/:PORT/on",
        "GET    /accessRights/:BASE/:NAME/appliance",
        "PATCH  /accessRights/:BASE/:NAME/appliance",
        "PUT    /accessRights/:BASE/:NAME/appliance",
        "GET    /accessRights/:BASE/:NAME/outlets",
        "POST   /accessRights/:BASE/:NAME/outlets",
        "DELETE /accessRights/:BASE/:NAME/outlets/:ID",
        "GET    /accessRights/:BASE/:NAME/pdus",
        "POST   /accessRights/:BASE/:NAME/pdus",
        "DELETE /accessRights/:BASE/:NAME/pdus/:ID",
        "GET    /accessRights/:BASE/:NAME/serialPorts",
        "POST   /accessRights/:BASE/:NAME/serialPorts",
        "DELETE /accessRights/:BASE/:NAME/serialPorts/:ID",
        "GET    /accessRights/:BASE/:NAME/serialPorts/:ID",
        "PATCH  /accessRights/:BASE/:NAME/serialPorts/:ID",
        "PUT    /accessRights/:BASE/:NAME/serialPorts/:ID",
        "GET    /accessRights/:BASE/:NAME/ups",
        "POST   /accessRights/:BASE/:NAME/ups",
        "DELETE /accessRights/:BASE/:NAME/ups/:ID",
        "GET    /accessRights/:BASE/:NAME/upsOutletGroups",
        "POST   /accessRights/:BASE/:NAME/upsOutletGroups",
        "DELETE /accessRights/:BASE/:NAME/upsOutletGroups/:ID",
        "GET    /accessRights/dsview",
        "PATCH  /accessRights/dsview",
        "PUT    /accessRights/dsview",
        "GET    /authentication",
        "PATCH  /authentication",
        "PUT    /authentication",
        "GET    /authentication/dsview",
        "PATCH  /authentication/dsview",
```

```
"PUT    /authentication/dsview",
"GET    /authentication/kerberos",
"PATCH  /authentication/kerberos",
"PUT    /authentication/kerberos",
"GET    /authentication/ldap",
"PATCH  /authentication/ldap",
"PUT    /authentication/ldap",
"GET    /authentication/radius",
"PATCH  /authentication/radius",
"PUT    /authentication/radius",
"GET    /authentication/tacacs",
"PATCH  /authentication/tacacs",
"PUT    /authentication/tacacs",
"GET    /auxPorts",
"GET    /auxPorts/:PORT",
"PATCH  /auxPorts/:PORT",
"PUT    /auxPorts/:PORT",
"GET    /casProfile",
"PATCH  /casProfile",
"PUT    /casProfile",
"GET    /casProfile/autoAnswer",
"POST   /casProfile/autoAnswer",
"DELETE /casProfile/autoAnswer/:ID",
"GET    /casProfile/autoAnswer/:ID",
"GET    /casProfile/matchStrings",
"POST   /casProfile/matchStrings",
"DELETE /casProfile/matchStrings/:ID",
"GET    /casProfile/matchStrings/:ID",
"GET    /casProfile/probeStrings",
"POST   /casProfile/probeStrings",
"DELETE /casProfile/probeStrings/:ID",
"GET    /casProfile/probeStrings/:ID",
"GET    /dialinProfile",
"PATCH  /dialinProfile",
"PUT    /dialinProfile",
"GET    /dialinProfile/callbackUsers",
"POST   /dialinProfile/callbackUsers",
"DELETE /dialinProfile/callbackUsers/:NAME",
"GET    /dialinProfile/callbackUsers/:NAME",
"PATCH  /dialinProfile/callbackUsers/:NAME",
"PUT    /dialinProfile/callbackUsers/:NAME",
"GET    /dialinProfile/pppOtpUsers",
"POST   /dialinProfile/pppOtpUsers",
"DELETE /dialinProfile/pppOtpUsers/:NAME",
"GET    /dialinProfile/pppOtpUsers/:NAME",
"GET    /digitalOut",
"GET    /digitalOut/:ID",
"PATCH  /digitalOut/:ID",
"PUT    /digitalOut/:ID",
"GET    /events",
"GET    /events/:ID",
"PATCH  /events/:ID",
"PUT    /events/:ID",
"GET    /events/applianceLogging",
"PATCH  /events/applianceLogging",
"PUT    /events/applianceLogging",
"GET    /events/dataBuffering",
"PATCH  /events/dataBuffering",
```

```
"PUT    /events/dataBuffering",
"GET    /events/dsview",
"PATCH  /events/dsview",
"PUT    /events/dsview",
"GET    /events/email",
"PATCH  /events/email",
"PUT    /events/email",
"GET    /events/sms",
"PATCH  /events/sms",
"PUT    /events/sms",
"GET    /events/snmp",
"PATCH  /events/snmp",
"PUT    /events/snmp",
"GET    /events/syslog",
"PATCH  /events/syslog",
"PUT    /events/syslog",
"GET    /events/trapForward",
"POST   /events/trapForward",
"DELETE /events/trapForward/:ID",
"GET    /events/trapForward/:ID",
"PATCH  /events/trapForward/:ID",
"PUT    /events/trapForward/:ID",
"GET    /groups",
"POST   /groups",
"DELETE /groups/:NAME",
"GET    /groups/:NAME",
"PATCH  /groups/:NAME",
"PUT    /groups/:NAME",
"GET    /modems",
"GET    /modems/:PORT",
"PATCH  /modems/:PORT",
"PUT    /modems/:PORT",
"GET    /monitoring/ipsec",
"GET    /monitoring/ipsec/:NAME",
"GET    /monitoring/network/devices",
"GET    /monitoring/network/devices/:INT",
"GET    /monitoring/network/routingTables/:TABLE",
"GET    /monitoring/serialPorts",
"DELETE /monitoring/serialPorts/:ID",
"GET    /monitoring/serialPorts/:ID",
"GET    /network/devices",
"GET    /network/devices/:INT",
"PATCH  /network/devices/:INT",
"PUT    /network/devices/:INT",
"GET    /network/firewall/:TABLE",
"POST   /network/firewall/:TABLE",
"DELETE /network/firewall/:TABLE/:NAME",
"GET    /network/firewall/:TABLE/:NAME",
"PATCH  /network/firewall/:TABLE/:NAME",
"PUT    /network/firewall/:TABLE/:NAME",
"GET    /network/firewall/:TABLE/:NAME/rules",
"POST   /network/firewall/:TABLE/:NAME/rules",
"DELETE /network/firewall/:TABLE/:NAME/rules/:ID",
"GET    /network/firewall/:TABLE/:NAME/rules/:ID",
"PATCH  /network/firewall/:TABLE/:NAME/rules/:ID",
"PUT    /network/firewall/:TABLE/:NAME/rules/:ID",
"POST   /network/firewall/:TABLE/:NAME/rules/:ID/move",
"GET    /network/hosts",
```

```
"POST   /network/hosts",
"DELETE /network/hosts/#ADDR",
"GET    /network/hosts/#ADDR",
"PATCH  /network/hosts/#ADDR",
"PUT    /network/hosts/#ADDR",
"GET    /network/ipsec/certificates",
"DELETE /network/ipsec/certificates/#NAME",
"POST   /network/ipsec/certificates/download",
"GET    /network/ipsec/connections",
"POST   /network/ipsec/connections",
"DELETE /network/ipsec/connections/:NAME",
"GET    /network/ipsec/connections/:NAME",
"PATCH  /network/ipsec/connections/:NAME",
"PUT    /network/ipsec/connections/:NAME",
"GET    /network/settings",
"PATCH  /network/settings",
"PUT   " /network/settings",
GET    /network/snmp",
"POST   /network/snmp",
"DELETE /network/snmp/#ID",
"GET    /network/snmp/#ID",
"PATCH  /network/snmp/#ID",
"PUT    /network/snmp/#ID",
"GET    /network/snmp/system",
"PATCH  /network/snmp/system",
"PUT    /network/snmp/system",
"GET    /network/staticRoutes/:TABLE",
"POST   /network/staticRoutes/:TABLE",
"DELETE /network/staticRoutes/:TABLE/*ID",
"GET    /network/staticRoutes/:TABLE/*ID",
"PATCH  /network/staticRoutes/:TABLE/*ID",
"PUT    /network/staticRoutes/:TABLE/*ID",
"GET    /pluggableDevices",
"GET    /pluggableDevices/:NAME",
"POST   /pluggableDevices/:NAME/delete",
"POST   /pluggableDevices/:NAME/eject",
"POST   /pluggableDevices/:NAME/setConsole",
"GET    /power/login",
"PATCH  /power/login",
"PUT    /power/login",
"GET    /power/networkPdus",
"POST   /power/networkPdus",
"DELETE /power/networkPdus/#ADDR",
"GET    /power/networkPdus/#ADDR",
"PATCH  /power/networkPdus/#ADDR",
"PUT    /power/networkPdus/#ADDR",
"GET    /power/networkUps",
"POST   /power/networkUps",
 "DELETE /power/networkUps/#ADDR",
"GET    /power/networkUps/#ADDR",
"PATCH  /power/networkUps/#ADDR",
"PUT    /power/networkUps/#ADDR",
"GET    /power/outletGroups",
"POST   /power/outletGroups",
"DELETE /power/outletGroups/:NAME",
"GET    /power/outletGroups/:NAME",
"POST   /power/outletGroups/:NAME/cycle",
"POST   /power/outletGroups/:NAME/off",
```

```
"POST   /power/outletGroups/:NAME/on",
"GET    /power/outletGroups/:NAME/outlets",
"POST   /power/outletGroups/:NAME/outlets",
"DELETE /power/outletGroups/:NAME/outlets/:ID",
"GET    /power/pdus",
"GET    /power/pdus/:NAME",
"PATCH  /power/pdus/:NAME",
"PUT    /power/pdus/:NAME",
"GET    /power/pdus/:NAME/banks",
"GET    /power/pdus/:NAME/banks/:ID",
"PATCH  /power/pdus/:NAME/banks/:ID",
"PUT    /power/pdus/:NAME/banks/:ID",
"POST   /power/pdus/:NAME/banks/:ID/resetValues",
"POST   /power/pdus/:NAME/cycle",
"POST   /power/pdus/:NAME/factoryDefaults",
"POST   /power/pdus/:NAME/firmwareDownload",
"POST   /power/pdus/:NAME/firmwareInstall",
"POST   /power/pdus/:NAME/off",
"POST   /power/pdus/:NAME/on",
"GET    /power/pdus/:NAME/outlets",
"GET    /power/pdus/:NAME/outlets/:ID",
"PATCH  /power/pdus/:NAME/outlets/:ID",
"PUT    /power/pdus/:NAME/outlets/:ID",
"POST   /power/pdus/:NAME/outlets/:ID/cycle",
"POST   /power/pdus/:NAME/outlets/:ID/lock",
"POST   /power/pdus/:NAME/outlets/:ID/off",
"POST   /power/pdus/:NAME/outlets/:ID/on",
"POST   /power/pdus/:NAME/outlets/:ID/resetValues",
"POST   /power/pdus/:NAME/outlets/:ID/unlock",
"GET    /power/pdus/:NAME/phases",
"GET    /power/pdus/:NAME/phases/:ID",
"PATCH  /power/pdus/:NAME/phases/:ID",
"PUT    /power/pdus/:NAME/phases/:ID",
"POST   /power/pdus/:NAME/phases/:ID/resetValues",
"POST   /power/pdus/:NAME/reboot",
"POST   /power/pdus/:NAME/rename",
"POST   /power/pdus/:NAME/resetValues",
"GET    /power/pdus/:NAME/sensors",
"GET    /power/pdus/:NAME/sensors/:ID",
"PATCH  /power/pdus/:NAME/sensors/:ID",
"PUT    /power/pdus/:NAME/sensors/:ID",
"POST   /power/pdus/:NAME/sensors/:ID/resetValues",
"GET    /power/ups",
"GET    /power/ups/:NAME",
"PATCH  /power/ups/:NAME",
"PUT    /power/ups/:NAME",
"GET    /power/ups/:NAME/outletGroups",
"GET    /power/ups/:NAME/outletGroups/:ID",
"PATCH  /power/ups/:NAME/outletGroups/:ID",
"PUT    /power/ups/:NAME/outletGroups/:ID",
"POST   /power/ups/:NAME/outletGroups/:ID/cycle",
"POST   /power/ups/:NAME/outletGroups/:ID/off",
"POST   /power/ups/:NAME/outletGroups/:ID/on",
"POST   /power/ups/:NAME/outputCycle",
"POST   /power/ups/:NAME/outputOff",
"POST   /power/ups/:NAME/outputOn",
"POST   /power/ups/:NAME/rename",
"POST   /power/ups/:NAME/resetPowerStats",
```

```
"POST   /power/ups/:NAME/restoreName",
"POST   /power/ups/:NAME/silenceAlarm",
"POST   /power/ups/:NAME/testBattery",
"GET    /resources",
"GET    /security",
"PATCH  /security",
"PUT    /security",
"GET    /sensors/1Wire",
"DELETE /sensors/1Wire/:ADDR",
"GET    /sensors/1Wire/:ADDR",
"PATCH  /sensors/1Wire/:ADDR",
"PUT    /sensors/1Wire/:ADDR",
"POST   /sensors/1Wire/refresh",
"GET    /sensors/digitalIn",
"GET    /sensors/digitalIn/:ID",
"PATCH  /sensors/digitalIn/:ID",
"PUT    /sensors/digitalIn/:ID",
"GET    /sensors/internal",
"PATCH  /sensors/internal",
"PUT    /sensors/internal",
"GET    /sensors/pdu",
"POST   /sensors/pdu/#NAME/reset",
"GET    /serialPorts",
"GET    /serialPorts/:PORT",
"PATCH  /serialPorts/:PORT",
"PUT    /serialPorts/:PORT",
"GET    /serialPorts/:PORT/alerts",
"POST   /serialPorts/:PORT/alerts",
"DELETE /serialPorts/:PORT/alerts/:ID",
"POST   /serialPorts/:PORT/alerts/clear",
"POST   /serialPorts/:PORT/alerts/deleteAny",
"GET    /serialPorts/:PORT/power",
"POST   /serialPorts/:PORT/power",
"DELETE /serialPorts/:PORT/power/:ID",
"GET    /sessions",
"DELETE /sessions/:ID",
"GET    /sessions/:ID",
"POST   /sessions/login",
"POST   /sessions/logout",
"GET    /sessions/refresh",
"GET    /system/bootConfig",
"PATCH  /system/bootConfig",
"PUT    /system/bootConfig",
"POST   /system/certificate/apply",
"POST   /system/certificate/download",
"POST   /system/certificate/generate",
"POST   /system/config/restore",
"POST   /system/config/save",
"GET    /system/dateAndTime",
"PATCH  /system/dateAndTime",
"PUT    /system/dateAndTime",
"GET    /system/dateAndTime/timezone/custom",
"PATCH  /system/dateAndTime/timezone/custom",
"PUT    /system/dateAndTime/timezone/custom",
"GET    /system/dateAndTime/timezones",
"POST   /system/factoryDefault",
"POST   /system/firmware/download",
"GET    /system/firmware/downloaded",
```

```
            "POST   /system/firmware/install",
            "GET    /system/firmware/version",
            "GET    /system/general",
            "PATCH  /system/general",
            "PUT    /system/general",
            "GET    /system/info",
            "POST   /system/integrity/generate",
            "POST   /system/integrity/verify",
            "POST   /system/reboot",
            "POST   /system/shutdown",
            "GET    /system/usage/flash",
            "GET    /system/usage/memory",
            "GET    /users",
            "POST   /users",
            "DELETE /users/:NAME",
            "GET    /users/:NAME",
            "PATCH  /users/:NAME",
            "PUT    /users/:NAME",
            "POST   /users/:NAME/unlock",
            "GET    /users/passwordRules",
            "PATCH  /users/passwordRules",
            "PUT    /users/passwordRules"
        ]
    }
```

APPENDICES

Appendix A: cURL

The cURL command line utility is one method that can be used to communicate with the RESTful API.

Examples:

```
$ curl -H "Content-Type: application/json" -H "Accept:application/json"
http://10.20.30.40:8080/api/v1/sessions/login -d
 '{"username":"admin","password":"avocent"}'

{
 "token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE1MDQxMTU2NzUsImlkIjoiYWRtaW4iLCJvcmlnX2lhdC
I6MTUwNDExMjA3NSwic2lkIjo4fQ.UYGXje5It2hAJryruP3etUaabSh5pfiPP_sXXZF37og"
}

$ curl -H "Content-Type: application/json" -H "Accept:application/json" -H
"Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE1MDQxMTU2NzUsImlkIjoiYWRta
W4iLCJvcmlnX2lhdCI6MTUwNDExMjA3NSwic2lkIjo4fQ.UYGXje5It2hAJryruP3etUaabSh5
pfiPP_sXXZF37og" http://10.20.30.40:8080/api/v1/system/info
{
 "serialNumber": "0012345678",
 "type": "ACS8048 with single power supply",
 "bootcode": "1.17",
 "firmware": "1.3.77.2909+551+28+11",
 "bootedFrom": "hardware",
 "powerSupply1": "on",
 "cpu": "ARMv7 Processor rev 0 (v7l)",
 "cores": 2
}
```

Appendix B: Helper Script

A bash shell helper script is provided in the ACS8000 root filesystem under the /usr/share/restapi/restapi-helper.sh path.

This helper script modifies the environment of the running shell to add GET/PUT/POST/PATCH/DELETE shell functions. These functions provide a simple command line interface to demonstrate the API. The shell functions utilize the 'curl' program, which must be installed on the system.

From a bash shell, the script must be "sourced" in order to make the necessary changes to the shell environment, which are shown as SETUP parameters and functions.

When sourcing the script, the last parameter is the IPv4 address of the ACS8000. In the following example, 10.20.30.40 is the console system IP address.

Example

```
$ . restapi-helper.sh 10.20.30.40
Token for Basic Authentication saved in /home/root/.acsrestapi/Basic-token-127.0.0.1
SETUP parameters:

ACSHOST          127.0.0.1
ACSPROTOCOL   http
ACSPORT          8180
ACSTOKENDIR   /home/root/.acsrestapi
ACSTOKENTYP   Basic (Basic Authentication)
ACSURL      /api/v1
ACSDEBUG     no

Change your ACSHOST parameter to the ACS IP Address.  For example:
ACSHOST=192.168.161.10

GET /resource
                Example:
                    GET /serialPorts/2

PUT /resource '{...json parameters...}'
                Example:
                     PUT /serialPorts/2 '{"physical":{"speed":38400}}'

PATCH /resource '{...json parameters...}'
                Example:
                    PATCH /security '{"idleTimeout":0}'

POST /resource '{...json parameters...}'
                Example:
                     POST /sessions/login '
{"username":"admin","password":"avocent"}'

DELETE /resource
```

These functions automatically prepend '/acs/v1' to the /resource.

After running the helper script, the current shell environment can execute POST, GET and the other commands mentioned previously.

Example

```
$ GET /system/info
{
 "serialNumber": "001234567",
 "type": "ACS8048 with single power supply",
 "bootcode": "1.18",
 "firmware": "1.3.77.2909+551+28+11",
 "firmwareDate": "Sep 20 2017 - 08:03:39",
 "bootedFrom": "hardware",
 "powerSupply1": "on",
 "cpu": "ARMv7 Processor rev 0 (v7l)",
 "cores": 2
```

Appendix C: Certificate Verification

When using HTTPS, in order to avoid certificate verification warnings, the client should be set to ignore SSL certificate verification.

With the Python requests package, this is done by adding verify=False to the requests command.

Example

```
>>> requests.get(URL, verify=False)
```

With cURL the -k option is used to disable certificate verification.

Example

```
$ curl -k -H "Content-Type: application/json" -H "Accept:application/json"
https://10.20.30.40:48048/api/v1/sessions/login -d
'{"username":"admin","password":"avocent"}'
```

The restapi-helper script automatically includes the -k option on the underlying curl commands when using https.

Vertiv.com | Vertiv Headquarters, 1050 Dearborn Drive, Columbus, OH, 43085, USA

590-1814-501D