



Configuring HTTP API

You can run an application either locally or remotely on the Cisco Edge 300 series switch to manage the switch by using HTTP API. Management of the switch consists of configuring the switch, monitoring the status, and installing and upgrading software.

The configuration of HTTP API for the Cisco Edge 300 series switch is supported from Release 1.5. This chapter explains each HTTP API including requests, replies, parameter restrictions, and error codes.



Note

Curl is used as a sample tool to request APIs. 10.140.44.134 is used as a sample IP address of the Cisco Edge 300 series switch, and cisco123! is used as a sample admin password.

To configure the Cisco Edge 300 series switch by using HTTP API, see the following sections:

- [System API, page 6-2](#)
- [Ethernet API, page 6-21](#)
- [Issue a Command, page 6-47](#)
- [Image Version Information, page 6-47](#)
- [AP Information, page 6-48](#)
- [Wifi Client Information, page 6-56](#)
- [RS232 Configuration, page 6-67](#)
- [Upgrade, page 6-67](#)
- [Error Codes, page 6-68](#)

System API

Use the commands in this section to configure the system API.



Note

Curl is used as a sample tool to request APIs. 10.140.44.134 is used as a sample IP address of the Cisco Edge 300 series switch, and cisco123! is used as a sample admin password.

Set Hostname

Example: set hostname to cisco

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
'{"hostname" : "cisco"}' https://10.140.44.134/api/1.0/sys/hostname
```

Reply

```
{"success": "true", "updatedAt": "2012-11-06 17:39:47"}
```

Parameter restrictions

The length of hostname should be 1 to 64 and the valid parameter set is {a-zA-Z0-9-}, or 004 error is reported.

Get Hostname

Example: get hostname

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/sys/hostname
```

Reply

```
{"hostname": "cisco", "success": "true", "getAt": "2012-11-06 17:44:37"}
```

Set Log Size

Example: set log size to 20 MB

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
'{"size": "20"}' https://10.140.44.134/api/1.0/sys/log/size
```

Reply

```
{"success": "true", "updatedAt": "2012-11-06 17:39:47"}
```

Parameter restrictions

The log size parameter should be an integer, in the range from 1 to 100, or 004 error is reported.

Get Log Size

Example: get log size

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/sys/log/size
```

Reply

```
{"size": "20", "success": "true", "getAt": "2011-04-21 04:33:55"}
```

Delete Logs

Example: delete all logs

Request

```
curl -k -X DELETE -H 'password: cisco123!' https://10.140.44.134/api/1.0/sys/log
```

Reply

```
{"success": "true", "updatedAt": "2012-12-27 08:24:08"}
```

Set Account

Example: change the password of admin account from cisco123! to cisco123

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"account": "cisco123"}' https://10.140.44.134/api/1.0/sys/account
```

Reply

```
{"success": "true", "updatedAt": "2012-11-06 17:48:06"}
```

Parameter restrictions

The password cannot be empty and the valid parameter set is {a-zA-Z0-9~!@#\$\$%^&*+=-_}, otherwise 004 error is reported. The password should follow the busy box linux password requirement, otherwise 005 error is reported.

Get Account

003 error is reported if violated.

Set LoginGui

Example: set loginGui to enable

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"loginGui": "enable"}' https://10.140.44.134/api/1.0/sys/loginGui
```

Reply

```
{"success": "true", "updatedAt": "2012-11-06 17:39:47"}
```

Parameter restrictions

Valid strings for loginGui: enable and disable. Otherwise, 004 error is reported.

Get LoginGui

Example: get login gui

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/sys/loginGui
```

Reply

```
{"loginGui": "enable", "success": "true", "getAt": "2012-11-06 19:17:41"}
```

Set Resolution

Example: set resolution to 1080p

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"resolution": "4"}' https://10.140.44.134/api/1.0/sys/resolution
```

Reply

```
{"success": "true", "updatedAt": "2012-11-06 17:39:47"}
```

Parameter restrictions

The valid parameter set is number 1–9. Otherwise, 004 error is reported.

Get Resolution

Example: get resolution

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/sys/resolution
```

Reply

```
{"resolution": "9", "success": "true", "getAt": "2012-12-14 08:55:27"}
```

Get Hdmi Info

Example: get hdmi information

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/sys/hdmi
```

Reply

```
{"hdmi": " - Manufacturer ID: 0x294d\n - Product code : 0x9135\n - Sink name : HDMI TV\n - Sink size (WxH) : 80 x 45\nCurrent working mode:\n 1920x1080p@59.94\nSupported modes:\n - 720x480p60 \n - 1024x768p60 \n - 1920x1080p60 \n - 1280x720p60 \n - 1920x1080p50 \n - 1280x720p50 \n - 1280x960p85 \n - 720x480p59.94 \n - 1024x768p59.94 \n - 1920x1080p59.94 \n - 1280x720p59.94 \n\n", "success": "true", "getAt": "2012-12-14 08:41:33"}
```

Set Bluetooth

Example: set bluetooth to on

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d  
'{"bluetooth" : "on"}' https://10.140.44.134/api/1.0/sys/bluetooth
```

Reply

```
{"success": "true", "updatedAt": "2012-11-06 17:39:47"}
```

Parameter restrictions

The valid parameter is on/off. Otherwise, 004 error is reported.

Get Bluetooth

Example: get bluetooth

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/sys/bluetooth
```

Reply

```
{"bluetooth": "on", "success": "true", "getAt": "2012-11-06 19:43:08"}
```

Set Language

Example: set language to 1

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d  
'{"language" : "1"}' https://10.140.44.134/api/1.0/sys/language
```

Reply

```
{"success": "true", "updatedAt": "2012-11-06 19:49:21"}
```

Parameter restrictions

The valid parameter set is number 1-9. Otherwise, 004 error is reported.

Get Language

Example: get language

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/sys/language
```

Reply

```
{"language": "1", "success": "true", "getAt": "2012-11-06 19:49:24"}
```

Set Locale

Example: set locale to ì8î

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"locale": "8"}' https://10.140.44.134/api/1.0/sys/locale
```

Reply

```
{"success": "true", "updatedAt": "2012-11-06 19:49:21"}
```

Parameter restrictions

The valid parameter set is number 0-26. Otherwise, 004 error is reported.

Get Locale

Example: get locale

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/sys/locale
```

Reply

```
{"locale": "9", "success": "true", "getAt": "2012-11-06 20:13:31"}
```

Set ntpServer

Example: set NTP server to 202.120.2.101

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"ntpServer": "ntp.sjtu.edu.cn"}' https://10.140.44.134/api/1.0/sys/ntpServer
```

Reply

```
{"success": "true", "updatedAt": "2012-11-06 19:49:21"}
```

Parameter restrictions

The parameter should be a valid IPv4 address or a valid domain name.

Get ntpServer

Example: get ntpServer

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/sys/ntpServer
```

Reply

```
{"ntpServer": "ntp.sjtu.edu.cn", "success": "true", "getAt": "2012-12-27 08:04:34"}
```

Set Time

Example: set time

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"time" : "2012-11-06 17:20:20"}' https://10.140.44.134/api/1.0/sys/time
```

Reply

```
{"success": "true", "updatedAt": "2012-11-06 19:49:21"}
```

Parameter restrictions

The parameter should be in YYYY-MM-DD HH:MM:SS format. Otherwise, 004 error is reported.

Get Time

Example: get time

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/sys/time
```

Reply

```
{"time": "2012-11-07 15:22:15", "success": "true", "getAt": "2012-11-07 06:22:15"}
```

Set CPU

N/A. 003 error is reported if it is requested.

Get CPU

Example: get CPU

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/sys/cpu
```

Reply

```
{ "cpu": "CPU: 9% usr 0% sys 0% nic 90% idle 0% io 0% irq 0%
sirq", "success": "true", "getAt": "2012-11-07 06:35:06" }
```

Set Memory

N/A. 003 error is reported if it is requested.

Get Memory

Example: get memory

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/sys/memory
```

Reply

```
{ "memory": "Mem: 378112K used, 1310780K free, 0K shrd, 57188K buff, 155920K
cached", "success": "true", "getAt": "2012-11-07 08:08:07" }
```

Set Process

N/A. 003 error is reported if it is requested.

Get Process

Example: get process

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/sys/proc
```

Reply

```
{ "proc": "UID          PID  PPID  C  STIME TTY          TIME CMD\nroot          1    0  0  10:57 ?          00:00:00
10:57 ? 00:00:01 init          \nroot          2    0  0  10:57 ?          00:00:00
[kthreadd]\nroot 3    2  0  10:57 ?          00:00:00 [migration\0]\nroot          4    2
0  10:57 ? 00:00:03 [ksoftirqd\0]\nroot          5    2  0  10:57 ?          00:00:00
[events\0]\nroot 6    2  0  10:57 ?          00:00:00 [khelper]\nroot          9    2  0
10:57 ?          00:00:00 [kstop\0]\nroot         134    2  0  10:57 ?          00:00:00
[kblockd\0]\nroot         136    2  0  10:57 ?          00:00:00 [kacpid]\nroot         137
2  0  10:57 ?          00:00:00 [kacpi_notify]\nroot         216    2  0  10:57 ?
00:00:00 [ata\0]\nroot         217    2  0  10:57 ?          00:00:00 [ata_aux]\nroot
218    2  0  10:57 ?          00:00:00 [ksuspend_usb]\nroot         224    2  0  10:57 ?
00:00:00 [khubd]\nroot         227    2  0  10:57 ?          00:00:00 [kseriod]\nroot         229
2  0  10:57 ?          00:00:00 [kgameportd]\nroot         232    2  0  10:57 ?          00:00:00
[kmmcd]\nroot         283    2  0  10:57 ?          00:00:00 [pdflush]\nroot         284    2
0  10:57 ?          00:00:00 [pdflush]\nroot         285    2  0  10:57 ?          00:00:00
[kswapd0]\nroot         326    2  0  10:57 ?          00:00:00 [aio\0]\nroot         337    2
0  10:57 ?          00:00:00 [nfsiod]\nroot         342    2  0  10:57 ?          00:00:00
[cifsoplockd]\nroot         343    2  0  10:57 ? 00:00:00 [cifsnotifyd]\nroot         527
2  0  10:57 ?          00:00:00 [scsi_ah_0]\nroot         529    2  0  10:57 ?          00:00:00
[scsi_ah_1]\nroot         533    2  0  10:57 ? 00:00:00 [mtddblockd]\nroot         579    2  0
```



```

10:57 ?          00:00:00 [kpsmoused]\nroot 588      2 0 10:57 ?          00:00:00
[hid_compat]\nroot      612      2 1 10:57 ? 00:06:35 [Glob_Spectra]\nroot      618
2 0 10:57 ?          00:02:18 [nandflush]\nroot 628      2 0 10:57 ?          00:00:00
[krfcomm]\nroot        631      2 0 10:57 ?          00:00:00 [rpciod\0]\nroot      664
2 0 10:57 ?          00:00:00 [scsi_ah_2]\nroot      665      2 0 10:57 ?          00:00:00
[usb-storage]\nroot    672      2 0 10:57 ?          00:00:06 [kjournald]\nroot      935
1 0 10:57 ?          00:00:00 udevd -d\nroot      1750     2 0 10:57 ?          00:00:00
[SEC int]\nroot        1854     2 0 10:57 ?          00:00:35 [Clock_ISR]\nroot      1869
1 0 10:57 ?          00:00:00 \bin\cdp eth0 start\nroot 2142     1 0 10:57 ?
00:03:14 \bin\gdl_server blender_config 1\nroot      2211     2 0 10:57 ?
00:00:00 [VidDec_hal_pars]\nroot      2212     2 0 10:57 ?          00:00:00
[VidDec_hal_deco]\nroot 2218     2 0 10:57 ?          00:00:00 [VidPProc_ISR]\nroot
2219     2 0 10:57 ?          00:00:00 [VidPProc_IO]\nroot      2223     2 0 10:57 ?
00:00:24 [VidRend_IO]\nroot      2224     2 0 10:57 ?          00:00:00 [VidRend_IO]\nroot
2232     2 0 10:57 ?          00:01:18 [Audio_Rend_ISR]\nroot      2233     2 0 10:57 ?
00:00:27 [Audio_Timing]\nroot      2234     2 0 10:57 ?          00:00:08
[Audio_Pipe_Mgr]\nroot 2235     2 0 10:57 ?          00:00:00 [Audio_DSP0_ISR]\nroot
2236     2 0 10:57 ? 00:00:00 [Audio_DSP1_ISR]\nroot      2426     1 0 10:58 ?
00:00:00 \bin\konfd\nroot 2899     1 0 10:58 ?          00:00:00 php-fpm: master process
(\usr\local\cisco\php\etc\php-fpm.conf)\nroot      2900     2899 0 10:58 ?
00:00:02 php-fpm: pool root \nroot      2901     2899 0 10:58 ?          00:00:02 php-fpm:
pool root \nroot      2907     1 0 10:58 ?          00:00:00 nginx: master process
\usr\local\cisco\nginx\sbin\nginx\nroot      2908     2907 0 10:58 ?          00:00:14
nginx: worker process \nroot      2917     1 0 10:58 ?          00:00:00
\usr\local\cisco\sbin\cupsd -C \usr\local\cisco\etc\cupsd\cupsd.conf\nroot
2920     1 0 10:58 ?          00:00:00 \usr\local\cisco\sbin\xinetd\nroot      2930
1 0 10:58 ?          00:00:19 \usr\local\cisco\sbin\snmpd\nroot      2934     1 0
10:58 ?          00:00:00 \sbin\sshd\nroot 2941     1 0 10:58 ?          00:00:16
\usr\local\cisco\sbin\cron\nroot      2943     1 0 10:58 ?          00:00:00
\usr\local\cisco\bin\mosaic_server 0 5050 10000\nroot      2944     1 0 10:58 ?
00:00:00 \usr\local\cisco\bin\mosaic_server 1 5052 12000\nroot 2949     1 0 10:58 ?
00:00:00 audio_setup_outputs\nroot      2958     2 0 10:58 ? 00:00:00 [Audio_Input]\nroot
2959     2949 0 10:58 ?          00:00:00 [sh] <defunct>\nroot 2964     2 0 10:58 ?
00:00:01 [kjournald]\nroot      2966     1 0 10:58 ? 00:00:01
\usr\local\cisco\sbin\rsyslogd -c5\n1000      2977     1 0 10:58 ? 00:00:08
dbus-daemon --system\n1001      2979     1 0 10:58 ?          00:00:00 hald
--daemon=yes\nroot      2980     2979 0 10:58 ?          00:00:00 hald-runner\nroot      2994
2980 0 10:58 ?          00:00:00 hald-addon-storage: polling \dev\ciscoapps (every 2
sec)\nroot 2997     1 0 10:58 ?          00:00:01 bluetoothd\nroot      3004     1 0 10:58
? 00:00:00 Agent_3g\nroot      3018     1 0 10:58 ?          00:00:00 wan_detector\nroot
3046     1 0 10:58 ?          00:00:05 slim\nroot      3089     3046 0 10:58 tty2      00:02:06
\usr\bin\X -auth \var\run\slim.auth\nroot      3134     1 0 10:58 ?          00:00:00
dhclient br0\nroot      3135     1 0 10:58 ?          00:00:19 \bin\sh
\scripts\status_check.sh\nroot 3209     1 0 10:59 ?          00:00:00 \sbin\smi\nroot
3243     2 0 10:59 ? 00:00:00 [kjournald]\nstudent      3380     3046 0 10:59 ?
00:00:00 \bin\sh \etc\xinitrc xfce4\nstudent      3392     3380 0 10:59 ?          00:00:00
\bin\sh \scripts\startxfce4\nstudent 3004     3392 0 10:59 ?          00:00:01
xfce4-session\nroot      3491     1 0 10:59 ? 00:00:00 init \nstudent      3502     1
0 10:59 ?          00:00:00 dbus-launch --autolaunch 4b8ead68809b704d85084ca50000005c
--binary-syntax --close-stderr\nstudent      3504     1 0 10:59 ? 00:00:00
\usr\local\cisco\bin\dbus-daemon --fork --print-pid 5 --print-address 7
--session\nstudent      3506     1 0 10:59 ?          00:00:00
\usr\local\cisco\lib\xfce4\xfconf\xfconfd\nstudent      3554     1 0 10:59 ?
00:00:21 xfwm4\nstudent      3555     1 0 10:59 ?          00:00:00 xfsettingsd\nstudent      3565
1 0 10:59 ?          00:00:35 xfce4-panel\nstudent      3576     1 0 10:59 ?          00:00:00
Thunar\nstudent      3578     1 0 10:59 ?          00:00:03 xfdesktop\nstudent      3597
3004 0 10:59 ?          00:00:00 3G_Dongle\nstudent      3598     3004 0 10:59 ?          00:00:13
BlueToothUI\nstudent      3602     3565 0 10:59 ?          00:00:24
\usr\local\cisco\lib\xfce4\panel\wrapper
\usr\local\cisco\lib\xfce4\panel\plugins\libsystray.so 6 16777251 systray
Notification Area Area where notification icons appear \nstudent      3621     3004 0 10:59 ?
00:00:03 wifi_status hide\nstudent      3632     3004 0 10:59 ?          00:00:12 wired_status
hide > \dev\null\nstudent      3635     1 0 10:59 ?          00:00:00
xfce4-settings-helper\nstudent      3679     1 0 10:59 ? 00:00:00

```

```

/usr/local/cisco/libexec/gvfsd/nstudent 3692 1 0 10:59 ? 00:00:00
/usr/local/cisco/libexec/gconfd-2/nstudent 3709 1 0 10:59 ? 00:00:00
/usr/local/cisco/libexec/gvfs-hal-volume-monitor/nstudent 3717 1 0 10:59 ?
00:00:00 /usr/local/cisco/libexec/gvfs-fuse-daemon
/apps/localconfig/student/.gvfs/nstudent 3730 1 0 10:59 ? 00:00:00
/usr/local/cisco/libexec/gvfsd-trash --spawner :1.11
/org/gtk/gvfs/exec_spaw/0/nstudent 0044 1 0 11:00 ? 00:00:00
/usr/local/cisco/lib/scim-1.0/scim-launcher -d -c simple -e all -f socket
--no-stay/nstudent 4068 1 0 11:00 ? 00:00:00
/usr/local/cisco/lib/scim-1.0/scim-helper-manager/nstudent 4069 1 0 11:00 ?
00:00:00 /usr/local/cisco/lib/scim-1.0/scim-panel-gtk --display :0.0 -c socket -d
--no-stay/nstudent 4071 1 0 11:00 ? 00:00:00
/usr/local/cisco/lib/scim-1.0/scim-launcher -d -c socket -e socket -f x11/nroot 6319
2980 0 16:40 ? 00:00:00 hald-addon-input: Listening on /dev/input/event0
/dev/input/event1 /dev/input/event2/nroot 6510 2 0 16:41 ? 00:00:00
[Audio_Recovery]/nroot 6574 2 0 16:41 ? 00:00:00 [scsi_ah_4]/nroot
6575 2 0 16:41 ? 00:00:00 [usb-storage]/nroot 6680 2980 0 16:41 ?
00:00:00 hald-addon-storage: polling /dev/sdb (every 2 sec)/nroot 7088 2 0
16:41 ? 00:00:00 [kjournald]/nstudent 7938 1 0 11:04 ? 00:00:08
/usr/local/cisco/bin/terminal/nstudent 7997 7938 0 11:04 ? 00:00:00
gnome-pty-helper/nstudent 7998 7938 0 11:04 pts/0 00:00:00 bash/nroot 10418
7998 0 11:40 pts/0 00:00:00 -bash/nroot 15412 3135 0 17:23 ? 00:00:00
sleep 1/nroot 15432 2901 0 17:23 ? 00:00:00 sh -c cd
'/usr/local/cisco/nginx/html/api/1.0/sys' ; /usr/local/cisco/bin/ps -ef
2>&1/nroot 15433 15432 0 17:23 ? 00:00:00 /usr/local/cisco/bin/ps
-ef\n", "success": "true", "getAt": "2012-11-07 08:23:50"}

```

Set Storage

N/A. 003 error is reported if it is requested.

Get Storage

Example: get storage

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/sys/storage
```

Reply

```

{"storage": "Filesystem", "Size", "Used", "Available", "Use%", "Mounted"}
on/n/dev/root 1.6G 1.2G 343.1M 78% /\ntmpfs 512.0M 428.0K
511.6M 0% /tmpntmpfs 4.0K 0 4.0K 0% /media\ntmpfs
20.0M 228.0K 19.8M 1% /var/n/dev/ciscoapps 1.8G 527.4M 1.2G
30% /apps/n/dev/Glob_Spectraal 96.6M 77.9M 13.8M 85%
/tmp/smi_spectraal\ntmpfs 32.0M 17.8M 14.2M 56%
/tmp/firefox_cached\n64.104.163.32:/var/www/html/api 25.6G 6.2G 18.1G 26%
/usr/local/cisco/nginx/html/api/n/dev/sdb1 7.3G 308.1M
6.7G 4% /media/sdb1\n", "success": "true", "getAt": "2012-11-07 08:35:55"}

```

Set Model

N/A. 003 error is reported if it is requested.

Get Model

Example: get model

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/sys/model
```

Reply

```
{"model": "CS-E300-AP-K9", "success": "true", "getAt": "2012-11-07 08:59:00"}
```

Set IP

Example: set IP address to 64.104.163.55 and netmask to 255.255.255.128

Request

```
curl -m 5 -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d  
'{"type" : "static", "ipv4": "64.104.163.55", "netmask": "255.255.255.128"}'  
https://64.104.163.47/api/1.0/sys/ip
```

Reply

```
{"success": "true", "updatedAt": "2012-11-06 19:49:21"}
```

Parameter restrictions

The parameter type is static or DHCP. If static is specified in the type field, ipv4 and netmask must also be specified. IPv4 must be a valid IPv4 address. Netmask must be one of the following strings:

```
"255.0.0.0",  
"255.128.0.0",  
"255.192.0.0",  
"255.224.0.0",  
"255.240.0.0",  
"255.248.0.0",  
"255.252.0.0",  
"255.254.0.0",  
"255.255.0.0",  
"255.255.128.0",  
"255.255.192.0",  
"255.255.224.0",  
"255.255.240.0",  
"255.255.248.0",  
"255.255.252.0",  
"255.255.254.0",  
"255.255.255.0",  
"255.255.255.128",
```

```
"255.255.255.192",
"255.255.255.224",
"255.255.255.240",
"255.255.255.248",
"255.255.255.252",
"255.255.255.254",
"255.255.255.255"
```

**Note**

Since the IP address is changed after the execution, “-m 5” must be specified to make sure that the command will not be appending forever.

Get Ip Address

Example: get ip address

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/sys/ip
```

Reply

```
{"ip": "10.140.44.134", "success": "true", "getAt": "2012-11-08 08:48:53"}
```

Set Gateway

Example: set gateway

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
'{"gateway": "64.104.163.1"}' https://10.140.44.134/api/1.0/sys/gateway
```

Reply

```
{"success": "true", "updatedAt": "2011-04-21 05:46:07"}
```

Parameter restrictions

The gateway should be a valid IP address.

Get Gateway

Example: get gateway

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/sys/gateway
```

Reply

```
{"gateway": "64.104.163.1", "success": "true", "getAt": "2012-11-08 08:49:59"}
```

Set DNS

Example: set DNS to 8.8.8.8

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
 '{"dns": "8.8.8.8"}' https://10.140.44.134/api/1.0/sys/dns
```

Reply

```
{"success": "true", "updatedAt": "2012-12-14 07:50:00"}
```

Parameter restrictions

DNS should be a valid IP address.

Get DNS

Example: get DNS

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/sys/dns
```

Reply

```
{"dns": "64.104.123.144 171.70.168.183 ", "success": "true", "getAt": "2012-11-08 08:50:37"}
```

Set Wifi Mode

Example: set WiFi mode

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
 '{"wifiMode": "client"}' https://10.140.44.134/api/1.0/sys/wifiMode
```

Reply

```
{"success": "true", "updatedAt": "2012-12-14 08:11:16"}
```

Parameter restrictions

The WiFi mode should be AP or client.

Get Wifi Mode

Example: get WiFi mode

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/sys/wifiMode
```

Reply

```
{"wifiMode": "ap", "success": "true", "getAt": "2012-12-14 08:09:59"}
```

Set a Proxy of Chrome Browser

Example: set a proxy of chrome browser

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"proxy" :
{"host": "10.10.10.10", "scheme": "http", "port": "10", "username": "cisco", "password": "cisco"}}'
```

Reply

```
{"success": "true", "updatedAt": "2012-12-14 08:11:16"}
```

Parameter restrictions

host: IP address/hostname. If not specified, the proxy setting will be set to none.

scheme: http/https

port: Should be an integer within the range 0 and 65535.

username and password: Optional. Specify the account information of the proxy.



Note You must specify both username and password. If only the username is specified, both the username and password will be deleted.

Get the Proxy of Chrome Browser

Example: get the proxy of chrome browser

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.28.47/api/1.0/sys/proxy
```

Reply

```
{"proxy": "http://cisco:cisco@10.10.10.10:10", "success": "true", "getAt": "2011-04-21
04:04:03"}
```

Set System Information

Example: set hostname to ce300, NTP server to 202.120.2.101, and log size to 30 MB.

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
'{"hostname": "ce300", "ntpServer" : "202.120.2.101", "log_size": 30}'
```

Reply

```
{"success": "true", "updatedAt": "2012-11-06 19:49:21"}
```

Parameter restrictions

The parameter should be a valid IPv4 address.

Get System Information

Example: get all system information

Request:

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/sys
```

Reply

```
curl -k -X GET -H 'password: cisco123!'
https://10.140.44.134/api/1.0/sys{"hostname":"intel_ce_linux", "size":"30",
"ip":"10.140.44.134", "gateway":"10.140.28.1", "dns":"64.104.123.144
171.70.168.183", "language":"1", "model":"CS-E300-AP-K9", "locale":"8", "time":"2012-12-14
16:13:03", "ntpServer":"10.81.254.202", "loginGui":"disable", "resolution":"9", "wifiMode":"cl
ient", "bluetooth":"on", "memory":"Mem: 380776K used, 1308116K free, 0K shrd, 27764K buff,
170748K cached", "storage":"Filesystem
Size Used Available Use% Mounted
on\n\dev\root 1.6G 1.2G 300.9M 81% \\/ntmpfs
512.0M 172.0K 511.8M 0% \\/tmp\ntmpfs 4.0K 0 4.0K
0% \\/media\ntmpfs 20.0M 88.0K 19.9M 0%
\/var\n\dev\ciscoapps 1.8G 539.6M 1.2G 30% \\/apps\n\dev\Glob_Spectraal
96.6M 84.4M 7.2M 92% \\/tmp\smi_spectraal\ntmpfs 32.0M
17.8M 14.2M 56% \\/tmp\firefox_cached\n10.140.28.35:\\/var\www\html\api
25.6G 7.2G 17.1G 30% \\/usr\local\cisco\nginx\html\api\n", "cpu":"CPU: 0% usr
72% sys 0% nic 27% idle 0% io 0% irq 0% sirq", "proc":"UID PID PPID C
STIME TTY TIME CMD\nroot 1 0 0 15:35 ? 00:00:01 init
\nroot 2 0 0 15:35 ? 00:00:00 [kthreadd]\nroot 3 2 0 15:35 ?
00:00:00 [migration\0]\nroot 4 2 0 15:35 ? 00:00:00 [ksoftirqd\0]\nroot
5 2 0 15:35 ? 00:00:00 [events\0]\nroot 6 2 0 15:35 ? 00:00:00
[khelper]\nroot 9 2 0 15:35 ? 00:00:00 [kstop\0]\nroot 134 2 0
15:35 ? 00:00:00 [kblockd\0]\nroot 136 2 0 15:35 ? 00:00:00
[kacpid]\nroot 137 2 0 15:35 ? 00:00:00 [kacpi_notify]\nroot 216
2 0 15:35 ? 00:00:00 [ata\0]\nroot 217 2 0 15:35 ? 00:00:00
[ata_aux]\nroot 218 2 0 15:35 ? 00:00:00 [ksuspend_usbd]\nroot 224 2
0 15:35 ? 00:00:00 [khubd]\nroot 227 2 0 15:35 ? 00:00:00
[kseriod]\nroot 229 2 0 15:35 ? 00:00:00 [kgameportd]\nroot 232 2 0
15:35 ? 00:00:00 [kmmcd]\nroot 283 2 0 15:35 ? 00:00:00
[pdflush]\nroot 284 2 0 15:35 ? 00:00:00 [pdflush]\nroot 285 2
0 15:35 ? 00:00:00 [kswapd0]\nroot 326 2 0 15:35 ? 00:00:00
[aio\0]\nroot 337 2 0 15:35 ? 00:00:00 [nfsiod]\nroot 342 2 0
15:35 ? 00:00:00 [cifsoplockd]\nroot 343 2 0 15:35 ? 00:00:00
[cifsdnotifyd]\nroot 527 2 0 15:35 ? 00:00:00 [scsi_ah_0]\nroot 529
2 0 15:35 ? 00:00:00 [scsi_ah_1]\nroot 533 2 0 15:35 ? 00:00:00
[mtddblockd]\nroot 579 2 0 15:35 ? 00:00:00 [kpsmoused]\nroot 588 2
0 15:35 ? 00:00:00 [hid_compat]\nroot 612 2 6 15:35 ? 00:02:22
[Glob_Spectra]\nroot 618 2 0 15:35 ? 00:00:11 [nandflush]\nroot 628
2 0 15:35 ? 00:00:00 [krfcomm]\nroot 631 2 0 15:35 ? 00:00:00
[rpciod\0]\nroot 688 2 0 15:35 ? 00:00:00 [scsi_ah_2]\nroot 689
2 0 15:35 ? 00:00:00 [usb-storage]\nroot 704 2 0 15:35 ? 00:00:00
[kjournal]\nroot 917 1 0 15:35 ? 00:00:00 udevd -d\nroot 2102
2 0 15:35 ? 00:00:00 [SEC_int]\nroot 2134 1 0 15:35 ? 00:00:00
\/bin\cdp eth0 start\nroot 2138 2 0 15:35 ? 00:00:04 [Clock_ISR]\nroot
2166 1 0 15:35 ? 00:00:21 \\/bin\gdl_server blender_config 1\nroot 2220
2 0 15:35 ? 00:00:00 [VidDec_hal_pars]\nroot 2221 2 0 15:35 ? 00:00:00
[VidDec_hal_deco]\nroot 2227 2 0 15:35 ? 00:00:00 [VidPProc_ISR]\nroot
2228 2 0 15:35 ? 00:00:00 [VidPProc_IO]\nroot 2232 2 0 15:35 ?
00:00:02 [VidRend_IO]\nroot 2233 2 0 15:35 ? 00:00:00 [VidRend_IO]\nroot
2237 2 0 15:35 ? 00:00:08 [Audio_Rend_ISR]\nroot 2238 2 0 15:35 ?
00:00:03 [Audio_Timing]\nroot 2239 2 0 15:35 ? 00:00:00
[Audio_Pipe_Mgr]\nroot 2240 2 0 15:35 ? 00:00:00 [Audio_DSP0_ISR]\nroot
2241 2 0 15:35 ? 00:00:00 [Audio_DSP1_ISR]\nroot 2446 1 0 15:36 ?
00:00:00 \\/usr\local\cisco\sbin\snmpd -p \\/var\run\snmpd.pid\nroot 2448 1
0 15:36 ? 00:00:00 \\/bin\konfd\nroot 2583 1 0 15:36 ? 00:00:00
```

```

php-fpm: master process (\usr\local\cisco\php\etc\php-fpm.conf)
\nroot      2584 2583 0 15:36 ? 00:00:00 php-fpm: pool root \nroot      2585 2583 0
15:36 ?      00:00:00 php-fpm: pool root \nroot      2592 1 0 15:36 ?
00:00:00 nginx: master process \usr\local\cisco\nginx\sbin\nginx\nroot      2602
1 0 15:36 ? 00:00:00 \usr\local\cisco\etc\cupsd\cupsd.conf\nroot      2606 1 0 15:36 ?
00:00:00 \usr\local\cisco\sbin\xinetd\nroot      2611 1 0 15:36 ?
00:00:00 \sbin\sshd\nroot      2620 1 0 15:36 ? 00:00:11
\usr\local\cisco\sbin\cron\nroot      2623 1 0 15:36 ?      00:00:00
\usr\local\cisco\bin\mosaic_server 0 5050 10000\nroot      2624 1 0 15:36 ?
00:00:00 \usr\local\cisco\bin\mosaic_server 1 5052 12000\nroot      2632 1 0
15:36 ? 00:00:00 audio_setup_outputs\nroot      2641 2 0 15:36 ?      00:00:00
[Audio_Input]\nroot      2643 2632 0 15:36 ?      00:00:00 [sh] <defunct>\nroot
2645 2 0 15:36 ?      00:00:00 [Audio_Recovery]\nroot      2647 2 0 15:36 ?
00:00:00 [kjournald]\nroot      2649 1 0 15:36 ?      00:00:00
\usr\local\cisco\sbin\rsyslogd -c5\n1000      2660 1 0 15:36 ?      00:00:00
dbus-daemon --system\n1001      2662 1 0 15:36 ?      00:00:00 hald
--daemon=yes\nroot      2663 2662 0 15:36 ?      00:00:00 hald-runner\nroot      2668
2663 0 15:36 ?      00:00:00 hald-addon-input: Listening on \dev\input\event2
\dev\input\event1 \dev\input\event0\nroot      2677 2663 0 15:36 ? 00:00:00
hald-addon-storage: polling \dev\ciscoapps (every 2 sec)\nroot      2680 1 0 15:36 ?
00:00:00 bluetoothd\nroot      2687 1 0 15:36 ?      00:00:00 Agent_3g\nroot
2715 1 0 15:36 ?      00:00:00 wan_detector\nroot      2743 1 0 15:36 ?
00:00:04 slim\nroot      2763 2743 1 15:36 tty2      00:00:22 \usr\bin\X\nroot 2830
1 0 15:36 ?      00:00:00 dhclient br0\nroot      2831 1 0 15:36 ? 00:00:01
\bin\sh \scripts\status_check.sh\nstudent      2886 2743 0 15:36 ?      00:00:00
\bin\sh \etc\xinitrc xfce4\nstudent      2891 2886 0 15:36 ?      00:00:00 \bin\sh
\scripts\startxfce4\nstudent      2916 2891 0 15:36 ?      00:00:01 xfce4-session\nroot
2944 1 0 15:36 ?      00:00:00 \sbin\smi\nroot      2958 1 0 15:36 ?
00:00:00 \bin\heartbeat\nroot      2995 2 0 15:36 ?      00:00:00
[kjournald]\nstudent 3100 1 0 15:37 ?      00:00:00 dbus-launch --autolaunch
a822b50ba91705398f791a9200000055 --binary-syntax --close-stderr\nstudent 3120 1 0
15:37 ?      00:00:00 \usr\local\cisco\bin\dbus-daemon --fork --print-pid 5
--print-address 7 --session\nstudent 3128 1 0 15:37 ?      00:00:00
\usr\local\cisco\lib\xfce4\xfconf\xfconfd\nroot      3211 1 0 15:37 ?
00:00:00 init \nstudent 3225 1 0 15:37 ?      00:00:00 xfsettingsd\nstudent
3244 1 0 15:37 ?      00:00:00 \usr\local\cisco\libexec\gvfsd\nstudent 3255
1 0 15:37 ?      00:00:00 \usr\local\cisco\libexec\gvfs-fuse-daemon
\apps\localconfig\student\.\gvfs\nstudent 3259 1 0 15:37 ?      00:00:02
xfwm4\nstudent 3269 1 0 15:37 ?      00:00:02 xfce4-panel\nstudent 3278 1
0 15:37 ?      00:00:00 Thunar --daemon\nstudent 3287 1 0 15:37 ?      00:00:02
xfdesktop\nstudent 3309 2916 0 15:37 ?      00:00:00 3G_Dongle\nstudent 3311 2916
0 15:37 ?      00:00:01 BlueToothUI\nstudent 3329 2916 0 15:37 ?      00:00:01
wifi_status hide\nstudent 3337 2916 0 15:37 ?      00:00:03 wired_status hide >
\dev\null\nstudent 3338 1 0 15:37 ?      00:00:00 xfce4-settings-helper\nstudent
3363 3269 0 15:37 ? 00:00:01 \usr\local\cisco\lib\xfce4\panel\wrapper
\usr\local\cisco\lib\xfce4\panel\plugins\libsystray.so 6 16777251 systray
Notification Area Area where notification icons appear \nstudent 3401 1 0 15:37 ?
00:00:00 \usr\local\cisco\libexec\gvfs-hal-volume-monitor\nstudent 3411 1 0
15:37 ?      00:00:00 \usr\local\cisco\libexec\gvfsd-trash --spawner :1.4
\org\gtk\gvfs\exec_spaw\0\nstudent 3443 1 0 15:37 ?      00:00:00
\usr\local\cisco\libexec\gconfd-2\nstudent 3942 1 0 15:38 ?      00:00:00
\usr\local\cisco\lib\scim-1.0\scim-launcher -d -c simple -e all -f socket
--no-stay\nstudent 4079 1 0 15:38 ?      00:00:00
\usr\local\cisco\lib\scim-1.0\scim-helper-manager\nstudent 4080 1 0 15:38 ?
00:00:00 \usr\local\cisco\lib\scim-1.0\scim-panel-gtk --display :0.0 -c socket -d
--no-stay\nstudent 4082 1 0 15:38 ?      00:00:00
\usr\local\cisco\lib\scim-1.0\scim-launcher -d -c socket -e socket -f x11\nstudent
4290 1 0 15:38 ?      00:00:03 \usr\local\cisco\bin\Terminal\nstudent 4349
4290 0 15:38 ?      00:00:00 gnome-pty-helper\nstudent 4350 4290 0 15:38 pts\0
00:00:00 bash\nroot 4428 4350 0 15:38 pts\0      00:00:00 -bash\nroot 5492
2592 0 15:39 ?      00:00:01 nginx: worker process \nroot 6203 2831 0
16:13 ?      00:00:00 sleep 1\nroot 6225 2585 0 16:13 ?      00:00:00 sh -c cd
'\usr\local\cisco\nginx\html\api\1.0\sys' ;

```



```
LC_ALL=zh_CN.utf-8;\/usr\/local\/cisco\/bin\/ps -ef 2>&1\nroot      6226 6225 0 16:13 ?
00:00:00 \\/usr\/local\/cisco\/bin\/ps -ef\nroot      15865    2 0 15:50 ?          00:00:00
[RtmpTimerTask]\nroot      15866 2 0 15:50 ?          00:00:02 [RtmpMimeTask]\nroot
15867    2 0 15:50 ?          00:00:00 [RtmpCmdQTask]\nstudent  23415 4290 0 15:58
pts\/1    00:00:00 bash\nroot      23433 23415 0 15:58 pts\/1    00:00:00 -bash\nroot
30190 23433 0 16:05 pts\/1    00:00:00 clish\n", "success": "true", "getAt": "2012-12-14
08:13:07"}
```

Set Custom Fonts

Example: set custom fonts

Request

- Fonts are located in a local directory:

```
curl -k -X PUT -H 'password: cisco' -H 'Content-Type: application/json' -d '{"fonts" :
"/usr/local/cisco/share/fonts/thai-ttf/Waree.ttf}'
https://10.75.189.228/api/1.0/sys/fonts
```

- Fonts are obtained from an HTTP server:

```
curl -k -X PUT -H 'password: cisco' -H 'Content-Type: application/json' -d '{"fonts" :
"http://dl.1001fonts.com/rochester.zip}' https://10.75.189.228/api/1.0/sys/fonts
```

Reply

```
{"success": "true", "updatedAt": "2014-04-17 02:33:34"}
```

Parameter restrictions

The fonts can be obtained either from a local directory or from a remote site through HTTP, HTTPS, or FTP.

Get Fonts List

Example: get fonts list

Request

```
curl -k -X GET -H 'password: cisco' https://10.75.189.228/api/1.0/sys/fonts
```

Reply

```
{"fonts": "Fixed:style=Bold\nFixed:style=Bold
SemiCondensed\nFixed:style=SemiCondensed\nFixed:style=Oblique
SemiCondensed\nFixed:style=Oblique\nFixed:style=ko\nFixed:style=ja\nFixed:style=Regular\n"
, "success": "true", "getAt": "2014-04-17 02:41:19"}
```

Remove Custom Fonts

Example: remove custom fonts

Request

```
curl -k -X DELETE -m 60 -H 'password: cisco' -H 'Content-Type: application/json' -d
'{"fonts" : "Waree.ttf}' https://10.75.189.228/api/1.0/sys/fonts
```

Reply

```
{"success": "true", "updatedAt": "2014-04-17 02:53:21"}
```

Parameter restrictions

Provide the custom installed fonts name.

Set Master Audio Volume

Example: set master volume to 77 and set mute to false

Request

```
curl -k -X PUT -H 'password: cisco' -H 'Content-Type: application/json' -d
'{"mute": "false", "volume": "77" }' https://64.104.169.83/api/1.0/sys/audio/default/Master
```

Reply

```
{"success": "true", "updatedAt": "2012-12-14 07:50:00"}
```

Parameter restrictions

The volume range is 0–100. Valid values for mute are true and false.

Get Master Audio Volume

Example: get master audio volume information

Request

```
curl -k -X GET -H 'password: cisco' https://64.104.169.83/api/1.0/sys/audio/default/Master
```

Reply

```
{"mute": "true", "volume": "50", "success": "true", "getAt": "2013-01-01 03:03:02"}
```

Set Capture Audio Volume

Example: set capture volume to 77 and set mute to false

Request

```
curl -k -X PUT -H 'password: cisco' -H 'Content-Type: application/json' -d
'{"mute": "false", "volume": "77" }' https://64.104.169.83/api/1.0/sys/audio/default/Capture
```

Reply

```
{"success": "true", "updatedAt": "2012-12-14 07:50:00"}
```

Parameter restrictions

The volume range is 0–100. Valid values for mute are true and false.

Get Capture Audio Volume

Example: get capture audio volume information

Request

```
curl -k -X GET -H 'password: cisco'
https://64.104.169.83/api/1.0/sys/audio/default/Capture
```

Reply

```
{"mute": "true", "volume": "50", "success": "true", "getAt": "2013-01-01 03:03:02"}
```

Set Time Zone

Example: set timezone to Asia/Shanghai

Request

```
curl -k -X PUT -H 'password: cisco' -H 'Content-Type: application/json' -d
'{"timezone": "Asia/Shanghai"}' https://64.104.169.102/api/1.0/sys/timezone
```

Reply

```
{"success": "true", "updatedAt": "2012-12-14 07:50:00"}
```

Parameter restrictions

The valid strings for timezone are as following:

Africa/Sao_Tome, Africa/Johannesburg, Africa/Maputo, Africa/Lagos, Africa/Mogadishu, Africa/Dakar, Africa/Lubumbashi, Africa/Nouakchott, Africa/Timbuktu, Africa/Abidjan, Africa/Niamey, Africa/Tripoli, Africa/Bujumbura, Africa/Lusaka, Africa/Freetown, Africa/Kinshasa, Africa/Luanda, Africa/Ndjamena, Africa/Ceuta, Africa/Maseru, Africa/Khartoum, Africa/Dar_es_Salaam, Africa/Addis_Ababa, Africa/Algiers, Africa/Asmara, Africa/Libreville, Africa/Asmera, Africa/Djibouti, Africa/Malabo, Africa/Gaborone, Africa/Ouagadougou, Africa/Kigali, Africa/Porto-Novo, Africa/Windhoek, Africa/Monrovia, Africa/Cairo, Africa/Banjul, Africa/Conakry, Africa/Harare, Africa/Bangui, Africa/Tunis, Africa/Mbabane, Africa/Bissau, Africa/El_Aaiun, Africa/Brazzaville, Africa/Blantyre, Africa/Lome, Africa/Kampala, Africa/Nairobi, Africa/Accra, Africa/Bamako, Africa/Casablanca, Africa/Douala, America/Hermosillo, America/Rosario, America/Bahia, America/Martinique, America/Dominica, America/Atikokan, America/Recife, America/Aruba, America/Mendoza, America/Santarem, America/Montserrat, America/Jamaica, America/St_Johns, America/Knox_IN, America/Buenos_Aires, America/St_Kitts, America/Chicago, America/Guyana, America/Boise, America/North_Dakota/Center, America/North_Dakota/New_Salem, America/Miquelon, America/Lima, America/Regina, America/Inuvik, America/Pangnirtung, America/Winnipeg, America/Rankin_Inlet, America/Halifax, America/Nassau, America/Rio_Branco, America/Manaus, America/Cuiaba, America/Cancun, America/Blanc-Sablon, America/Moncton, America/Edmonton, America/Danmarkshavn, America/Jujuy, America/Ensenada, America/Tegucigalpa, America/Santiago, America/Cayenne, America/Merida, America/Belem, America/St_Lucia, America/Puerto_Rico, America/Glace_Bay, America/Coral_Harbour, America/Tijuana, America/Sao_Paulo, America/Mazatlan, America/La_Paz, America/Atka, America/Los_Angeles, America/Noronha, America/Tortola, America/Cayman, America/Porto_Velho, America/Caracas, America/Swift_Current, America/Mexico_City, America/Maceio, America/Cordoba, America/Rainy_River, America/Iqaluit, America/New_York, America/Montreal, America/Araguaina, America/Denver, America/Adak, America/Shiprock, America/Barbados, America/St_Vincent, America/Chihuahua, America/Guatemala, America/St_Thomas, America/Cambridge_Bay, America/Phoenix, America/Eirunepe, America/Belize, America/Guadeloupe, America/Antigua, America/Nome, America/Catamarca, America/Port_of_Spain, America/Indiana/Petersburg, America/Indiana/Vincennes, America/Indiana/Knox, America/Indiana/Marengo, America/Indiana/Winamac, America/Indiana/Vevay, America/Indiana/Indianapolis, America/Indiana/Tell_City, America/Paramaribo, America/Grand_Turk, America/Boa_Vista,

America/Panama, America/Thunder_Bay, America/Whitehorse, America/Godthab, America/Toronto,
 America/Goose_Bay, America/Guayaquil, America/Kentucky/Monticello,
 America/Kentucky/Louisville, America/Juneau, America/Campo_Grande, America/Anchorage,
 America/Argentina/Mendoza, America/Argentina/Buenos_Aires, America/Argentina/Rio_Gallegos,
 America/Argentina/La_Rioja, America/Argentina/Tucuman, America/Argentina/Jujuy,
 America/Argentina/Salta, America/Argentina/Cordoba, America/Argentina/San_Luis,
 America/Argentina/Ushuaia, America/Argentina/ComodRivadavia, America/Argentina/Catamarca,
 America/Argentina/San_Juan, America/Montevideo, America/Asuncion, America/Monterrey,
 America/Curacao, America/Menominee, America/Havana, America/Costa_Rica, America/Yakutat,
 America/Detroit, America/Managua, America/Louisville, America/Dawson_Creek, America/Dawson,
 America/Resolute, America/Nipigon, America/Yellowknife, America/Santo_Domingo,
 America/Fort_Wayne, America/Bogota, America/Indianapolis, America/Port-au-Prince,
 America/Grenada, America/Porto_Acre, America/Anguilla, America/St_Barthelemy,
 America/Fortaleza, America/Marigot, America/Scoresbysund, America/Virgin, America/El_Salvador,
 America/Thule, America/Vancouver, Antarctica/South_Pole, Antarctica/Mawson, Antarctica/Palmer,
 Antarctica/Rothera, Antarctica/DumontDUrville, Antarctica/Syowa, Antarctica/McMurdo,
 Antarctica/Davis, Antarctica/Vostok, Antarctica/Casey, Arctic/Longyearbyen, Asia/Brunei,
 Asia/Kuwait, Asia/Amman, Asia/Riyadh89, Asia/Aden, Asia/Thimphu, Asia/Jerusalem, Asia/Riyadh,
 Asia/Ujung_Pandang, Asia/Macao, Asia/Istanbul, Asia/Phnom_Penh, Asia/Shanghai, Asia/Bahrain,
 Asia/Taipei, Asia/Seoul, Asia/Bangkok, Asia/Ashgabat, Asia/Yerevan, Asia/Dhaka, Asia/Macau,
 Asia/Karachi, Asia/Tashkent, Asia/Dacca, Asia/Chongqing, Asia/Calcutta, Asia/Dubai,
 Asia/Chungking, Asia/Kashgar, Asia/Ashkhabad, Asia/Vientiane, Asia/Yakutsk, Asia/Beirut,
 Asia/Pontianak, Asia/Nicosia, Asia/Irkutsk, Asia/Tbilisi, Asia/Kuching, Asia/Tehran, Asia/Tokyo,
 Asia/Singapore, Asia/Bishkek, Asia/Dushanbe, Asia/Baku, Asia/Hong_Kong, Asia/Baghdad,
 Asia/Ulan_Bator, Asia/Damascus, Asia/Choibalsan, Asia/Jayapura, Asia/Yekaterinburg,
 Asia/Katmandu, Asia/Qyzylorda, Asia/Qatar, Asia/Kolkata, Asia/Kathmandu, Asia/Makassar,
 Asia/Ho_Chi_Minh, Asia/Thimbu, Asia/Oral, Asia/Sakhalin, Asia/Riyadh87, Asia/Magadan,
 Asia/Gaza, Asia/Anadyr, Asia/Krasnoyarsk, Asia/Aqtau, Asia/Riyadh88, Asia/Muscat, Asia/Urumqi,
 Asia/Omsk, Asia/Ulaanbaatar, Asia/Vladivostok, Asia/Kuala_Lumpur, Asia/Colombo,
 Asia/Samarkand, Asia/Kabul, Asia/Almaty, Asia/Harbin, Asia/Kamchatka, Asia/Aqtobe,
 Asia/Pyongyang, Asia/Novosibirsk, Asia/Manila, Asia/Jakarta, Asia/Rangoon, Asia/Tel_Aviv,
 Asia/Hovd, Asia/Dili, Asia/Saigon, Atlantic/Stanley, Atlantic/Madeira, Atlantic/St_Helena,
 Atlantic/Azores, Atlantic/Faroe, Atlantic/Jan_Mayen, Atlantic/South_Georgia, Atlantic/Faeroe,
 Atlantic/Reykjavik, Atlantic/Bermuda, Atlantic/Canary, Atlantic/Cape_Verde, Australia/Sydney,
 Australia/South, Australia/Currie, Australia/Perth, Australia/Yancowinna, Australia/North,
 Australia/West, Australia/Darwin, Australia/Lord_Howe, Australia/Lindeman, Australia/Victoria,
 Australia/NSW, Australia/ACT, Australia/Brisbane, Australia/Melbourne, Australia/Eucla,
 Australia/Hobart, Australia/Queensland, Australia/Adelaide, Australia/Tasmania, Australia/Canberra,
 Australia/LHI, Australia/Broken_Hill, Brazil/West, Brazil/Acre, Brazil/East, Brazil/DeNoronha,
 Canada/Pacific, Canada/Mountain, Canada/Atlantic, Canada/Central, Canada/Eastern,
 Canada/East-Saskatchewan, Canada/Saskatchewan, Canada/Newfoundland, Canada/Yukon, CET,
 Chile/EasterIsland, Chile/Continental, CST6CDT, Cuba, EET, Egypt, Eire, EST, EST5EDT,
 Etc/GMT+12, Etc/GMT+9, Etc/GMT, Etc/GMT-13, Etc/GMT0, Etc/GMT+11, Etc/GMT+5,
 Etc/GMT-0, Etc/GMT-2, Etc/GMT-10, Etc/GMT-11, Etc/GMT+2, Etc/UCT, Etc/GMT-1, Etc/GMT+3,
 Etc/GMT+0, Etc/UTC, Etc/GMT+10, Etc/GMT-7, Etc/Zulu, Etc/GMT-6, Etc/GMT+1, Etc/GMT-5,
 Etc/GMT-12, Etc/GMT+4, Etc/GMT-9, Etc/GMT+7, Etc/GMT+8, Etc/Universal, Etc/GMT-3,
 Etc/GMT-14, Etc/GMT-8, Etc/Greenwich, Etc/GMT+6, Etc/GMT-4, Europe/Bucharest, Europe/Riga,
 Europe/Kaliningrad, Europe/Monaco, Europe/San_Marino, Europe/Budapest, Europe/Gibraltar,
 Europe/Lisbon, Europe/Belgrade, Europe/Athens, Europe/Warsaw, Europe/Vatican, Europe/Chisinau,
 Europe/Istanbul, Europe/Madrid, Europe/Paris, Europe/Bratislava, Europe/Dublin, Europe/Helsinki,
 Europe/London, Europe/Zaporozhye, Europe/Volgograd, Europe/Tirane, Europe/Simferopol,
 Europe/Samara, Europe/Sarajevo, Europe/Mariehamn, Europe/Tallinn, Europe/Prague,
 Europe/Andorra, Europe/Ljubljana, Europe/Vienna, Europe/Kiev, Europe/Isle_of_Man, Europe/Oslo,

Europe/Nicosia, Europe/Skopje, Europe/Brussels, Europe/Rome, Europe/Belfast, Europe/Amsterdam, Europe/Sofia, Europe/Minsk, Europe/Zagreb, Europe/Luxembourg, Europe/Moscow, Europe/Tiraspol, Europe/Vaduz, Europe/Stockholm, Europe/Podgorica, Europe/Zurich, Europe/Copenhagen, Europe/Berlin, Europe/Jersey, Europe/Vilnius, Europe/Malta, Europe/Guernsey, Europe/Uzhgorod, Factory, GB, GB-Eire, GMT, GMT0, GMT-0, GMT+0, Greenwich, Hongkong, HST, Iceland, Indian/Cocos, Indian/Kerguelen, Indian/Christmas, Indian/Reunion, Indian/Antananarivo, Indian/Mauritius, Indian/Chagos, Indian/Maldives, Indian/Mayotte, Indian/Comoro, Indian/Mahe, Iran, Israel, Jamaica, Japan, Kwajalein, Libya, MET, Mexico/BajaNorte, Mexico/BajaSur, Mexico/General, Mideast/Riyadh89, Mideast/Riyadh87, Mideast/Riyadh88, MST, MST7MDT, Navajo, NZ, NZ-CHAT, Pacific/Majuro, Pacific/Guadalcanal, Pacific/Noumea, Pacific/Tongatapu, Pacific/Tarawa, Pacific/Palau, Pacific/Wallis, Pacific/Johnston, Pacific/Apia, Pacific/Kwajalein, Pacific/Kiritimati, Pacific/Norfolk, Pacific/Wake, Pacific/Easter, Pacific/Enderbury, Pacific/Fiji, Pacific/Ponape, Pacific/Gambier, Pacific/Pitcairn, Pacific/Truk, Pacific/Samoa, Pacific/Nauru, Pacific/Auckland, Pacific/Niue, Pacific/Fakaofu, Pacific/Rarotonga, Pacific/Chatham, Pacific/Tahiti, Pacific/Marquesas, Pacific/Funafuti, Pacific/Guam, Pacific/Saipan, Pacific/Honolulu, Pacific/Yap, Pacific/Pago_Pago, Pacific/Midway, Pacific/Kosrae, Pacific/Efate, Pacific/Galapagos, Pacific/Port_Moresby, Poland, Portugal, PRC, PST8PDT, ROC, ROK, Singapore, Turkey, UCT, Universal, US/Indiana-Starke, US/Pacific, US/Mountain, US/East-Indiana, US/Aleutian, US/Hawaii, US/Central, US/Eastern, US/Michigan, US/Samoa, US/Arizona, US/Alaska, UTC, WET, W-SU, Zulu,

Get Time Zone

Example: get timezone information

Request

```
curl -k -X GET -H 'password: cisco' https://64.104.169.102/api/1.0/sys/timezone
```

Reply

```
{"timezone": "Asia/Shanghai", "success": "true", "getAt": "2013-01-01 03:19:46"}
```

Ethernet API

Use the commands IN this section to configure Ethernet API.



Note

Curl is used as a sample tool to request APIs. 10.140.44.134 is used as a sample IP address of the Cisco Edge 300 series switch, and cisco123! is used as a sample admin password.

Set Gi1 Status

N/A. 003 error is reported if it is requested.

Get Gi1 Status

Example: get gi1 status

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/gi1/status
```

Reply

```
{"status": "enable", "success": "true", "getAt": "2012-11-08 08:47:44"}
```

Set Gi1 MAC

N/A. 003 error is reported if it is requested.

Get Gi1 MAC

Example: get gi1 mac

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/gi1/mac
```

Reply

```
{"mac": "1C:AA:07:97:A3:C0", "success": "true", "getAt": "2012-11-08 08:51:19"}
```

Set Gi1 output-queue-strategy

Example: set gi1 output-queue-strategy to wrr

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"oqs" : "wrr"}' https://10.140.44.134/api/1.0/eth/gi1/oqs
```

Reply

```
{"success": "true", "updatedAt": "2012-11-08 08:54:26"}
```

Parameter restrictions

wrr and strict are the valid strings for oqs. Otherwise, 004 error is reported.

Get Gi1 output-queue-strategy

Example: get gi1 output-queue-strategy

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/gi1/oqs
```

Reply

```
{"oqs": "wrr", "success": "true", "getAt": "2012-11-08 09:07:08"}
```

Set Gi1 Pause

Example: set gi1 pause to on

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"pause" : "on"}' https://10.140.44.134/api/1.0/eth/gi1/pause
```

Reply

```
{"success": "true", "updatedAt": "2012-11-08 08:54:26"}
```

Parameter restrictions

on and off are the valid strings for oqs. Otherwise, 004 error is reported.

Get Gi1 Pause

Example: get gi1 pause

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/gi1/pause
```

Reply

```
{"pause": "on", "success": "true", "getAt": "2012-11-08 09:29:55"}
```

Set Gi1 Priority

Example: set gi1 pause to on

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"priority" : "normal"}' https://10.140.44.134/api/1.0/eth/gi1/priority
```

Reply

```
{"success": "true", "updatedAt": "2012-11-08 09:37:31"}
```

Parameter restrictions

normal and high are the valid strings for oqs. Otherwise, 004 error is reported.

Get Gi1 Priority

Example: get gi1 priority

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/gi1/priority
```

Reply

```
{"priority": "normal", "success": "true", "getAt": "2012-11-08 09:39:52"}
```

Set Gi1 Rate Limit

Example: set gi1 rate limit to unknown-unicast

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"rateLim": "set unknown-unicast 100"}' https://10.140.44.134/api/1.0/eth/gi1/rateLim
```

Reply

```
{"success": "true", "updatedAt": "2012-11-08 09:37:31"}
```

Parameter restrictions

Only none and set broadcast/unknown-unicast/both [1-100] are valid parameters.

Get Gi1 Rate Limit

Example: get gi1 rate limit

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/gi1/rateLim
```

Reply

```
{"rateLim": "set unknown-unicast 100", "success": "true", "getAt": "2012-11-09 03:21:16"}
```

Set Gi1 Speed

Example: set gi1 rate limit to unknown-unicast

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"speed": "auto"}' https://10.140.44.134/api/1.0/eth/gi1/speed
```

Reply

```
{"success": "true", "updatedAt": "2012-11-08 09:37:31"}
```

Parameter restrictions

Only auto, 10, 100, and 1000 are valid parameters.

Get Gi1 Speed

Example: get gi1 rate limit

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/gi1/speed
```


Reply

```
{"speed": "auto", "success": "true", "getAt": "2012-11-14 07:03:00"}
```

Set Gi1 Duplex

Example: set gi1 duplex to auto

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"duplex": "auto"}' https://10.140.44.134/api/1.0/eth/gi1/duplex
```

Reply

```
{"success": "true", "updatedAt": "2012-11-08 09:37:31"}
```

Parameter restrictions

Only auto, full, and half are valid parameters.

Get Gi1 Duplex

Example: get gi1 rate limit

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/gi1/duplex
```

Reply

```
{"duplex": "auto", "success": "true", "getAt": "2012-11-15 01:18:14"}
```

Set Gi1 Information

Example: set gi1 rate limit to unknown-unicast, to pause to off

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"rateLim": "set unknown-unicast 100", "pause": "off"}' https://10.140.44.134/api/1.0/eth/gi1
```

Reply

```
{"success": "true", "updatedAt": "2012-11-08 09:37:31"}
```

Parameter restrictions

Only none and set broadcast/unknown-unicast/both [1-100] are valid parameters.

Get Gi1 Information

Example: get gi1 information

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/gi1
```

Reply

```
{ "status": "enable", "dns": "64.104.123.144
171.70.168.183", "mac": "1C:AA:07:97:A3:C0", "oqs": "strict", "pause": "on", "priority": "normal",
"rateLim": "set unknown-unicast 100", "speed": "100", "success": "true", "getAt": "2012-11-09
08:53:52" }
```

Set Fe1 Status

Example: set fe1 status to disable

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"status"
: "disable"}' https://10.140.44.134/api/1.0/eth/fe1/status
```

Reply

```
{ "success": "true", "updatedAt": "2012-11-08 08:54:26" }
```

Parameter restrictions

enable and disable are the valid strings for status. Otherwise, 004 error is reported.

Get Fe1 Status

Example: get fe1 status

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe1/status
```

Reply

```
{ "status": "disable", "success": "true", "getAt": "2012-11-21 06:33:40" }
```

Set Fe1 output-queue-strategy

Example: set gil output-queue-strategy to wrr

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"oqs" :
"wrr"}' https://10.140.44.134/api/1.0/eth/fe1/oqs
```

Reply

```
{ "success": "true", "updatedAt": "2012-11-21 06:35:35" }
```

Parameter restrictions

wrr and strict are the valid strings for oqs. Otherwise, 004 error is reported.

Get Fe1 output-queue-strategy

Example: get fe1 output-queue-strategy

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe1/oqs
```

Reply

```
{"oqs": "wrr", "success": "true", "getAt": "2012-11-08 09:07:08"}
```

Set Fe1 Priority

Example: set fe1 pause to ìonî

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"priority": "normal"}' https://10.140.44.134/api/1.0/eth/fe1/priority
```

Reply

```
{"success": "true", "updatedAt": "2012-11-21 06:42:18"}
```

Parameter restrictions

normal and high are the valid strings for oqs. Otherwise, 004 error is reported.

Get Fe1 Priority

Example: get gi1 priority

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe1/priority
```

Reply

```
{"priority": "normal", "success": "true", "getAt": "2012-11-08 09:39:52"}
```

Set Fe1 Rate Limit

Example: set fe1 rate limit to ìunknown-unicastî

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"rateLim": "set unknown-unicast 100"}' https://10.140.44.134/api/1.0/eth/fe1/rateLim
```

Reply

```
{"success": "true", "updatedAt": "2012-11-21 06:43:28"}
```

Parameter restrictions

Only none and set broadcast/unknown-unicast/both [1-100] are valid parameters.

Get Fe1 Rate Limit

Example: get fe1 rate limit

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe1/rateLim
```

Reply

```
{"rateLim": "set unknown-unicast 100", "success": "true", "getAt": "2012-11-09 03:21:16"}
```

Set Fe1 Speed

Example: set fe1 rate limit to unknown-unicast

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"speed" : "auto"}' https://10.140.44.134/api/1.0/eth/fe1/speed
```

Reply

```
{"success": "true", "updatedAt": "2012-11-08 09:37:31"}
```

Parameter restrictions

Only auto, 10, and 100 are valid parameters.

Get Fe1 Speed

Example: get fe1 rate limit

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe1/speed
```

Reply

```
{"speed": "auto", "success": "true", "getAt": "2012-11-14 07:03:00"}
```

Set Fe1 Duplex

Example: set fe1 duplex to auto

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"duplex" : "auto"}' https://10.140.44.134/api/1.0/eth/fe1/duplex
```

Reply

```
{"success": "true", "updatedAt": "2012-11-08 09:37:31"}
```

Parameter restrictions

Only auto, full and half are valid parameters.

Get Fe1 Duplex

Example: get fe1 rate limit

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe1/duplex
```

Reply

```
{"duplex":"auto","success":"true","getAt":"2012-11-15 01:18:14"}
```

Set Fe1 Information

Example: set fe1 rate limit to unknown-unicast, to pause to off

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"rateLim": "set unknown-unicast 100","duplex":"auto"}' https://10.140.44.134/api/1.0/eth/fe1
```

Reply

```
{"success":"true","updatedAt":"2012-11-08 09:37:31"}
```

Parameter restrictions

Only none and set broadcast/unknown-unicast/both [1-100] are valid parameters.

Get Fe1 Information

Example: get fe1 information

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe1
```

Reply

```
{"status":"disable","oqs":"wrr","priority":"normal","rateLim":"set unknown-unicast 100","speed":"auto","duplex":"auto","success":"true","getAt":"2012-11-21 06:47:52"}
```

Set Fe2 Status

Example: set fe2 status to `disable`

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"status": "disable"}' https://10.140.44.134/api/1.0/eth/fe2/status
```

Reply

```
{"success":"true","updatedAt":"2012-11-08 08:54:26"}
```

Parameter restrictions

enable and disable are the valid strings for status. Otherwise, 004 error is reported.

Get Fe2 Status

Example: get fe2 status

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe2/status
```

Reply

```
{"status": "disable", "success": "true", "getAt": "2012-11-21 06:33:40"}
```

Set Fe2 output-queue-strategy

Example: set gil output-queue-strategy to wrr

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"oqs" : "wrr"}' https://10.140.44.134/api/1.0/eth/fe2/oqs
```

Reply

```
{"success": "true", "updatedAt": "2012-11-21 06:35:35"}
```

Parameter restrictions

wrr and strict are the valid strings for oqs. Otherwise, 004 error is reported.

Get Fe2 output-queue-strategy

Example: get fe2 output-queue-strategy

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe2/oqs
```

Reply

```
{"oqs": "wrr", "success": "true", "getAt": "2012-11-08 09:07:08"}
```

Set Fe2 Priority

Example: set fe2 pause to ion

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"priority" : "normal"}' https://10.140.44.134/api/1.0/eth/fe2/priority
```

Reply

```
{"success": "true", "updatedAt": "2012-11-21 06:42:18"}
```

Parameter restrictions

normal and high are the valid strings for oqs. Otherwise, 004 error is reported.

Get Fe2 Priority

Example: get gi1 priority

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe2/priority
```

Reply

```
{"priority": "normal", "success": "true", "getAt": "2012-11-08 09:39:52"}
```

Set Fe2 Rate Limit

Example: set fe2 rate limit to unknown-unicast

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"rateLim" : "set unknown-unicast 100"}' https://10.140.44.134/api/1.0/eth/fe2/rateLim
```

Reply

```
{"success": "true", "updatedAt": "2012-11-21 06:43:28"}
```

Parameter restrictions

Only none and set broadcast/unknown-unicast/both [1-100] are valid parameters.

Get Fe2 Rate Limit

Example: get fe2 rate limit

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe2/rateLim
```

Reply

```
{"rateLim": "set unknown-unicast 100", "success": "true", "getAt": "2012-11-09 03:21:16"}
```

Set Fe2 Speed

Example: set fe2 rate limit to unknown-unicast

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"speed" : "auto"}' https://10.140.44.134/api/1.0/eth/fe2/speed
```

Reply

```
{"success": "true", "updatedAt": "2012-11-08 09:37:31"}
```

Parameter restrictions

Only auto, 10, and 100 are valid parameters.

Get Fe2 Speed

Example: get fe2 rate limit

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe2/speed
```

Reply

```
{"speed": "auto", "success": "true", "getAt": "2012-11-14 07:03:00"}
```

Set Fe2 Duplex

Example: set fe2 duplex to auto

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"duplex": "auto"}' https://10.140.44.134/api/1.0/eth/fe2/duplex
```

Reply

```
{"success": "true", "updatedAt": "2012-11-08 09:37:31"}
```

Parameter restrictions

Only auto, full and half are valid parameters.

Get Fe2 Duplex

Example: get fe2 rate limit

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe2/duplex
```

Reply

```
{"duplex": "auto", "success": "true", "getAt": "2012-11-15 01:18:14"}
```

Set Fe2 Information

Example: set fe2 rate limit to unknown-unicast, to pause to off

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"rateLim": "set unknown-unicast 100", "duplex": "auto"}' https://10.140.44.134/api/1.0/eth/fe2
```

Reply

```
{"success": "true", "updatedAt": "2012-11-08 09:37:31"}
```

Parameter restrictions

Only none and set broadcast/unknown-unicast/both [1-100] are valid parameters.

Get Fe2 Information

Example: get fe2 information

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe2
```

Reply

```
{"status": "disable", "oqs": "wrr", "priority": "normal", "rateLim": "set unknown-unicast 100", "speed": "auto", "duplex": "auto", "success": "true", "getAt": "2012-11-21 06:47:52"}
```

Set Fe3 Status

Example: set fe3 status to disable

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"status": "disable"}' https://10.140.44.134/api/1.0/eth/fe3/status
```

Reply

```
{"success": "true", "updatedAt": "2012-11-08 08:54:26"}
```

Parameter restrictions

enable and disable are the valid strings for status. Otherwise, 004 error is reported.

Get fe3 status

Example: get fe3 status

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe3/status
```

Reply

```
{"status": "disable", "success": "true", "getAt": "2012-11-21 06:33:40"}
```

Set Fe3 output-queue-strategy

Example: set gi1 output-queue-strategy to wrr

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"oqs": "wrr"}' https://10.140.44.134/api/1.0/eth/fe3/oqs
```

Reply

```
{"success": "true", "updatedAt": "2012-11-21 06:35:35"}
```

Parameter restrictions

wrr and strict are the valid strings for oqs. Otherwise, 004 error is reported.

Get Fe3 output-queue-strategy

Example: get fe3 output-queue-strategy

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe3/oqs
```

Reply

```
{"oqs": "wrr", "success": "true", "getAt": "2012-11-08 09:07:08"}
```

Set Fe3 priority

Example: set fe3 pause to 10n

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"priority": "normal"}' https://10.140.44.134/api/1.0/eth/fe3/priority
```

Reply

```
{"success": "true", "updatedAt": "2012-11-21 06:42:18"}
```

Parameter restrictions

normal and high are the valid strings for oqs. Otherwise, 004 error is reported.

Get fe3 priority

Example: get fe3 priority

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe3/priority
```

Reply

```
{"priority": "normal", "success": "true", "getAt": "2012-11-08 09:39:52"}
```

Set Fe3 Rate Limit

Example: set fe3 rate limit to unknown-unicast

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"rateLim": "set unknown-unicast 100"}' https://10.140.44.134/api/1.0/eth/fe3/rateLim
```

Reply

```
{"success": "true", "updatedAt": "2012-11-21 06:43:28"}
```

Parameter restrictions

Only none and set broadcast/unknown-unicast/both [1-100] are valid parameters.

Get Fe3 Rate Limit

Example: get fe3 rate limit

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe3/rateLim
```

Reply

```
{"rateLim": "set unknown-unicast 100", "success": "true", "getAt": "2012-11-09 03:21:16"}
```

Set Fe3 Speed

Example: set fe3 rate limit to unknown-unicast

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"speed" : "auto"}' https://10.140.44.134/api/1.0/eth/fe3/speed
```

Reply

```
{"success": "true", "updatedAt": "2012-11-08 09:37:31"}
```

Parameter restrictions

Only auto, 10, and 100 are valid parameters.

Get Fe3 Speed

Example: get fe3 rate limit

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe3/speed
```

Reply

```
{"speed": "auto", "success": "true", "getAt": "2012-11-14 07:03:00"}
```

Set Fe3 Duplex

Example: set fe3 duplex to auto

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"duplex" : "auto"}' https://10.140.44.134/api/1.0/eth/fe3/duplex
```

Reply

```
{"success": "true", "updatedAt": "2012-11-08 09:37:31"}
```

Parameter restrictions

Only auto, full, and half are valid parameters.

Get Fe3 Duplex

Example: get fe3 rate limit

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe3/duplex
```

Reply

```
{"duplex": "auto", "success": "true", "getAt": "2012-11-15 01:18:14"}
```

Set Fe3 Information

Example: set fe3 rate limit to unknown-unicast, pause, or off

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"rateLim": "set unknown-unicast 100", "duplex": "auto"}' https://10.140.44.134/api/1.0/eth/fe3
```

Reply

```
{"success": "true", "updatedAt": "2012-11-08 09:37:31"}
```

Parameter restrictions

Only none and set broadcast/unknown-unicast/both [1-100] are valid parameters.

Get Fe3 Information

Example: get fe3 information

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe3
```

Reply

```
{"status": "disable", "ogs": "wrr", "priority": "normal", "rateLim": "set unknown-unicast 100", "speed": "auto", "duplex": "auto", "success": "true", "getAt": "2012-11-21 06:47:52"}
```

Set Fe3 Status

Example: set fe3 status to disable

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"status": "disable"}' https://10.140.44.134/api/1.0/eth/fe3/status
```

Reply

```
{"success": "true", "updatedAt": "2012-11-08 08:54:26"}
```

Parameter restrictions

enable and disable are the valid strings for status. Otherwise, 004 error is reported.

Get Fe3 Status

Example: get fe3 status

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe3/status
```

Reply

```
{"status": "disable", "success": "true", "getAt": "2012-11-21 06:33:40"}
```

Set Fe3 output-queue-strategy

Example: set gi1 output-queue-strategy to wrr

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"oqs" : "wrr"}' https://10.140.44.134/api/1.0/eth/fe3/oqs
```

Reply

```
{"success": "true", "updatedAt": "2012-11-21 06:35:35"}
```

Parameter restrictions

wrr and strict are the valid strings for oqs. Otherwise, 004 error is reported.

Get Fe3 output-queue-strategy

Example: get fe3 output-queue-strategy

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe3/oqs
```

Reply

```
{"oqs": "wrr", "success": "true", "getAt": "2012-11-08 09:07:08"}
```

Set Fe3 Priority

Example: set fe3 pause to on

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"priority" : "normal"}' https://10.140.44.134/api/1.0/eth/fe3/priority
```

Reply

```
{"success": "true", "updatedAt": "2012-11-21 06:42:18"}
```

Parameter restrictions

normal and high are the valid strings for oqs. Otherwise, 004 error is reported.

Get Fe3 Priority

Example: get gil priority

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe3/priority
```

Reply

```
{"priority": "normal", "success": "true", "getAt": "2012-11-08 09:39:52"}
```

Set Fe3 Rate Limit

Example: set fe3 rate limit to unknown-unicast

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"rateLim" : "set unknown-unicast 100"}' https://10.140.44.134/api/1.0/eth/fe3/rateLim
```

Reply

```
{"success": "true", "updatedAt": "2012-11-21 06:43:28"}
```

Parameter restrictions

Only none and set broadcast/unknown-unicast/both [1-100] are valid parameters.

Get Fe3 Rate Limit

Example: get fe3 rate limit

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe3/rateLim
```

Reply

```
{"rateLim": "set unknown-unicast 100", "success": "true", "getAt": "2012-11-09 03:21:16"}
```

Set Fe3 Speed

Example: set fe3 rate limit to unknown-unicast

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"speed" : "auto"}' https://10.140.44.134/api/1.0/eth/fe3/speed
```

Reply

```
{"success": "true", "updatedAt": "2012-11-08 09:37:31"}
```

Parameter restrictions

Only auto, 10, 100, and 1000 are valid parameters.

Get Fe3 Speed

Example: get fe3 rate limit

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe3/speed
```

Reply

```
{"speed": "auto", "success": "true", "getAt": "2012-11-14 07:03:00"}
```

Set Fe3 Duplex

Example: set fe3 duplex to auto

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"duplex": "auto"}' https://10.140.44.134/api/1.0/eth/fe3/duplex
```

Reply

```
{"success": "true", "updatedAt": "2012-11-08 09:37:31"}
```

Parameter restrictions

Only auto, full, and half are valid parameters.

Get Fe3 Duplex

Example: get fe3 rate limit

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe3/duplex
```

Reply

```
{"duplex": "auto", "success": "true", "getAt": "2012-11-15 01:18:14"}
```

Set Fe3 Information

Example: set fe3 rate limit to unknown-unicast, to pause to off

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"rateLim": "set unknown-unicast 100", "duplex": "auto"}' https://10.140.44.134/api/1.0/eth/fe3
```

Reply

```
{"success": "true", "updatedAt": "2012-11-08 09:37:31"}
```

Parameter restrictions

Only none and set broadcast/unknown-unicast/both [1-100] are valid parameters.

Get Fe3 Information

Example: get fe3 information

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe3
```

Reply

```
{"status": "disable", "oqs": "wrr", "priority": "normal", "rateLim": "set unknown-unicast 100", "speed": "auto", "duplex": "auto", "success": "true", "getAt": "2012-11-21 06:47:52"}
```

Set Fe4 Status

Example: set fe4 status to disable

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"status": "disable"}' https://10.140.44.134/api/1.0/eth/fe4/status
```

Reply

```
{"success": "true", "updatedAt": "2012-11-08 08:54:26"}
```

Parameter restrictions

enable and disable are the valid strings for status. Otherwise, 004 error is reported.

Get Fe4 Status

Example: get fe4 status

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe4/status
```

Reply

```
{"status": "disable", "success": "true", "getAt": "2012-11-21 06:33:40"}
```

Set Fe4 output-queue-strategy

Example: set gil output-queue-strategy to wrr

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"oqs": "wrr"}' https://10.140.44.134/api/1.0/eth/fe4/oqs
```

Reply

```
{"success": "true", "updatedAt": "2012-11-21 06:35:35"}
```

Parameter restrictions

wrr and strict are the valid strings for oqs. Otherwise, 004 error is reported.

Get Fe4 output-queue-strategy

Example: get fe4 output-queue-strategy

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe4/oqs
```

Reply

```
{"oqs": "wrr", "success": "true", "getAt": "2012-11-08 09:07:08"}
```

Set Fe4 priority

Example: set fe4 pause to on

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"priority": "normal"}' https://10.140.44.134/api/1.0/eth/fe4/priority
```

Reply

```
{"success": "true", "updatedAt": "2012-11-21 06:42:18"}
```

Parameter restrictions

normal and high are the valid strings for oqs. Otherwise, 004 error is reported.

Get Fe4 Priority

Example: get fe4 priority

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe4/priority
```

Reply

```
{"priority": "normal", "success": "true", "getAt": "2012-11-08 09:39:52"}
```

Set Fe4 Rate Limit

Example: set fe4 rate limit to unknown-unicast

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"rateLim": "set unknown-unicast 100"}' https://10.140.44.134/api/1.0/eth/fe4/rateLim
```

Reply

```
{"success": "true", "updatedAt": "2012-11-21 06:43:28"}
```

Parameter restrictions

Only none and set broadcast/unknown-unicast/both [1-100] are valid parameters.

Get Fe4 Rate Limit

Example: get fe4 rate limit

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe4/rateLim
```

Reply

```
{"rateLim": "set unknown-unicast 100", "success": "true", "getAt": "2012-11-09 03:21:16"}
```

Set Fe4 Speed

Example: set fe4 rate limit to unknown-unicast

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"speed" : "auto"}' https://10.140.44.134/api/1.0/eth/fe4/speed
```

Reply

```
{"success": "true", "updatedAt": "2012-11-08 09:37:31"}
```

Parameter restrictions

Only auto, 10, and 100 are valid parameters.

Get Fe4 Speed

Example: get fe4 rate limit

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe4/speed
```

Reply

```
{"speed": "auto", "success": "true", "getAt": "2012-11-14 07:03:00"}
```

Set Fe4 Duplex

Example: set fe4 duplex to auto

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"duplex" : "auto"}' https://10.140.44.134/api/1.0/eth/fe4/duplex
```

Reply

```
{"success": "true", "updatedAt": "2012-11-08 09:37:31"}
```

Parameter restrictions

Only auto, full, and half are valid parameters.

Get Fe4 Duplex

Example: get fe4 rate limit

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe4/duplex
```

Reply

```
{"duplex":"auto","success":"true","getAt":"2012-11-15 01:18:14"}
```

Set Fe4 Information

Example: set fe4 rate limit to unknown-unicast, to pause to off

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"rateLim": "set unknown-unicast 100","duplex":"auto"}' https://10.140.44.134/api/1.0/eth/fe4
```

Reply

```
{"success":"true","updatedAt":"2012-11-08 09:37:31"}
```

Parameter restrictions

Only none and set broadcast/unknown-unicast/both [1-100] are valid parameters.

Get Fe4 Information

Example: get fe4 information

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe4
```

Reply

```
{"status":"disable","ogs":"wrr","priority":"normal","rateLim":"set unknown-unicast 100","speed":"auto","duplex":"auto","success":"true","getAt":"2012-11-21 06:47:52"}
```

Set Fe4 Status

Example: set fe4 status to disable

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"status": "disable"}' https://10.140.44.134/api/1.0/eth/fe4/status
```

Reply

```
{"success":"true","updatedAt":"2012-11-08 08:54:26"}
```

Parameter restrictions

enable and disable are the valid strings for status. Otherwise, 004 error is reported.

Get Fe4 Status

Example: get fe4 status

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe4/status
```

Reply

```
{"status": "disable", "success": "true", "getAt": "2012-11-21 06:33:40"}
```

Set Fe4 output-queue-strategy

Example: set gil output-queue-strategy to wrr

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"oqs" : "wrr"}' https://10.140.44.134/api/1.0/eth/fe4/oqs
```

Reply

```
{"success": "true", "updatedAt": "2012-11-21 06:35:35"}
```

Parameter restrictions

wrr and strict are the valid strings for oqs. Otherwise, 004 error is reported.

Get Fe4 output-queue-strategy

Example: get fe4 output-queue-strategy

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe4/oqs
```

Reply

```
{"oqs": "wrr", "success": "true", "getAt": "2012-11-08 09:07:08"}
```

Set Fe4 Priority

Example: set fe4 pause to on

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"priority" : "normal"}' https://10.140.44.134/api/1.0/eth/fe4/priority
```

Reply

```
{"success": "true", "updatedAt": "2012-11-21 06:42:18"}
```

Parameter restrictions

normal and high are the valid strings for oqs. Otherwise, 004 error is reported.

Get Fe4 Priority

Example: get gi1 priority

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe4/priority
```

Reply

```
{"priority": "normal", "success": "true", "getAt": "2012-11-08 09:39:52"}
```

Set Fe4 Rate Limit

Example: set fe4 rate limit to unknown-unicast

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"rateLim" : "set unknown-unicast 100"}' https://10.140.44.134/api/1.0/eth/fe4/rateLim
```

Reply

```
{"success": "true", "updatedAt": "2012-11-21 06:43:28"}
```

Parameter restrictions

Only none and set broadcast/unknown-unicast/both [1-100] are valid parameters.

Get Fe4 Rate Limit

Example: get fe4 rate limit

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe4/rateLim
```

Reply

```
{"rateLim": "set unknown-unicast 100", "success": "true", "getAt": "2012-11-09 03:21:16"}
```

Set Fe4 Speed

Example: set fe4 rate limit to unknown-unicast

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"speed" : "auto"}' https://10.140.44.134/api/1.0/eth/fe4/speed
```

Reply

```
{"success": "true", "updatedAt": "2012-11-08 09:37:31"}
```

Parameter restrictions

Only auto, 10, 100, and 1000 are valid parameters.

Get Fe4 Speed

Example: get fe4 rate limit

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe4/speed
```

Reply

```
{"speed": "auto", "success": "true", "getAt": "2012-11-14 07:03:00"}
```

Set Fe4 Duplex

Example: set fe4 duplex to auto

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"duplex": "auto"}' https://10.140.44.134/api/1.0/eth/fe4/duplex
```

Reply

```
{"success": "true", "updatedAt": "2012-11-08 09:37:31"}
```

Parameter restrictions

Only auto, full, and half are valid parameters.

Get Fe4 Duplex

Example: get fe4 rate limit

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe4/duplex
```

Reply

```
{"duplex": "auto", "success": "true", "getAt": "2012-11-15 01:18:14"}
```

Set Fe4 Information

Example: set fe4 rate limit to unknown-unicast, to pause to off

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"rateLim": "set unknown-unicast 100", "duplex": "auto"}' https://10.140.44.134/api/1.0/eth/fe4
```

Reply

```
{"success": "true", "updatedAt": "2012-11-08 09:37:31"}
```

Parameter restrictions

Only none and set broadcast/unknown-unicast/both [1-100] are valid parameters.

Get Fe4 Information

Example: get fe4 information

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/eth/fe4
```

Reply

```
{"status": "disable", "oqs": "wrr", "priority": "normal", "rateLim": "set unknown-unicast 100", "speed": "auto", "duplex": "auto", "success": "true", "getAt": "2012-11-21 06:47:52" }
```

Issue a Command



Note

Curl is used as a sample tool to request APIs. 10.140.44.134 is used as a sample IP address of the Cisco Edge 300 series switch, and cisco123! is used as a sample admin password.

Reboot Cisco Edge 300

Example: reboot Cisco Edge 300

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/cmd/reboot
```

Reply

```
{"reboot": "true", "success": "true", "getAt": "2012-11-12 05:10:43" }
```

Image Version Information



Note

Curl is used as a sample tool to request APIs. 10.140.44.134 is used as a sample IP address of the Cisco Edge 300 series switch, and cisco123! is used as a sample admin password.

Get OS Version Information

Example: get os version information

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/image/os
```

Reply

```
{"os": "1.3.9.1", "success": "true", "getAt": "2012-11-12 05:51:56" }
```

Get 3rd App Version Information

Example: get 3rd app version information

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/image/3rdapp
```

Reply

```
{"3rdapp": "1.3.9.1", "success": "true", "getAt": "2012-11-12 05:56:39"}
```

Get OS and 3rd App Version in One Go

Example: get os and 3rdapp version information in one go

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/image
```

Reply

```
{"os": "1.3.9.1", "3rdapp": "1.3.9.1", "success": "true", "getAt": "2012-11-12 08:18:58"}
```

AP Information



Note

Curl is used as a sample tool to request APIs. 10.140.44.134 is used as a sample IP address of the Cisco Edge 300 series switch, and cisco123! is used as a sample admin password.

Set AP SSID

Example: set SSID to cisco

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"ssid": "cisco"}' https://10.140.44.134/api/1.0/wifi/ap/ssid
```

Reply

```
{"success": "true", "updatedAt": "2012-11-12 06:25:47"}
```

Parameter restrictions

The length of AP SSID should be between 1 to 32 characters and the valid parameter set is {a-zA-Z0-9-_
} or 004 error is reported.

Get AP SSID

Example: get SSID

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/wifi/ap/ssid
```

Reply

```
{"ssid": "abc", "success": "true", "getAt": "2012-11-12 06:22:54"}
```

Set AP Radio

Example: set AP radio to off

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"radio" : "off"}' https://10.140.44.134/api/1.0/wifi/ap/radio
```

Reply

```
{"success": "true", "updatedAt": "2012-11-12 06:34:20"}
```

Parameter restrictions

Only on and off are valid parameters or 004 error is reported.

Get Radio Status

Example: get radio status

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/wifi/ap/radio
```

Reply

```
{"radio": "on", "success": "true", "getAt": "2012-11-12 06:33:20"}
```

Set Wireless Mode

Example: set wireless mode to 9

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"wirelessMode" : "7"}' https://10.140.44.134/api/1.0/wifi/ap/wirelessMode
```

Reply

```
{"success": "true", "updatedAt": "2012-11-12 07:25:00"}
```

Parameter restrictions

Only 0, 1, 4, 6, 7, and 9 are valid parameters or 004 error is reported.

Get Wireless Mode

Example: get wireless mode

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/wifi/ap/wirelessMode
```

Reply

```
{"wirelessMode": "9", "success": "true", "getAt": "2012-11-12 07:23:22"}
```

Set Channel Number

Example: set channel number to 9

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
'{"channelNumber" : "9"}' https://10.140.44.134/api/1.0/wifi/ap/channelNumber
```

Reply

```
{"success": "true", "updatedAt": "2012-11-12 07:32:04"}
```

Parameter restrictions

Only 0–14 integers are valid parameters or 004 error is reported.

Get Channel Number

Example: get channel number

Request

```
curl -k -X GET -H 'password: cisco123!'
https://10.140.44.134/api/1.0/wifi/ap/channelNumber
```

Reply

```
{"channelNumber": "6", "success": "true", "getAt": "2012-11-12 07:29:41"}
```

Set Channel Allocation

Example: set channel allocation to china

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
'{"channelAllocation" : "4"}' https://10.140.44.134/api/1.0/wifi/ap/channelAllocation
```

Reply

```
{"success": "true", "updatedAt": "2012-11-12 07:32:04"}
```

Parameter restrictions

Only 1–4 integers are valid parameters or 004 error is reported.

Get Channel Allocation

Example: get channel allocation

Request

```
curl -k -X GET -H 'password: cisco123!'
https://10.140.44.134/api/1.0/wifi/ap/channelAllocation
```

Reply

```
{"channelNumber": "6", "success": "true", "getAt": "2012-11-12 07:29:41" }
```

Set Channel Bandwidth

Example: set channel bandwidth to 20

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
'{"channelBandwidth" : "20"}' https://10.140.44.134/api/1.0/wifi/ap/channelBandwidth
```

Reply

```
{"success": "true", "updatedAt": "2012-11-12 07:32:04" }
```

Parameter restrictions

Only 20 and 20/40 are valid parameters or 004 error is reported.

Get Channel Bandwidth

Example: get channel bandwidth

Request

```
curl -k -X GET -H 'password: cisco123!'
https://10.140.44.134/api/1.0/wifi/ap/channelBandwidth
```

Reply

```
{"channelNumber": "6", "success": "true", "getAt": "2012-11-12 07:29:41" }
```

Set Transmit Power

Example: set transmit power to 50

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
'{"transmitPower" : "50"}' https://10.140.44.134/api/1.0/wifi/ap/transmitPower
```

Reply

```
{"success": "true", "updatedAt": "2012-11-12 07:49:35" }
```

Parameter restrictions

Only 1–100 integers are valid parameters or 004 error is reported.

Get Transmit Power

Example: get transmit power

Request

```
curl -k -X GET -H 'password: cisco123!'
https://10.140.44.134/api/1.0/wifi/ap/transmitPower
```

Reply

```
{"transmitPower": "100", "success": "true", "getAt": "2012-11-12 07:48:15"}
```

Set MCS

Example: set mcs to 15

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"mcs" :
"15"}' https://10.140.44.134/api/1.0/wifi/ap/mcs
```

Reply

```
{"success": "true", "updatedAt": "2012-11-12 07:49:35"}
```

Parameter restrictions

Only 0–15 and 33 integers are valid parameters or 004 error is reported.

Get MCS

Example: get mcs

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/wifi/ap/mcs
```

Reply

```
{"mcs": "33", "success": "true", "getAt": "2012-11-12 07:57:08"}
```

Set IGMP Snoop

Example: set igmp snoop to on

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
'{"igmpSnoop" : "on"}' https://10.140.44.134/api/1.0/wifi/ap/igmpSnoop
```

Reply

```
{"success": "true", "updatedAt": "2012-11-12 08:08:05"}
```

Parameter restrictions

Only on and off are valid parameters or 004 error is reported.

Get IGMP Snoop

Example: get igmp snoop

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/wifi/ap/igmpSnoop
```

Reply

```
{"igmpSnoop": "off", "success": "true", "getAt": "2012-11-12 08:06:44"}
```

Set Encryption

Example 1: set encryption mode to open and type to none

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"mode" : "open", "type": "none"}' https://10.140.44.134/api/1.0/wifi/ap/encryption
```

Reply

```
{"success": "true", "updatedAt": "2012-11-12 08:08:05"}
```

Example 2: set encryption mode to open, type to wep, key number to 1, key type to ASCII, key value to cisco

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"mode" : "open", "type": "wep", "keyNum": "1", "keyType": "ascii", "key": "cisco"}' https://10.140.44.134/api/1.0/wifi/ap/encryption
```

Reply

```
{"success": "true", "updatedAt": "2012-11-12 08:08:05"}
```

Parameter restrictions

The following are the commands on Cisco Edge 300:

```
encryption mode open type none  
encryption mode open type wep key [1-4] [ascii|hex] [key]
```

When the mode is open, only none and wep are valid types.

When you use the none type, you must not specify any other parameters.

When you use the wep type, you must specify the key number, key type, and key.

Key number: 1–4

Key type: ASCII or hex

When the key type is ASCII, {a-zA-Z0-9-_-} is the valid character set, and the length must be 5 or 13.
 When the key type is hex, {a-f0-9} is the valid character set, and the length must be 10 or 26.

Example 3: set encryption mode to shared, type to wep, key number to 1, key type to ASCII, key value to cisco

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"mode": "shared", "type": "wep", "keyNum": "1", "keyType": "ascii", "key": "cisco"}' https://10.140.44.134/api/1.0/wifi/ap/encryption
```

Reply

```
{"success": "true", "updatedAt": "2012-11-12 08:08:05"}
```

Parameter restrictions

The following is the command on Cisco Edge 300:

```
encryption mode shared type wep key [1-4] [ascii|hex] [key]
```

When you use the shared mode, you must specify other 4 parameters:

Key type: must be wep

Key number: 1–4

Key type: ASCII or hex

When the key type is ASCII, {a-zA-Z0-9-_-} is the valid character set, and the length must be 5 or 13.

When the key type is hex, {a-f0-9} is the valid character set, and the length must be 10 or 26.

Example 4: set encryption mode to wpa and key type to aes

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"mode": "wpa", "type": "aes"}' https://10.140.44.134/api/1.0/wifi/ap/encryption
```

```
Reply{"success": "true", "updatedAt": "2012-11-12 08:08:05"}
```

Parameter restrictions

The following is the command on Cisco Edge 300:

```
encryption mode wpa type [tkip|aes|tkipaes]
```

When you use wpa mode, you must specify type as following:

type: must be tkip, aes, or tkipaes

Example 5: set encryption mode to wpa and key type to aes

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"mode": "wpa", "type": "aes"}' https://10.140.44.134/api/1.0/wifi/ap/encryption
```

Reply

```
{"success": "true", "updatedAt": "2012-11-12 08:08:05"}
```

Parameter restrictions

The following is the command on Cisco Edge 300:

```
encryption mode [wpa|wpa2|wpa1wpa2] type [tkip|aes|tkipaes]
```

When you use wpa, wpa2, or wpa1wpa2 mode, you must specify type as following:

type: must be tkip, aes, or tkipaes

Example 6: set encryption mode to wpapsk and key type to tkipaes

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
'{"mode": "wpapsk", "type": "tkipaes", "passPhrase": "cisco12345"}'
https://10.140.44.134/api/1.0/wifi/ap/encryption
```

Reply

```
{"success": "true", "updatedAt": "2012-11-12 08:08:05"}
```

Parameter restrictions

The following is the command on Cisco Edge 300:

```
encryption mode [wpapsk|wpa2psk|wpapskwpa2psk] type [tkip|aes|tkipaes] pass-phrase [key]
```

When you use wpapsk, wpa2psk, or wpapskwpa2psk mode, you must specify type as following:

type: must be tkip, aes, or tkipaes,

pass phase: The valid character set for passphrase is {0-9a-zA-Z_-}, and the length is between 8 and 63.

Set Radius Server

Example 1: set RADIUS server host to 1.1.1.1, auth-port to 444, and key to cisco123

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
'{"host": "1.1.1.1", "auth-port": "444", "key": "cisco123"}'
https://10.140.44.134/api/1.0/wifi/ap/radius
```

Reply

```
{"success": "true", "updatedAt": "2012-11-12 08:08:05"}
```

Parameter restrictions

host: mandatory parameter. You must enter a valid IPv4 address.

auth-port: optional parameter. You must enter a number range between 0 and 65535.

key: optional parameter. The valid character set is {0-9 a-z A-Z _-`~!@#%&^*()+;,<>./[]{}&}

Get Radius Server

Example: get RADIUS server

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/wifi/ap/radius
```

Reply

```
{"host_1": "1.1.1.1", "auth-port_1": "444", "key_1": "cisco123", "host_2": "2.2.2.2", "key_2": "2.2.2.2", "host_3": "3.3.3.3", "auth-port_3": "1234", "key_3": "cisco", "success": "true", "getAt": "2012-11-14 06:35:30"}
```

Set AP Information

Example: set AP SSID to cisco and RADIUS server to {"host": "1.1.1.1", "auth-port": "555", "key": "cisco123"}

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"ssid": "cisco", "radius": {"host": "1.1.1.1", "auth-port": "555", "key": "cisco123"}}' https://10.140.44.134/api/1.0/wifi/ap/
```

Reply

```
{"success": "true", "updatedAt": "2012-11-19 02:46:54"}
```

Get AP Information

Example: get AP information

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.134/api/1.0/wifi/ap
```

Reply

```
{"channelAllocation": "3", "igmpSnoop": "off", "radio": "off", "wirelessMode": "7", "channelBandwidth": "20", "mcs": "15", "radius": {"host_1": "1.1.1.1", "auth-port_1": "555", "key_1": "cisco123"}, "channelNumber": "9", "multicastMcs": "15", "ssid": "cisco", "encryption": {"mode": "wpa2psk", "keyType": "tkipaes", "key": "Cisco123"}, "transmitPower": "50", "success": "true", "getAt": "2012-11-19 02:56:25"}
```

Wifi Client Information

**Note**

Curl is used as a sample tool to request APIs. 10.140.44.134 is used as a sample IP address of the Cisco Edge 300 series switch, and cisco123! is used as a sample admin password.

**Note**

Assume that the WiFi client IP address is 10.140.44.148.

Get ID of a Network

Example: get a new network ID

Request

```
curl -k -X GET -H 'password: cisco123!'  
https://10.140.44.148/api/1.0/wifi/client/new_network_id
```

Reply

```
{"new_network_id": "3", "success": "true", "getAt": "2011-04-21 04:26:46"}
```

Get SSID of a Network

Example: get SSID of network 0

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.148/api/1.0/wifi/client/0/ssid
```

Reply

```
{"ssid": "blizzard", "success": "true", "getAt": "2011-04-21 04:09:14"}
```

Parameter restrictions

The length of parameter should be less than 33 characters or 004 error is reported.

Set SSID of a Network

Example: set an SSID of network 0 to Cisco Edge 300

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"ssid" :  
"blizzard_2"}' https://10.140.44.148/api/1.0/wifi/client/0/ssid
```

Reply

```
{"ssid": "blizzard", "success": "true", "getAt": "2011-04-21 05:07:50"}
```

Get an SSID Scan Status of a Network

Example: check SSID scan status of network 0

Request

```
curl -k -X GET -H 'password: cisco123!'  
https://10.140.44.148/api/1.0/wifi/client/0/scan_ssid
```

Reply

```
{"scan_ssid": "0", "success": "true", "getAt": "2011-04-21 04:20:36"}
```

Set SSID Scan of a Network

Example: set an SSID scan of network 1

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
 '{"scan_ssid": "1"}' https://10.140.44.148/api/1.0/wifi/client/0/scan_ssid
```

Reply

```
{"success": "true", "updatedAt": "2011-04-21 06:13:49"}
```

Parameter restrictions

Only 0 and 1 are valid parameters or 004 error is reported.

Get Key Management Type of a Network

Example: get key management type of network 0

Request

```
curl -k -X GET -H 'password: cisco123!'
 https://10.140.44.148/api/1.0/wifi/client/0/key_mgmt
```

Reply

```
{"key_mgmt": "WPA-EAP", "success": "true", "getAt": "2011-04-21 04:22:47"}
```

Set Key Management Type of a Network

Example: set key management type of a network to WPA-EAP

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
 '{"key_mgmt": "WPA-EAP"}' https://10.140.44.148/api/1.0/wifi/client/0/key_mgmt
```

Reply

```
{"success": "true", "updatedAt": "2011-04-21 06:13:49"}
```

Parameter restrictions

Only WPA-PSK, WPA-EAP, and None are valid parameters or 004 error is reported.

Get Pairwise Type of a Network

Example: get pairwise type of network

Request

```
curl -k -X GET -H 'password: cisco123!'
 https://10.140.44.148/api/1.0/wifi/client/0/pairwise
```

Reply

```
{"pairwise": "CCMP", "success": "true", "getAt": "2011-04-21 04:27:17"}
```

Set Pairwise Type of a Network

Example: set pairwise type of a network to CCMP

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"pairwise": "CCMP"}' https://10.140.44.148/api/1.0/wifi/client/0/pairwise
```

Reply

```
{"success": "true", "updatedAt": "2011-04-21 06:13:49"}
```

Parameter restrictions

Only CCMP and TKIP are valid parameters or 004 error is reported.

Get Group of a Network

Example: get group of network 0

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.148/api/1.0/wifi/client/0/group
```

Reply

```
{"group": "CCMP TKIP WEP104 WEP40", "success": "true", "getAt": "2011-04-21 04:29:11"}
```

Set Group of a Network

Example: set group of a network to CCMP TKIP WEP104 WEP40

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d '{"group": "CCMP"}' https://10.140.44.148/api/1.0/wifi/client/0/group
```

Reply

```
{"success": "true", "updatedAt": "2011-04-21 06:13:49"}
```

Parameter restrictions

Only CCMP, TKIP, WEP104, and WEP40 are valid parameters or 004 error is reported.

Get PSK of a Network

Example: get psk of network 0

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.148/api/1.0/wifi/client/0/psk
```

Reply

```
{"psk": "*", "success": "true", "getAt": "2011-04-21 04:33:16"}
```



Note If PSK is not set, the return value is FAIL. If it is set, the return value is *.

Set PSK of a Network

Example: set psk of a network to cisco

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
'{"psk":"cisco"}' https://10.140.44.148/api/1.0/wifi/client/0/psk
```

Reply

```
{"success":"true","updatedAt":"2011-04-21 06:13:49"}
```

Parameter restrictions

The PSK parameter must be a string includes 0-9,a-z,A-Z _- and the length should be 8–63.

Get wep_key0 of a Network

Example: get wep_key0 of network 0

Request

```
curl -k -X GET -H 'password: cisco123!'
https://10.140.44.148/api/1.0/wifi/client/0/wep_key0
```

Reply

```
{"wep_key0":"*", "success":"true", "getAt":"2011-04-21 04:37:46"}
```



Note The wep_key0 is in cipher text to protect private information of customer.

Set wep_key0 of a Network

Example 1: set wep_key0 of network 0 to cisco

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
'{"wep_key0":"cisco"}' https://10.140.44.148/api/1.0/wifi/client/0/wep_key0
```

Reply

```
{"success":"true","updatedAt":"2011-04-21 06:13:49"}
```

Example 2: set wep_key0 of network 0 to 01234567890

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
'{"wep_key0":"0x123456789a"}' https://10.140.44.148/api/1.0/wifi/client/0/wep_key0
```

Reply

```
{"success": "true", "updatedAt": "2011-04-21 04:11:51"}
```

Parameter restrictions

Parameter length should be 5 or 13 in ASCII or 10 or 26 in hex digit. If the parameter is in hex digit, it must have a prefix of 0x (The prefix length is not counted). Or 004 error is reported.

Get wep_key1 of a Network

Example: get wep_key1 of network 0

Request

```
curl -k -X GET -H 'password: cisco123!'
https://10.140.44.148/api/1.0/wifi/client/0/wep_key1
```

Reply

```
{"wep_key1": "*", "success": "true", "getAt": "2011-04-21 04:37:46"}
```



Note The wep_key1 is in cipher text to protect private information of customer.

Set wep_key1 of a Network

Example 1: set wep_key1 of network 0 to cisco

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
'{"wep_key1": "cisco"}' https://10.140.44.148/api/1.0/wifi/client/0/wep_key1
```

Reply

```
{"success": "true", "updatedAt": "2011-04-21 06:13:49"}
```

Example 2: set wep_key1 of network 0 to 01234567890

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
'{"wep_key1": "0x123456789a"}' https://10.140.44.148/api/1.0/wifi/client/0/wep_key1
```

Reply

```
{"success": "true", "updatedAt": "2011-04-21 04:11:51"}
```

Parameter restrictions

Parameter length should be 5 or 13 in ASCII or 10 or 26 in hex digit. If the parameter is in hex digit, it must have a prefix of 0x (The prefix length is not counted). Or 004 error is reported.

Get wep_key2 of a Network

Example: get wep_key2 of network 0

Request

```
curl -k -X GET -H 'password: cisco123!'
https://10.140.44.148/api/1.0/wifi/client/0/wep_key2
```

Reply

```
{"wep_key2": "***", "success": "true", "getAt": "2011-04-21 04:37:46"}
```



Note The wep_key2 is in cipher text to protect private information of customer.

Set wep_key2 of a Network

Example 1: set wep_key2 of network 0 to cisco

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
'{"wep_key2": "cisco"}' https://10.140.44.148/api/1.0/wifi/client/0/wep_key2
```

Reply

```
{"success": "true", "updatedAt": "2011-04-21 06:13:49"}
```

Example 2: set wep_key2 of network 0 to 01234567890

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
'{"wep_key2": "0x123456789a"}' https://10.140.44.148/api/1.0/wifi/client/0/wep_key2
```

Reply

```
{"success": "true", "updatedAt": "2011-04-21 04:11:51"}
```

Parameter restrictions

Parameter length should be 5 or 13 in ASCII or 10 or 26 in hex digit. If the parameter is in hex digit, it must have a prefix of 0x (The prefix length is not counted). Or 004 error is reported.

Get wep_key3 of a Network

Example: get wep_key3 of network 0

Request

```
curl -k -X GET -H 'password: cisco123!'
https://10.140.44.148/api/1.0/wifi/client/0/wep_key3
```

Reply

```
{"wep_key3": "***", "success": "true", "getAt": "2011-04-21 04:37:46"}
```



Note The wep_key3 is in cipher text to protect the private information of customer.

Set wep_key3 of a Network

Example 1: set wep_key3 of network 0 to cisco

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
 '{"wep_key3": "cisco"}' https://10.140.44.148/api/1.0/wifi/client/0/wep_key3
```

Reply

```
{"success": "true", "updatedAt": "2011-04-21 06:13:49"}
```

Example 2: set wep_key3 of network 0 to 01234567890

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
 '{"wep_key3": "0x123456789a"}' https://10.140.44.148/api/1.0/wifi/client/0/wep_key3
```

Reply

```
{"success": "true", "updatedAt": "2011-04-21 04:11:51"}
```

Parameter restrictions

Parameter length should be 5 or 13 in ASCII or 10 or 26 in hex digit. If the parameter is in hex digit, it must have a prefix of 0x (The prefix length is not counted). Or 004 error is reported.

Get EAP Type of a Network

Example: get EAP type of network 0

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.148/api/1.0/wifi/client/0/eap
```

Reply

```
{"eap": "PEAP", "success": "true", "getAt": "2011-04-21 04:38:13"}
```

Set EAP Type of a Network

Example: set EAP type of network 0 to PEAP

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
 '{"eap": "PEAP"}' https://10.140.44.148/api/1.0/wifi/client/0/eap
```

Reply

```
{"success": "true", "updatedAt": "2011-04-21 06:13:49"}
```

Parameter restrictions

Only MSCHAPV2, TLS, PEAP, TTLS, FAST, and LEAP are valid parameters or 004 error is reported.

Get EAP Identity String of a Network

Example: get EAP identity string of network 0

Request

```
curl -k -X GET -H 'password: cisco123!'
https://10.140.44.148/api/1.0/wifi/client/0/identity
```

Reply

```
{"identity": "ce300", "success": "true", "getAt": "2011-04-21 04:43:04"}
```

Set EAP Identity String of a Network

Example: set EAP identity string of network 0 to ce300

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
'{"identity": "ce300"}' https://10.140.44.148/api/1.0/wifi/client/0/identity
```

Reply

```
{"success": "true", "updatedAt": "2011-04-21 06:13:49"}
```

Get Password of a Network

Example: get EAP password of network 0

Request

```
curl -k -X GET -H 'password: cisco123!'
https://10.140.44.148/api/1.0/wifi/client/0/password
```

Reply

```
{"password": "*", "success": "true", "getAt": "2011-04-21 04:49:46"}
```



Note The password is encrypted, which is shown as *.

Set Password of a Network

Example: set password of network 0 to ce300

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
'{"password": "ce300"}' https://10.140.44.148/api/1.0/wifi/client/0/password
```

Reply

```
{"success": "true", "updatedAt": "2011-04-21 06:13:49"}
```


Set the Status of a Network

Example: set the status of network 0 to enable

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
 '{"status": "enable"}' https://10.140.44.148/api/1.0/wifi/client/0/status
```

Reply

```
{"success": "true", "updatedAt": "2011-04-21 06:13:49"}
```

Parameter restrictions

Only enable and disable are valid parameters or 004 error is reported.

Remove a Network

Example: remove network 0

Request

```
curl -k -X DELETE -m 10 -H 'password: cisco123!' -H 'Content-Type: application/json'
 https://10.140.44.148/api/1.0/wifi/client/0
```

Reply

```
{"success": "true", "updatedAt": "2011-04-21 06:13:49"}
```



Note

The parameter `-m` sets the maximum time in seconds that the whole operation is allowed to take, which in this example is 10 seconds. If there is only one network on ce300, the remove operation will disconnect it from network. Therefore, if `-m` is not set, the request is holding.

Save the Network Configuration

Example: save the network configuration

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.148/api/1.0/wifi/client/saving
```

Reply

```
{"saving": "OK", "success": "true", "getAt": "2011-04-21 04:20:33"}
```

Show Connection Status

Example: show connection status

Request

```
curl -k -X GET -H 'password: cisco123!' https://10.140.44.148/api/1.0/wifi/client/
```

Reply

```
{ "conn_status": "bssid=a4:56:30:5d:e1:d0\nssid=blizzard\nnid=0\nmode=station\npairwise_cipher=CCMP\nwpa_group_cipher=CCMP\nwpa_key_mgmt=WPA2\nIEEE802.1X\nEAP\nwpa_state=COMPLETED\nip_address=10.140.44.148\naddress=1c:aa:07:97:a3:c8\nSupplicant PAE state=AUTHENTICATED\nsuppPortStatus=Authorized\nEAP state=SUCCESS\nselectedMethod=25 (EAP-PEAP)\nEAP TLS cipher=AES256-SHA\nEAP-PEAPv1 Phase2 method=GTC\n", "success": "true", "getAt": "2011-04-21 04:56:29" }
```

Reload the Saved Configuration

Example: reload the saved configuration

Request

```
curl -k -X GET -H 'password: cisco123!'
https://10.140.44.148/api/1.0/wifi/client/reconfiguration
```

Reply

```
{ "reconfiguration": "OK", "success": "true", "getAt": "2011-04-21 04:58:15" }
```

Export Configuration File

Example: export wifi-network-only configuration file to /tmp

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
'{"type": "wifi-network-only", "destination": "/tmp"}'
https://10.140.44.148/api/1.0/configuration/export
```

Reply

```
{ "success": "true", "updatedAt": "2011-04-21 05:12:11" }
```

Parameter restrictions

The type field can be wifi-network-only, overall, and startup-config. The destination field can be any string that contains 0-9 A-Z a-z.

Import Configuration File

Example: import wifi-network-only configuration file from /tmp

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
'{"type": "wifi-network-only", "source": "/abc"}'
https://10.140.44.148/api/1.0/configuration/import
```

Reply

```
{ "success": "true", "updatedAt": "2011-04-21 06:49:02" }
```

Parameter restrictions

The type field can be wifi-network-only, overall, and startup-config. The source field can be any string containing 0-9 A-Z a-z.

RS232 Configuration

**Note**

Curl is used as a sample tool to request APIs. 10.140.44.134 is used as a sample IP address of the Cisco Edge 300 series switch, and cisco123! is used as a sample admin password.

Configure RS232

Example: configure RS-232 with the device name of /dev/ttyS0, baud rate of 9600, data rates of 8, stop bits of 1, parity of none, hex command of 123456

Request

```
curl -k -X PUT -H 'password: cisco123!' -H 'Content-Type: application/json' -d
'{"device": "/dev/ttyS0", "data_rate": "9600", "data_bits": "8", "stop_bits": "1", "parity_bits": "n", "command": "123456"}' https://64.104.163.36/api/1.0/rs232/configuration
```

Reply

```
{"success": "true", "updatedAt": "2011-04-21 07:11:28"}
```

Parameter restrictions

The device parameter should be a string started with /dev/.

The data_rate parameter should be a string in the set {50, 75, 110, 134, 150, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 500000, 576000}.

The data_bits parameter should be 7 or 8.

The stop_bits parameter should be 1 or 2.

The parity_bits parameter should be n, o, and e.

Upgrade

**Note**

Curl is used as a sample tool to request APIs. 10.140.44.134 is used as a sample IP address of the Cisco Edge 300 series switch, and cisco123! is used as a sample admin password.

Upgrade an Image

Example: upgrade an image using smi_local

Request

```
curl -k -X PUT -H 'password: cisco' -H 'Content-Type: application/json' -d
'{"protocol": "nfs", "url": "64.104.169.66:/home/", "path": "/1.5rb1/", "version": "1.5rb1", "upgr
ade-fm": "on", "upgrade-os": "off", "upgrade-apps": "off", "force-upgrade": "on", "clear-config": "
off"}' https://64.104.169.70/api/1.0/upgrade/images
```

Reply

```
{"success": "true", "updatedAt": "2011-04-21 07:11:28"}
```

Parameter restrictions

- version—The package version, which supports 1.5rb1 or above.
- protocol—The protocol to get a delivery image (http/https/ftp/udisk/nfs/cifs).
- url—The parent folder URL of the delivery image. The '/' at the end of the URL is necessary.
- path—For nfs/cifs/udisk, if the image is not in the top directory, it can be used to specified the sub-directory path.
- upgrade-fm—Valid value is on or off. Specify whether to upgrade factory or not.
- upgrade-os—Valid value is on or off. Specify whether to upgrade OS or not.
- upgrade-apps—Valid value is on or off. Specify whether to upgrade Cisco apps or not.
- force-upgrade—Valid value is on or off. Specify whether to compare versions between image and current system.
- username—User name for HTTP, FTP, or CIFS.
- password—Password for HTTP, FTP, or CIFS.
- clear-config—Valid value is on or off, while on means to clear config of smi_local; otherwise it should be off.

Pre-condition

Have super-user privileges when you perform the upgrade operation.

Error Codes

Table 6-1 Error Codes and Descriptions

Error Code	Description
001	Invalid password
002	Content type error
003	The HTTP method used is not supported for this resource
004	Illegal parameter
005	Error is found when the commands are executed
006	Invalid resource
101	Json: maximum stack depth exceeded
102	Json: underflow or the modes mismatch
103	Json: unexpected control character found

Table 6-1 *Error Codes and Descriptions*

Error Code	Description
104	Json: syntax error, malformed JSON
105	Json: malformed UTF-8 characters, possibly incorrectly encoded
106	Json: unknown error

