

Dell Networking N-Series
N2200-ON, and N3200-ON
Switches

RESTful API User Guide
Version 6.8.0

Notes and Cautions



NOTE: A NOTE indicates important information that helps you make better use of your computer.



CAUTION: A CAUTION indicates potential damage to hardware or loss of data if instructions are not followed.

India

This product conforms to the relevant Essential Requirements of TEC, Department of Telecommunications, Ministry of Communications, Govt of India, New Delhi[1]110001

Information in this publication is subject to change without notice.

Copyright © 2022 Dell Inc. All rights reserved.

Reproduction of these materials in any manner whatsoever without the written permission of Dell Inc. is strictly forbidden.

This product is protected by U.S. and international copyright and intellectual property laws. Dell™ and the Dell logo are trademarks of Dell Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

Marketing Models: N2224X-ON, N2224PX-ON, N2248X-ON, N2248PX-ON, N3208PX-ON, N3224T-ON, N3224F-ON, N3248X-ON, N3248TE-ON, N3224PX-ON, N3224P-ON, N3248P-ON, N3248PXE-ON

Regulatory Models: E04W, E05W, E06W, E07W, E15W, E16W, E17W, E18W, E30W, E31W, E32W, E41W

Contents

1	About This Document	7
	Acronyms and Abbreviations	7
	Document Conventions	7
	Additional Documentation.	8
2	RESTful OpEN API Overview	9
	Base URL.	9
	Special Keywords and Parameters	10
	Response Status Message	11
	HTTP Response Codes.	11
	Error Codes.	12
3	REST Document Requests	15
	Sessions	15
4	Rest API Reference	17
	Authentication	17
	Switching	17
	Overlay Networking	17

System	18
Authentication	19
Login	19
Logout	23
dot1dbaseconfig	25
dot1dtpconfig	28
dot1dtpfdbentries	32
dot1dtpportentries	36
dot1sconfig	40
dot1sinterfaces	45
dot1smstentries	54
fdb	65
fdbstats	71
Switching	76
Interfaces	76
VLANs	91
Overlay Networking	105
IP	105
lagports	111
lags	115
mfdbs	125
System	130
Registry	130
serviceport	133
Snooping Configuration	138
Snooping Interfaces	144
snoopingqueriers	152
snoopingssms	158
Snooping VLANs	163
stormcontrolports	171
swmirror	179
swmonitor	186

swportmonitor	193
System1213	200
systemconfig	205
systemcontrol.	210
systemmisc	213

About This Document

This document describes how to use the Dell Networking RESTful API, which allows you to configure various operational parameters and attributes, including selected protocols on the target switch. It is intended for software developers who design RESTful applications that target Dell Networking switch platforms.

Acronyms and Abbreviations

In most cases, acronyms and abbreviations are defined on first use.

Document Conventions

Table 1-1 describes the typographical conventions this document uses.

Table 1-1. Document Conventions

Convention	Description
Bold	Page names, field names, menu options, button names, and CLI commands and keywords.
<code>courier font</code>	Command-line text (CLI output) and file names
[]	In a command line, square brackets indicate an optional entry.
{ }	In a command line, inclusive brackets indicate a selection of compulsory parameters separated by the character. One option must be selected. For example: spanning-tree mode {stp rstp mstp} means that for the spanning-tree mode command, stp , rstp , or mstp must be entered.
<i>Italic</i>	In a command line, indicates a variable.
<Enter>	Any individual key on the keyboard.
CTRL + Z	A keyboard combination that involves pressing the Z key while holding the CTRL key.

Additional Documentation

The following documents for the Dell Networking N-Series switches are available at www.dell.com/support:

- *Dell Networking N1108EP-ON, N1500, N2200-ON, and N3200-ON Series Switches User's Configuration Guide*—discusses and provides examples on how to configure, monitor, and maintain Dell Networking N-Series switches by using web-based Dell OpenManage Switch Administrator utility or the command-line interface (CLI).
- *Dell Networking N1108EP-ON, N1500, N2200-ON, and N3200-ON Series Switches CLI Reference Guide*—provides information about the command-line interface (CLI) commands used to configure and manage the switch. The document provides in-depth CLI descriptions, syntax, default values, and usage guidelines.

RESTful OpEN API Overview

The Dell Networking RESTful API allows you to configure various operational parameters and attributes, including selected protocols on the target switch. Since the API is based on REST principles, it is very easy to write and test applications. You can use your browser to test URLs, and you can use pretty much any HTTP client in any programming language to interact with the API.

Please note, although Internet-based browsers can issue requests, the intended use of this REST API is a programming interface. Most modern programming languages provide support for constructing and parsing JSON, as well as issuing HTTP requests. Examples include Python's 'requests' and 'json' modules, and the Java classes in the `org.apache.http` and `org.json.simple` namespaces. The REST of the API examples in this document demonstrate the use of the libcurl library using the cURL command-line tool found at <http://curl.haxx.se/>.

Base URL

All URL references in this documentation have the following base: `serveraddress:RESTful_port/open/version`, where:

- The server-address is the IP address or host name of the target switch.
- RESTful_port is the configured Layer 4 port number in the range 1025 to 65535 for RESTful connection. The default RESTful port is 8080.

Following the server-address:

- RESTful_port is the *open* resource, which is used to redirect the request to the Dell Networking N-Series RESTful APIs for Dell OpEN.
- Versioning is mandatory and is part of the URL. The version is denoted by using the letter *v* followed by a single version number.

For example, `http://10.10.10.1:8080/open/v1/`

Special Keywords and Parameters

RESTful APIs support several resource keywords that are used to expand and complement the REST API. The supported keywords are as follows:

- **debug**
 - Debugging capabilities are available for each request in an effort to assist and provide some insight into the details of the request. Debugging is available and defined using the following levels:
 - 0: Disabled (default).
 - 1: Trace. General tracing, may identify OpEN API function and return codes.
 - 2: Data. Request data is echoed for verification purposes.
 - 3: Verbose. The highest level of verbosity.
 - The debug parameter format is `?debug=<level>`
- **fields**
 - The `fields` parameter is used to identify or reduce the resource content when used with a GET request. By default, the complete data set for the particular resource is returned. The amount of data returned may be overwhelming for some requests. The `fields` parameter can be used to reduce and identify only the interesting or desired resources.
 - The `fields` parameter format is `?fields=<resource1>,<resource2>,...`
- **method**
 - Some clients can make only GET requests and have the inability to make POST, PUT, or DELETE requests. To assist with this limitation, all the APIs can be performed using a GET request using the special method parameter.
 - The method parameter format is `?method=<post|put|get|delete>`
- **offset/limit**

- The `offset` and `limit` keywords relate to pagination when used within a GET request. The offset denotes the starting index, whereas the limit identifies the number of resources to return. The default is also `offset:1` and `limit:10`. Note that `offset` and `limit` can be specified independently.
 - The `offset/limit` parameter format is `?offset=<index>,&limit=<count>`
- `suppress_response_codes`
 - Some clients accept only HTTP success codes (200). This parameter can be used to suppress the response codes in the header. Note that regardless of the suppression parameter, every response contains a status message. For more information, please see HTTP Response Codes.
 - The `suppress_response_codes` parameter format is `?suppress_response_codes=true`

Response Status Message

Each authenticated request is accompanied by a status message in the response which provides detailed information.

Table 2-2. Status Message Information

Parameter	Type	Description
<code>response_code</code>	integer	HTTP status code.
<code>message</code>	string	OK or a textual description of the failure.
<code>more_info</code>	string	Optional. Additional information, such as allowable ranges, hints, and so on.
<code>error_code</code>	integer	OpEN API error codes.

HTTP Response Codes

The HTTP status codes utilized are kept to a minimum to simplify the API and its use. Note that all the HTTP response codes are possible returns for all RESTful requests.

Table 2-3. HTTP Response Codes

Response Code	Description
200	The request has succeeded.
400	The request could not be understood by the server due to malformed syntax or invalid parameters.
401	The request is not authorized.
403	The request is well-formed, but forbidden for the requested resource.
404	The server has not found anything matching the Request-URI. This is a generic failure.
405	The request URI is not allowed.
409	Something prohibits the operation, but it can be retried.
500	The server encountered an internal error and could not complete the request. This is a generic failure.
501	The requested feature is not available.

Error Codes

The error codes directly map to the OpEN API function library. The status `response_code` should provide enough information for normal operation. In some cases, however, the `error_code` may provide additional information.

Table 2-4. Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-20	OPEN_E_RPC	Error in RPC.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-23	OPEN_E_FULL	Maximum count is already reached or table full.
-24	OPEN_E_EXISTS	Already exists.
-25	OPEN_E_TIMEOUT	Operation timeout.

Table 2-4. Error Codes (Continued)

Error Code	Code Name	Description
-26	OPEN_E_FAIL	Operation fail.
-27	OPEN_E_DISABLED	Disabled.
-28	OPEN_E_UNAVAIL	Parameter/feature is not supported.
-29	OPEN_E_NOT_FOUND	Parameter not found.
-30	OPEN_E_EMPTY	Nothing to report or table is empty.
-31	OPEN_E_ERROR	Error conditions.

REST Document Requests

Note that the markdown formatted document, as well as specific subsections of the document, are available as resources. Document requests can be made using a doc resource as shown in the following examples:

```
http://10.10.10.1:8080/open/v1/doc
http://10.10.10.1:8080/open/v1/interfaces/doc
http://10.10.10.1:8080/open/v1/vlans/doc
```

Sessions

Every request requires prior authentication and must contain query arguments that specify a session ID.

For HTTPS:

```
SIDSSL=<sid-value>
```

For HTTP:

```
SID=<sid-value>
```

See [Login](#) for details on how to authenticate and obtain an sid-value.

The following is an HTTPS example:

```
curl -b "SIDSSL=<sid-value>" -GET -XPUT -d "ipv4addr=
10.14.240.45&ipv4netmask=255.255.255.0&ipv4gateway=
10.14.240.1" 10.10.10.1:8080/open/v1/networkport
```


Rest API Reference

RESTful APIs are split into several sub-resources. This section describes the following resources:

Authentication

- [Login](#)
- [Logout](#)
- [dot1dbaseconfig](#)
- [dot1dtpconfig](#)
- [dot1dtpfdbentries](#)
- [dot1dtpportentries](#)
- [dot1sconfig](#)
- [dot1sinterfaces](#)
- [dot1smstientries](#)
- [fdbbs](#)
- [fdbstats](#)

Switching

- [Interfaces](#)
- [VLANs](#)

Overlay Networking

- [IP](#)
- [lagports](#)
- [lags](#)
- [mfdbs](#)

System

- [Registry](#)
- [serviceport](#)
- [Snooping Configuration](#)
- [Snooping Interfaces](#)
- [snoopingqueriers](#)
- [snoopingssms](#)
- [Snooping VLANs](#)
- [stormcontrolports](#)
- [swmirror](#)
- [swmonitor](#)
- [swportmonitor](#)
- [System1213](#)
- [systemconfig](#)
- [systemcontrol](#)
- [systemmisc](#)

Authentication

Login

Implements a RESTful API providing user authentication and authorization.

A user session is required for all RESTful API requests. A user session is established by logging in with a valid username and password. Upon successful login, a session identifier (SID) is returned. Use this SID with all future requests. When you are finished with your session, logout. A successful logout request deletes the SID and invalidates the session. Note that the SID will automatically be deleted if the hard or soft time-out is exceeded. The username, password, and switch access level, as well as the session time-out parameters, are managed by your system administrator.

It is recommended that you always use HTTPS to ensure security.

Login CREATE

POST /open/v1/login

Use this call to create a login session.

Upon successful login, a user SID (or SIDSSL for HTTPS) is returned. Use this SID for future requests until you are done with the session by logging out. Note that while the login feature provides access control to the switch over HTTP connections, it does not provide security. This is because the SID is sent as cleartext over HTTP. A spoofer listening for packets can determine the SID and issues requests to the switch through the RESTful API using the same SID. For security, HTTPS is a better choice.

Request Details

Table 4-5. Login CREATE Request Details

Parameter	Type	Description
username	String	User name or identifier.
password	String	User password.

Response Details

Table 4-6. Login CREATE Response Details

Parameter	Type	Description
SID	String	Unique session ID for HTTP requests.
SIDSSL	String	Unique session ID for HTTPS requests.
accessLevel	Integer	15 = Read/write. 1 = Read only.

HTTP and HTTPS Response Codes

The possible HTTP response codes are 200, 400, 401, 403, 404, and 501.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-7. Login CREATE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Login successful.
-22	OPEN_E_PARAM	Error in parameters passed.
-23	OPEN_E_FULLL	Maximum number of login sessions exceeded.
-26	OPEN_E_FAIL	Authentication failed.
-28	OPEN_E_UNAVAIL	Feature not available.
-29	OPEN_E_NOT_FOUND	Invalid user name or password.

Request Example

```
openssl s_client -showcerts -connect 10.10.10.1:443 >
cacert.pem
curl -cacert cacert.pem -d "username=admin&password=test1234"
https://10.10.10.1:8080/open/v1/login
curl -d "username=admin&password=test1234"
http://10.10.10.1:8080/open/v1/login

browser:

http://10.10.10.1:8080/open/v1/login?username=admin&password=
test1234
```

Response Example

Returns an appropriate status message.

NOTE: For Internet browser requests, the SID value is returned as a cookie and is not returned as part of the following JSON data.

```
{
  "SIDSSL" :
rQAVUkzFbLUwhNvDHnfBIezOAETHpQFiGIddSDITQeSXuncbchCKO
DaohTVYldhSLMVfRfYIjsh,
  "accessLevel" : 15,
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Error Example

```
curl -cacert cacert.pem -d "username=admin&password=
badpassword" https://10.10.10.1:8080/open/v1/login
curl -d "username=admin&password=badpassword"
http://10.10.10.1:8080/open/v1/login
```

browser:

```
http://10.10.10.1:8080/open/v1/login?method=post&username=
admin&password=badpassword
```

Response Example

```
{
  "status": {
    "response_code": 401,
    "msg":
    "Unauthorized",
    "more_info":
    "Invalid username or password.",
    "error_code": -26
  }
}
```

Login READ

Login does not support read semantics.

Login UPDATE

Login does not support update semantics.

Login DELETE

Login does not support delete semantics.

Logout

Implements a RESTful API providing user session logout.

Logout CREATE

POST /open/v1/logout

Use this call to logout of a session. A successful logout request deletes the session ID and invalidates the session.

Request Details

Table 4-8. Logout CREATE Request Details

Parameter	Type	Description
SID	String	Unique session ID for HTTP requests.
SIDSSL	String	Unique session ID for HTTPS requests.

Response Details

HTTP and HTTPS Response Codes

The possible HTTP Response codes are 200 and 401.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-9. Logout CREATE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Logout successful.

Request Example

```
curl -cacert cacert.pem -b "SIDSSL=
uNyhgTSxGgEXx1SDhZykD0llaCjfYwMVjlcqgwpMevJBGdHOcfAHTMSwocBOyn
JIanAImPuSkFu" https://10.10.10.1:8080/open/v1/logout
curl -b "SID=
uNyhgTSxGgEXx1SDhZykD0llaCjfYwMVjlcqgwpMevJBGdHOcfAHTMSwocBOyn
JIanAImPuSkFu" http://10.10.10.1:8080/open/v1/logout
```

browser

```
http://10.10.10.1:8080/open/v1/logout
```

Response Example

Returns an appropriate status message.

```
{
  "status": {
    "response_code": 200,
    "msg":
    "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Error Example

```
curl -cacert cacert.pem -b "SIDSSL=badsid"
https://10.10.10.1:8080/open/v1/logout
```

Response Example

```
{
  "status": {
    "response_code": 401,
    "msg": "Unauthorized",
    "more_info": "",
    "error_code": -26
  }
}
```

Logout READ

Logout does not support read semantics.

Logout UPDATE

Logout does not support update semantics.

Logout DELETE

Logout does not support delete semantics.

dot1dbaseconfig

dot1dbaseconfig implements a RESTful API for the OpEN RFC1493 API. It focuses on the “dot1dBridge” object defined by that RFC.

dot1dbaseconfig CREATE

This component does not support create semantics.

dot1dbaseconfig READ

`GET /open/v1/dot1dbaseconfig`

Retrieves the various attributes supported by this resource.

Request Details

Table 4-10. dot1dbaseconfig READ Request Details

Parameter	Type	Description
fields	—	Partial response syntax.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-11. dot1dbaseconfig READ Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Request failed.

Request Example

```
curl -GET 10.10.10.1:8080/open/v1/dot1dbaseconfig
```

browser

```
http://10.10.10.1:8080/open/v1/dot1dbaseconfig
```

Response Example

```
{
  "baseBridgeAddress": "11:22:33:44:55:66",
  "baseNumPorts": 20,
  "baseType": 2,
  "status":
  {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Partial Request Example

The response can be reduced by using the partial response syntax as denoted by the 'fields' parameter.

```
curl -GET -d "fields=baseBridgeAddress"
10.10.10.1:8080/open/v1/dot1dbaseconfig
```

browser

```
http://10.10.10.1:8080/open/v1/dot1dbaseconfig?fields=
baseBridgeAddress
```

Response Example

```
{
  "baseBridgeAddress": "11:22:33:44:55:66",
  "status":
  {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Error Example

```
curl -GET -d "tenant=-99"
10.10.10.1:8080/open/v1/dot1dbaseconfig
```

browser

```
http://10.10.10.1:8080/open/v1/dot1dbaseconfig?method=
get&tenant=-99
```

Response Example

Returns an appropriate status message.

```
{
  "status":
  {
    "response_code": 400,
    "msg": "Bad Request",
    "more_info": "Invalid parameter",
    "error_code": -22
  }
}
```

dot1dbaseconfig UPDATE

This component does not support update semantics

dot1dbaseconfig DELETE

This component does not support delete semantics.

dot1dtpconfig

dot1dtpconfig implements a RESTful API for the OpEN RFC1493 API. It focuses on the *dot1dBridge* object defined by that RFC.

dot1dtpconfig CREATE

This component does not support create semantics.

dot1dtpconfig READ

```
GET /open/v1/dot1dtpconfig
```

Retrieves the various attributes supported by this resource.

Request Details

Table 4-12. dot1dtpconfig READ Request Details

Parameter	Type	Description
learnedEntryDiscards	integer	Number of MAC addresses that were not added to the table.
agingTime	integer	Timeout period for aging out dynamically-learned forwarding information.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-13. dot1dtpconfig READ Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Request failed.

Request Example

```
curl -GET 10.10.10.1:8080/open/v1/dot1dtpconfig  
browser  
http://10.10.10.1:8080/open/v1/dot1dtpconfig
```

Response Example

```
{  
  "learnedEntryDiscards": 7,  
  "agingTime": 15,  
  "status":  
  {  
    "response_code": 200,  
    "msg": "OK",  
    "more_info": "",  
    "error_code": 0  
  }  
}
```

Error Example

```
curl -GET -d "tenant=-99"  
10.10.10.1:8080/open/v1/dot1dtpconfig  
browser  
http://10.10.10.1:8080/open/v1/dot1dtpconfig?method=  
get&tenant=-99
```

Response Example

Returns an appropriate status message.

```
{ "status": { "response_code": 400, "msg": "Bad Request",  
  "more_info": "Invalid parameter", "error_code": -22 } }
```

dot1dtpconfig UPDATE

```
PUT /open/v1/dot1dtpconfig
```

Use this call to update various dot1dtp attributes.

Request Details

Table 4-14. dot1dtpconfig UPDATE Request Details

Parameter	Type	Description
agingTime	integer	Timeout period for aging out dynamically learned forwarding information (mandatory).

NOTE: Minimum = 10 seconds, maximum = 1 000,000 seconds

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-15. dot1dtpconfig UPDATE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Request failed.

Request Example

```
curl -GET -XPUT -d "agingTime=60"  
10.10.10.1:8080/open/v1/dot1dtpconfig
```

browser

```
http://10.10.10.1:8080/open/v1/dot1dtpconfig?method=  
put&agingTime=60
```

Response Example

```
{  
  "status":  
  {  
    "response_code": 200,  
    "msg": "OK",  
  }  
}
```

```

        "more_info": "",
        "error_code": 0
    }
}

```

Error Example

curl -GET -XPUT 10.10.10.1:8080/open/v1/dot1dtpconfig
 browser
<http://10.10.10.1:8080/open/v1/dot1dtpconfig?method=put>

Response Example

```

{
    "status":
    {
        "response_code": 400,
        "msg": "Bad Request",
        "more_info": "Invalid parameter"
        "error_code": -22
    }
}

```

dot1dtpconfig DELETE

This component does not support delete semantics.

dot1dtpfdbentries

dot1dtpfdbentries implements a RESTful API for the OpEN RFC1493 API. It focuses on the *dot1dBridge* object defined by that RFC.

dot1dtpfdbentries CREATE

This component does not support create semantics.

dot1dtpfdbentries READ

`GET /open/v1/dot1dtpfdbentries`

Retrieves the various attributes supported by this resource.

Request Details

Table 4-16. dot1dtpfdbentries READ request Details

Parameter	Type	Description
fdbAddress	string	MAC address.
fields	—	Partial response syntax.
fields	port	Port number.
—	fdbStatus	The status of the dot1dTpFdbEntry.
offset	integer	Denotes beginning index for pagination. Default is 1. This parameter is not valid for single entry retrievals.
limit	integer	Denotes total retrieval count. Default is 10. This parameter is not valid for single entry retrievals.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-17. dot1dtpfdbentries READ Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Request failed.

Multiple dot1dtpfdbentries Request Example

The following example shows a multiple dot1dtpfdbentries retrieval. Note that pagination is implicit.

```
curl -GET 10.10.10.1:8080/open/v1/dot1dtpfdbentries
browser
http://10.10.10.1:8080/open/v1/dot1dtpfdbentries
```

Response Example

```
{
  "dot1dTpFdbEntries": [
    {
      "fdbAddress": "11:22:33:44:55:66:11:11",
      "port": "0/1",
      "fdbStatus": "learned"
    },
    {
      "fdbAddress": "11:22:33:44:55:77:11:11",
      "port": "0/2",
      "fdbStatus": "learned"
    },
    {
      "fdbAddress": "11:22:33:44:55:88:11:11",
      "port": "0/3",
      "fdbStatus": "learned"
    }
  ],
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

```
}
```

Single dot1dtpfdbentries Partial Request Example

```
curl -GET -d "fdbAddress=11:22:33:44:55:66:11:11&fields=port"  
10.10.10.1:8080/open/v1/dot1dtpfdbentries
```

browser

```
http://10.10.10.1:8080/open/v1/dot1dtpfdbentries?fdbAddress=  
11:22:33:44:55:66:11:11&fields=port
```

Response Example

```
{  
  "dot1dtpfdbentries": [  
    {  
      "fdbAddress": "11:22:33:44:55:66:11:11",  
      "port": "0/1"  
    }  
  ],  
  "status": {  
    "response_code": 200,  
    "msg": "OK",  
    "more_info": "",  
    "error_code": 0  
  }  
}
```

Error Example

```
curl -GET -d "index=-99"  
10.10.10.1:8080/open/v1/dot1dtpfdbentries
```

browser

```
http://10.10.10.1:8080/open/v1/dot1dtpfdbentries?method=  
get&index=-99
```

Response Example

Returns an appropriate status message.

```
{  
  "status":  
    {  
      "response_code": 400,  
      "msg": "Bad Request",  
      "more_info": "Invalid parameter",  
      "error_code": -22  
    }  
}
```

```
    }  
}
```

dot1dtpfdbentries UPDATE

This component does not support update semantics.

dot1dtpfdbentries DELETE

This component does not support delete semantics.

dot1dtpportentries

dot1dtpportentries implements a RESTful API for the OpEN RFC1493 API. It focuses on the *dot1dBridge* object defined by that RFC.

dot1dtpportentries CREATE

This component does not support create semantics.

dot1dtpportentries READ

`GET /open/v1/dot1dtpportentries`

Retrieves the various attributes supported by this resource.

Request Details

Table 4-18. dot1dtpportentries READ Request Details

Parameter	Type	Description
port	string	Interface number.
fields	—	Partial response syntax.
fields	maxInfo	Maximum size of the INFO field this port will receive or transmit.
—	inFrames	Number of frames received by this port from its segment.
—	outFrames	Number of frames transmitted by this port to its segment.
—	inDiscards	Count of valid frames received which were discarded by the Forwarding Process.
offset	integer	Denotes beginning index for pagination. Default is 1. This parameter is not valid for single entry retrievals.
limit	integer	Denotes total retrieval count. Default is 10. This parameter is not valid for single entry retrievals.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-19. dot1dtpportentries READ Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Request failed.

Multiple dot1dtpportentries Request Example

The following example shows a multiple dot1dtpportentries retrieval. Note that pagination is implicit.

```
curl -GET 10.10.10.1:8080/open/v1/dot1dtpportentries
browser
http://10.10.10.1:8080/open/v1/dot1dtpportentries
```

Response Example

```
{
  "dot1dTpPortEntries": [
    {
      "port": "0/1",
      "maxInfo": 255,
      "inFrames": 13,
      "outFrames": 31,
      "inDiscards": 0
    },
    {
      "port": "0/2",
      "maxInfo": 255,
      "inFrames": 15,
      "outFrames": 41,
      "inDiscards": 0
    },
    {
      "port": "0/3",
      "maxInfo": 255,
      "inFrames": 17,
      "outFrames": 51,
```

```

        "inDiscards": 1
    },
    "status": {
        "response_code": 200,
        "msg": "OK",
        "more_info": "",
        "error_code": 0
    }
}

```

Single dot1dtpportentries Partial Request Example

```

curl -GET -d "port=0/1&fields=inFrames,outFrames"
10.10.10.1:8080/open/v1/dot1dtpportentries

```

browser

```

http://10.10.10.1:8080/open/v1/dot1dtpportentries?port=
0/1&fields=inFrames,outFrames

```

Response Example

```

{
    "dot1dTpPortEntries": [
        {
            "port": "0/1",
            "maxInfo": 255,
            "inFrames": 13,
            "outFrames": 31,
            "inDiscards": 0
        }
    ],
    "status": {
        "response_code": 200,
        "msg": "OK",
        "more_info": "",
        "error_code": 0
    }
}

```

Error Example

```

curl -GET -d "index=-99"
10.10.10.1:8080/open/v1/dot1dtpportentries

```

browser

```
http://10.10.10.1:8080/open/v1/dot1dtpportentries?method=
get&index=-99
```

Response Example

Returns an appropriate status message.

```
{
  "status":
  {
    "response_code": 400,
    "msg": "Bad Request",
    "more_info": "Invalid parameter",
    "error_code": -22
  }
}
```

dot1dtpportentries UPDATE

This component does not support update semantics.

dot1dtpportentries DELETE

This component does not support delete semantics.

dot1sconfig

STP Config CREATE

This component does not support create semantics.

STP Config READ

```
GET /open/v1/dot1sconfig
```

Retrieves STP configuration.

Request Details

Table 4-20. STP Config READ Request Details

Parameter	Type	Description
fields	—	Partial response syntax.
fields	mode	Spanning Tree Operational mode.
—	configName	MSTP Configuration Name.
—	configRevision	MSTP Configuration revision level.
—	forceVersion	Value of the MSTP Force Protocol Version parameter (802.1d, 802.1w, 802.1s).
—	bpduGuardMode	Spanning Tree BPDU Guard Mode.
—	bpduFilterMode	Spanning Tree BPDU Filter Mode.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-21. dot1sconfig READ Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Operation Fail.

STP Config Request Example

The following example shows STP parameters retrieval.

```
curl -GET 10.10.10.1:8080/open/v1/dot1sconfig
browser
http://10.10.10.1:8080/open/v1/dot1sconfig
```

Response Example

```
{
  "dot1sconfig":
  {
    "mode": "enabled",
    "forceVersion": "802.1s",
    "configName": "20-0C-C8-4D-95-D8",
    "configRevision": 0,
    "bpduGuardMode": "disabled",
    "bpduFilterMode": "disabled",
  }
  "status":
  {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Error Request Example

The following example shows the request with invalid parameter.

```
curl -GET "vlanId=1" 10.10.10.1:8080/open/v1/dot1sconfig
```

```
browser
  http://10.10.10.1:8080/open/v1/dot1sconfig?vlanId=1
```

Response Example

```
{
  "status":
  {
    "response_code": 400,
    "msg": "Bad Request",
    "more_info": "Invalid parameter",
    "error_code": -22
  }
}
```

STP Config UPDATE

```
PUT /open/v1/dot1sconfig
```

Use this call to update STP Configuration on the switch.

Request Details

Table 4-22. STP Config UPDATE Request Details

Parameter	Type	Description
mode	string	Spanning Tree Operational Mode (enabled, disabled).
configName	string	MSTP Configuration Name.
configRevision	integer	MSTP Configuration revision level.
forceVersion	string	Value of the MSTP Force Protocol Version parameter. (802.1d, 802.1w, 802.1s)
bpduGuardMode	string	Spanning Tree BPDU Guard Mode (enabled, disabled).
bpduFilterMode	string	Spanning Tree BPDU Filter Mode (enabled, disabled).

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-23. STP Config UPDATE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Operation Fail.

Update STP Config Request Example

```
curl -GET -XPUT -d "mode=enabled&forceVersion=802.1s"
10.10.10.1:8080/open/v1/dot1sconfig
```

browser

```
http://10.10.10.1:8080/open/v1/dot1sconfig?method=put&mode=
enabled&forceVersion=802.1s
```

Response Example

```
{
  "status":
  {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Update STP Config Error Request Example

```
curl -GET -XPUT -d "forceVersion=none"
10.10.10.1:8080/open/v1/dot1sconfig
```

browser

```
http://10.10.10.1:8080/open/v1/dot1sconfig?method=
put&forceVersion=none
```

Response Example

```
{
  "status":
  {
    "response_code": 400,
    "msg": "Bad Request",
    "more_info": "Invalid parameter",
    "error_code": -22
  }
}
```

STP Config DELETE

This component does not support delete semantics.

dot1sinterfaces

dot1sinterfaces CREATE

dot1sinterfaces does not support CREATE semantics.

dot1sinterfaces READ

```
GET /open/v1/dot1sinterfaces
```

Retrieves dot1sinterfaces parameters.

Request Details

Table 4-24. dot1sinterfaces READ Request Details

Parameter	Type	Description
interface	string	Interface name in slot/port format (optional).
mstId	integer	MSTI id. Used only for ancillary request (intfPortFwdState, intfPortRole).
fields	—	Partial response syntax.
fields	bpduFilterMode	Spanning Tree BPDU Filter mode for an interface (enabled, disabled).
—	bpduFloodMode	Spanning Tree BPDU Flood mode for an interface (enabled, disabled).
—	intfEdgePortMode	MSTP Administrative Edge Port mode of an interface (enabled, disabled).
—	intfGuardMode	Spanning Tree Guard mode (root, loop, none).
—	intfMode	Spanning Tree Admin Mode on an interface (enabled, disabled).
—	intfPortFwdState	Ancillary request parameter to determine spanning Tree Port Forwarding State. Use only with mstId parameter (discarding, learning, forwarding, disabled, manual_forwarding, not_participate).
—	intfPortRole	Ancillary request parameter to determine spanning Tree Port Role. Use only with mstId parameter (disabled, root, designed, alternate, backup, master).

Table 4-24. dot1sinterfaces READ Request Details (Continued)

Parameter	Type	Description
offset	integer	Denotes beginning index for pagination. Default is 1. This parameter is not valid for single STP retrievals.
limit	integer	Denotes total retrieval count. Default is 10. This parameter is not valid for single STP retrievals.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-25. dot1sinterfaces READ Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Operation Fail.

Multiple dot1sinterfaces Request Example

The following example shows a dot1sinterfaces parameters retrieval.

```
curl -GET 10.10.10.1:8080/open/v1/dot1sinterfaces
browser
http://10.10.10.1:8080/open/v1/dot1sinterfaces
```

Response Example

```
{
  "dot1sinterfaces": [
    {
      "interface": "0/1",
```

```

        "bpduFilterMode": "disabled",
        "bpduFloodMode": "disabled",
        "intfEdgePortMode": "disabled",
        "intfGuardMode": "loop",
        "intfMode": "enabled"
    },
    {
        "interface": "0/2",
        "bpduFilterMode": "enabled",
        "bpduFloodMode": "disabled",
        "intfEdgePortMode": "disabled",
        "intfGuardMode": "loop",
        "intfMode": "enabled"
    },
    {
        "interface": "0/3",
        "bpduFilterMode": "disabled",
        "bpduFloodMode": "enabled",
        "intfEdgePortMode": "enabled",
        "intfGuardMode": "loop",
        "intfMode": "enabled"
    }
],
"status":
{
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
}
}

```

Multiple dot1sinterfaces Partial Request Example

An example of a request that limits the output to the first two dot1sinterfaces entries.

```

curl -GET -d "limit=2" 10.10.10.1:8080/open/v1/dot1sinterfaces
browser
http://10.10.10.1:8080/open/v1/dot1sinterfaces?limit=2

```

Response Example

```

{

```

```

"dot1sinterfaces": [
  {
    "interface": "0/1",
    "bpduFilterMode": "disabled",
    "bpduFloodMode": "disabled",
    "intfEdgePortMode": "disabled",
    "intfGuardMode": "loop",
    "intfMode": "enabled"
  },
  {
    "interface": "0/2",
    "bpduFilterMode": "enabled",
    "bpduFloodMode": "disabled",
    "intfEdgePortMode": "disabled",
    "intfGuardMode": "loop",
    "intfMode": "enabled"
  }
],
"status":
{
  "response_code": 200,
  "msg": "OK",
  "more_info": "",
  "error_code": 0
}
}

```

Single dot1sinterfaces Request Example

Single dot1sinterfaces retrievals are specified by using the interface parameter. All the available dot1sinterfaces resources are retrieved by default.

The following example shows a specific dot1sinterfaces request for interface 0/10:

```

curl -GET -d "interface=0/10"
10.10.10.1:8080/open/v1/dot1sinterfaces

```

browser

```

http://10.10.10.1:8080/open/v1/dot1sinterfaces?interface=0/10

```

Response Example

```

{
  "dot1sinterfaces": [
    {
      "interface": "0/10",

```

```

        "bpduFilterMode": "disabled",
        "bpduFloodMode": "disabled",
        "intfEdgePortMode": "disabled",
        "intfGuardMode": "loop",
        "intfMode": "enabled"
    },
    ],
    "status":
    {
        "response_code": 200,
        "msg": "OK",
        "more_info": "",
        "error_code": 0
    }
}

```

Single dot1sinterfaces Partial Request Example

The response can be reduced by using the partial response syntax as denoted by the *fields* parameter.

```

curl -GET -d "interface=0/10&fields=
intfPortFwdState,intfPortRole"
10.10.10.1:8080/open/v1/dot1sinterfaces

```

browser

```

http://10.10.10.1:8080/open/v1/dot1sinterfaces?interface=
0/10&fields=intfPortFwdState,intfPortRole&mstId=0

```

Response Example

```

{
    "dot1sinterfaces": [
        {
            "interface": "0/10",
            "mstId": 0,
            "intfPortFwdState": "disabled"
            "intfPortRole": "disabled",
        }
    ],
    "status":
    {
        "response_code": 200,
        "msg": "OK",
        "more_info": "",
    }
}

```

```

        "error_code": 0
    }
}

```

Ancillary Single Request Example

An example using an ancillary request parameter to determine Port Forwarding State and Port Role.

```

curl -GET -d "interface=0/10&mstId=1&fields=
intfPortFwdState,intfPortRole"
10.10.10.1:8080/open/v1/dot1sinterfaces

```

browser

```

http://10.10.10.1:8080/open/v1/dot1sinterfaces?interface=
0/10&mstId=0&fields=intfPortFwdState,intfPortRole

```

Response Example

```

{
  "dot1intfs": [
    {
      "interface": "0/10",
      "mstId": 0,
      "intfPortFwdState": "disabled",
      "intfPortRole": "disabled"
    }
  ],
  "status":
  {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}

```

Error Request Example

The following example shows the request with an invalid parameter.

```

curl -GET "vlanId=1" 10.10.10.1:8080/open/v1/dot1sinterfaces

```

browser

```

http://10.10.10.1:8080/open/v1/dot1sinterfaces?vlanId=1

```

Response Example

```
{
  "status":
  {
    "response_code": 400,
    "msg": "Bad Request",
    "more_info": "Invalid parameter",
    "error_code": -22
  }
}
```

dot1sinterfaces UPDATE

```
PUT /open/v1/dot1sinterfaces
```

Use this call to update an existing dot1sinterfaces parameters on the switch. The dot1sinterfaces parameters name is mandatory.

Request Details

Table 4-26. dot1sinterfaces UPDATE Request Details

Parameter	Type	Description
interface	string	Interface name in slot/port format. Required.
bpduFilterMode	string	Spanning Tree BPDU Filter mode for an interface (enabled, disabled).
bpduFloodMode	string	Spanning Tree BPDU Flood mode for an interface (enabled, disabled).
intfEdgePortMode	string	MSTP Administrative Edge Port mode of an interface (enabled, disabled).
intfGuardMode	string	Spanning Tree Guard mode (root, loop, none).
intfMode	string	Spanning Tree Admin Mode on an interface (enabled, disabled).

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-27. dot1sinterfaces UPDATE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Operation Fail.

Update dot1sinterfaces Request Example

The following example updates the Spanning Tree BPDU Filter mode and Gets Spanning Tree Guard mode for an interface located in slot 0, port 1.

```
curl -GET -XPUT -d "interface=0/1&bpduFilterMode=
enabled&intfGuardMode=loop"
10.10.10.1:8080/open/v1/dot1sinterfaces
```

browser

```
http://10.10.10.1:8080/open/v1/dot1sinterfaces?method=
put&interface=0/1&bpduFilterMode=enabled&intfGuardMode=loop
```

Response Example

```
{
  "status":
  {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

dot1sinterfaces Error Request Example

The following example shows the request with an invalid parameter.

```
curl -GET -XPUT -d "forceVersion=802.1s"
10.10.10.1:8080/open/v1/dot1sinterfaces
```

browser

```
http://10.10.10.1:8080/open/v1/dotlsinterfaces?method=
put&forceVersion=802.1s
```

Response Example

```
{
  "status":
  {
    "response_code": 400,
    "msg": "Bad Request",
    "more_info": "Invalid parameter",
    "error_code": -22
  }
}
```

INTERFACE DELETE

dotlsinterfaces does not support DELETE semantics.

dot1smstientries

MSTI CREATE

POST /open/v1/dot1smstientries

Use this call to create a MST instance with the ID specified on the switch. The mstId ID is mandatory.

Request Details

Table 4-28. MSTI CREATE Request Details

Parameter	Type	Description
mstId	integer	The MSTI ID. (0 is not allowed as an MSTI ID.)

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, 409, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-29. MSTI CREATE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Operation Fail.

Request Example

```
curl -POST -d "mstId=2"  
10.10.10.1:8080/open/v1/dot1smstientries
```

browser

```
http://10.10.10.1:8080/open/v1/dotlsmstientries?method=
post&mstId=2
```

Response Example

Returns an appropriate status message.

```
{
  "status":
  {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Error Example

```
curl -POST -d "mstId=2"
10.10.10.1:8080/open/v1/dotlsmstientries
```

browser

```
http://10.10.10.1:8080/open/v1/dotlsmstientries?method=
post&mstId=0
```

Response Example

Returns an appropriate status message.

```
{
  "status":
  {
    "response_code": 500,
    "msg": "Fail",
    "more_info": "Operation failed",
    "error_code": "-26"
  }
}
```

MST READ

```
GET /open/v1/dotlsmstientries
```

Retrieves dot1smstientries parameters.

Request Details

Table 4-30. MST READ Request Details

Parameter	Type	Description
mstId	integer	MSTI ID.
fields	—	Partial response syntax.
fields	mstId	Spanning Tree MST Instance ID.
—	fdbIds	Forwarding Database IDs for the Spanning Tree MST Instance.
—	vlan	VLAN for the Spanning Tree MST Instance. Fields: • id - VLAN ID. • type - Type of VLAN (default, static, dynamic, unconfigured). • name - User specified VLAN's name or <i>unconfigured</i>.
—	priority	Spanning Tree MST Instance priority.
—	vlanMax	Ancillary request parameter to determine the maximum number of Spanning Tree VLANs.
offset	integer	Denotes beginning index for pagination. Default is 1. This parameter is not valid for single STP retrievals.
limit	integer	Denotes total retrieval count. Default is 10. This parameter is not valid for single STP retrievals.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-31. MST READ Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Operation Fail.

Multiple MST Entry Request Example

The following example shows a multiple MST entry retrieval. Note that pagination is implicit.

```
curl -GET -d 10.10.10.1:8080/open/v1/dot1smstentries
browser
http://10.10.10.1:8080/open/v1/dot1smstentries
```

Response Example

```
{
  "dot1smstentries" : [
    {
      "mstId" : 1,
      "fdbIds" : [
        1,
        2,
      ],
      "vlans" : [
        {
          "id": 1,
          "type": "default",
          "name": "vlan1"
        },
        {
          "id": 2,
          "type": "unconfigured",
          "name": "unconfigured"
        }
      ],
      "priority" : 1,
    },
  ],
}
```

```

{
  "mstId" : 2,
  "fdbIds": [
    3,
    4,
  ],
  "vlans": [
    {
      "id": 3,
      "type": "static",
      "name": "vlan3"
    }
  ],
  "priority" : 2,
},
{
  "mstId" : 3,
  "fdbIds": [
    5,
    6,
  ],
  "vlans": [
    {
      "id": 4,
      "type": "dynamic",
      "name": "vlan4"
    }
  ],
  "priority" : 3,
}
],
"status":
{
  "response_code": 200,
  "msg": "OK",
  "more_info": "",
  "error_code": 0
}
}

```

Multiple MST Entries Partial Request Example

An example of a request that limits the size of the output to 2.

```

curl -GET -d "limit=2"
10.10.10.1:8080/open/v1/dot1smstientries

```

browser

`http://10.10.10.1:8080/open/v1/dot1smstientries?limit=2`

Response Example

```
{
  "dot1smstientries" : [
    {
      "mstId" : 1,
      "fdbIds" : [
        1,
        2,
      ],
      "vlans" : [
        {
          "id": 1,
          "type": "default",
          "name": "vlan1"
        },
        {
          "id": 2,
          "type": "default",
          "name": "vlan2"
        }
      ],
      "priority" : 1,
    },
    {
      "mstId" : 2,
      "fdbIds" : [
        3,
        4,
      ],
      "vlans" : [
        {
          "id": 3,
          "type": "default",
          "name": "vlan3"
        }
      ],
      "priority" : 2,
    }
  ],
  "status":
  {
```

```

        "response_code": 200,
        "msg": "OK",
        "more_info": "",
        "error_code": 0
    }
}

```

Single MST Entry Request Example

Single MST entry retrievals are specified by using the mstId of the forwarding entry. All the available data for the dot1smst entry are retrieved by default.

```

curl -GET -d "mstId=1"
10.10.10.1:8080/open/v1/dot1smstentries

```

browser

```

http://10.10.10.1:8080/open/v1/dot1smstentries?mstId=1

```

Response Example

```

{
  "dot1smstentries" : [
    {
      "mstId" : 1,
      "fdbIds": [
        1,
        2,
      ],
      "vlans": [
        {
          "id": 1,
          "type": "default",
          "name": "default"
        }
      ],
      "priority" : 1,
    }
  ],
  "status":
  {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}

```

```
}
```

Error Request Example

The following example shows the request with invalid parameter.

```
curl -GET -d "vlanId=2"  
10.10.10.1:8080/open/v1/dot1smstientries
```

browser

```
http://10.10.10.1:8080/open/v1/dot1smstientries?vlanId=2
```

Response Example

```
{  
  "status":  
  {  
    "response_code": 400,  
    "msg": "Bad Request",  
    "more_info": "Invalid parameter",  
    "error_code": -22  
  }  
}
```

MSTI UPDATE

```
PUT /open/v1/dot1smstientries
```

Use this call to update an existing MSTI on the switch. The MSTI ID is mandatory.

Request Details

Table 4-32. MSTI UPDATE Request Details

Parameter	Type	Description
mstId	integer	Spanning Tree MST Instance ID.
vlanId	integer	VLAN Id for the Spanning Tree MST Instance.
action	string	Action to be used for VLAN. Used only with vlanId (<i>enable, disable</i>).
priority	integer	Spanning Tree MST Instance priority.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, 404, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-33. MSTI UPDATE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Operation Fail.

MST Request Example

```
curl -GET -XPUT -d "mstId=1&vlanId=1&action=disable"
10.10.10.1:8080/open/v1/dot1smstientries

browser

http://10.10.10.1:8080/open/v1/dot1smstientries?method=
put&mstId=1&vlanId=1&action=disable
```

Response Example

```
{
  "status":
  {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

MSTI Error Example

```
curl -GET -XPUT -d "mstId=1&action=enable"
10.10.10.1:8080/open/v1/dot1smstientries
```

browser

```
http://10.10.10.1:8080/open/v1/dot1smstientries?method=
put&mstId=1&action=enable
```

Response Example

```
{
  "status":
  {
    "response_code": 400,
    "msg": "Bad request",
    "more_info": "Invalid parameter",
    "error_code": -22
  }
}
```

MSTI DELETE

```
DELETE /open/v1/dot1smstientries
```

Use this call to delete an existing MST instance on the switch. The MST ID is mandatory.

Request Details

Table 4-34. MSTI DELETE Request Details

Parameter	Type	Description
mstId	integer	MSTI ID.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-35. MSTI DELETE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Operation Fail.

Request Example

```
curl -GET -XDELETE -d "mstId=2"  
10.10.10.1:8080/open/v1/dot1smstientries
```

browser

```
http://10.10.10.1:8080/open/v1/dot1smstientries?method=  
delete&mstId=2
```

Response Example

```
{  
  "status":  
  {  
    "response_code": 200,  
    "msg": "OK",  
    "more_info": "",  
    "error_code": 0  
  }  
}
```

Error Request Example

```
curl -GET -XDELETE -d "vlanId=2"  
10.10.10.1:8080/open/v1/dot1smstientries
```

browser

```
http://10.10.10.1:8080/open/v1/dot1smstientries?method=  
delete&vlanId=2
```

Response Example

```
{  
  "status":
```

```

    {
        "response_code": 400,
        "msg": "Bad request",
        "more_info": "Invalid parameter",
        "error_code": -22
    }
}

```

fdb

FDB CREATE

This component does not support create semantics.

FDB READ

```
GET /open/v1/fdb
```

Retrieves all of the variables that are associated with the FDB object.

Request Details

Table 4-36. FDB READ Request Details

Parameter	Type	Description
mac	string	MAC address of the entry (requires <i>vlanId</i>).
vlanId	integer	VLAN ID of the entry.
interface	string	Interface of the entry (slot/port).
fields	—	Partial response syntax.
—	interface	Interface number of the entry.
—	vlanId	VLAN ID of the entry.
—	mac	MAC address of the entry.
—	entryType	Type of the entry.
offset	integer	Denotes beginning index for pagination. Default is 1. This parameter is not valid for single FDB entry retrievals.
limit	integer	Denotes total retrieval count. Default is 10. This parameter is not valid for single FDB entry retrievals.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-37. FDB READ Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-22	OPEN_E_PARAM	Invalid request.
-26	OPEN_E_FAIL	Operation failed.

Multiple FDB Entries Request Example

```
curl -GET 10.10.10.1:8080/open/v1/fdbs
browser
http://10.10.10.1:8080/open/v1/fdbs
```

Response Example

```
{
  "fdbEntries": [
    {
      "interface": "1/1",
      "vlanId": 1,
      "mac": "00:11:22:33:44:55",
      "entryType": "learned"
    },
    {
      "interface": "1/2",
      "vlanId": 2,
      "mac": "00:11:22:33:44:66",
      "entryType": "learned"
    },
    {
      "interface": "1/3",
      "vlanId": 3,
      "mac": "00:11:22:33:44:77",

```

```

        "entryType": "learned"
    },
],
"status":
{
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
}
}

```

Multiple FDB Entries Partial Request Example

```

curl -GET -d "limit=2" 10.10.10.1:8080/open/v1/fdb
browser
http://10.10.10.1:8080/open/v1/fdb?limit=2

```

Response Example

```

{
    "fdbEntries": [
    {
        "interface": "1/1",
        "vlanId": 1,
        "mac": "00:11:22:33:44:55",
        "entryType": "learned"
    },
    {
        "interface": "1/2",
        "vlanId": 2,
        "mac": "00:11:22:33:44:66",
        "entryType": "learned"
    }
    ],
    "status":
    {
        "response_code": 200,
        "msg": "OK",
        "more_info": "",
        "error_code": 0
    }
}

```

Single FDB Entry Partial Request Example

```
curl -GET -d "mac=00:11:22:33:44:55&vlanId=1"  
10.10.10.1:8080/open/v1/fdbs
```

browser

```
http://10.10.10.1:8080/open/v1/fdbs?mac=  
00:11:22:33:44:55&vlanId=1
```

Response Example

```
{  
  "fdbEntries": [  
    {  
      "interface": "1/1",  
      "vlanId": 1,  
      "mac": "00:11:22:33:44:55",  
      "entryType": "learned"  
    }  
  ],  
  "status":  
    {  
      "response_code": 200,  
      "msg": "OK",  
      "more_info": "",  
      "error_code": 0  
    }  
}
```

Error Example

```
curl -GET -d "x=-99" 10.10.10.1:8080/open/v1/fdbs
```

browser

```
http://10.10.10.1:8080/open/v1/fdbs?method=get&x=-99
```

Response Example

Returns an appropriate status message.

```
{  
  "status":  
    {  
      "response_code": 400,  
      "msg": "Bad Request",  
      "error_code": 0  
    }  
}
```

```

    "more_info": "Invalid arguments",
    "error_code": -22
  }
}

```

FDB UPDATE

This component does not support delete semantics.

FDB DELETE

```
DELETE /open/v1/fdbs
```

Use this call to delete an existing FDB entry on the switch. Only one of parameters (*mac*, *vlanId*, *interface*, *all*) is allowed to use per particular request.

Request Details

Table 4-38. FDB DELETE Request Details

Parameter	Type	Description
mac	string	Delete all learned entries in FDB table with the specified MAC.
vlanId	integer	Delete all learned entries in FDB table with the specified VLAN ID.
interface	string	Delete all learned entries in FDB table with the specified interface number (slot/port).
all	—	Delete all learned entries in FDB table.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, 404, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-39. FDB DELETE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Other internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Operation Fail.

Request Example

```
curl -GET -XDELETE -d "mac=00:11:22:33:44:55"  
10.10.10.1:8080/open/v1/fdbs
```

browser

```
http://10.10.10.1:8080/open/v1/fdbs?method=delete&mac=  
00:11:22:33:44:55
```

Response Example

```
{  
  "status":  
  {  
    "response_code": 200,  
    "msg": "OK",  
    "more_info": "",  
    "error_code": 0  
  }  
}
```

Error Request Example

```
curl -GET -XDELETE -d "vlanId=1&interface=1/2"  
10.10.10.1:8080/open/v1/fdbs
```

browser

```
http://10.10.10.1:8080/open/v1/fdbs?method=delete&vlanId=  
1&interface=1/2
```

Response Example

```
{  
  "status":
```

```

    {
      "response_code": 400,
      "msg": "Bad Request",
      "more_info": "Invalid parameter",
      "error_code": -22
    }
  }
}

```

fdbstats

fdbstats CREATE

This component does not support create semantics.

fdbstats READ

```
GET /open/v1/fdbstats
```

Retrieves all of the variables that are associated with the FDB statistics.

Request Details

Table 4-40. fdbstats READ Request Details

Parameter	Type	Description
fields	—	Partial response syntax.
—	dynamicEntries	Count of the dynamic entries in the FDB table.
—	staticEntries	Count of the static entries in the FDB table.
—	maxTableEntries	Max number of the entries FDB table can hold.
—	currentTableEntries	Current number of the entries in the FDB table.
—	greatestTableEntries	Greatest number of the entries FDB table had held.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-41. fdbstats READ Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-22	OPEN_E_PARAM	Invalid request.
-26	OPEN_E_FAIL	Operation failed.

Request Example

```
curl -GET 10.10.10.1:8080/open/v1/fdbstats
browser
http://10.10.10.1:8080/open/v1/fdbstats
```

Response Example

```
{
  "fdbstats":
  {
    "staticEntries": 8,
    "dynamicEntries": 2,
    "maxTableEntries": 131072,
    "currentTableEntries": 5,
    "greatestTableEntries": 12
  },
  "status":
  {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Full Request Example

Use additional keyword for additional output.

```
curl -GET "fields=staticEntries,dynamicEntries"  
10.10.10.1:8080/open/v1/fdbstats
```

browser

```
http://10.10.10.1:8080/open/v1/fdbstats?fields=  
staticEntries,dynamicEntries
```

Response Example

```
{  
  "fdbstats":  
  {  
    "staticEntries": 8,  
    "dynamicEntries": 2  
  },  
  "status":  
  {  
    "response_code": 200,  
    "msg": "OK",  
    "more_info": "",  
    "error_code": 0  
  }  
}
```

Error Example

```
curl -GET -d "x=-99" 10.10.10.1:8080/open/v1/fdbstats
```

browser

```
http://10.10.10.1:8080/open/v1/fdbstats?method=get&x=-99
```

Response Example

Returns an appropriate status message.

```
{  
  "status":  
  {  
    "response_code": 400,  
    "msg": "Bad Request",  
    "more_info": "Invalid arguments",  
  }  
}
```

```
    "error_code": -22
  }
}
```

fdbstats UPDATE

PUT /open/v1/fdbstats

Use this call to update attributes of a fdbstats object.

Request Details

Table 4-42. fdbstats UPDATE Request Details

Parameter	Type	Description
greatestEntriesReset	—	Reset the greatest number of entries in the forwarding database.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-43. fdbstats UPDATE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Update successful.
-22	OPEN_E_PARAM	Error in parameters passed, or none of the optional parameters was specified.
-26	OPEN_E_FAIL	Update failed.

Request Example

```
curl -GET -XPUT -d "greatestEntriesReset=true"
10.10.10.1:8080/open/v1/fdbstats
```

browser

```
http://10.10.10.1:8080/open/v1/fdbstats?method=
put&greatestEntriesReset=true
```

Response Example

```
{
  "status":
  {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Error Example

```
curl -GET -XPUT -d "test=60" 10.10.10.1:8080/open/v1/fdbstats
browser
http://10.10.10.1:8080/open/v1/fdbstats?method=put&test=60
```

Response Example

```
{
  "status":
  {
    "response_code": 400,
    "msg": "Bad Request",
    "more_info": "Invalid arguments",
    "error_code": -22
  }
}
```

fdbstats DELETE

This component does not support delete semantics.

Switching

Interfaces

Interface CREATE

NOTE: Interfaces cannot be created. However, you can reconfigure or UPDATE their parameters using PUT commands.

Interface READ

```
GET /open/v1/interfaces
```

Retrieves one or more available interfaces.

Request Details

Table 4-44. Interface READ Request Details

Parameter	Type	Description
interface	slot/port	Interface name. Identifies the port or LAG. Interface name is mandatory.
vlan	integer	VLAN ID (optional).
fields	—	Partial response syntax.
—	acceptFrame	Indicates how the interface handles untagged and priority tagged frames.
—	adminMode	The administrative state of the interface, which is either enabled or disabled.
—	autoNegotiate	Indicates the auto-negotiation mode for physical interfaces. The mode is reported as unknown for LAGs and VLANs.
—	description	The current description, if any, associated with the interface to help identify it.
—	duplexMode	Indicates the port duplex mode for physical interfaces. The mode is reported as unknown for LAGs and VLANs.

Table 4-44. Interface READ Request Details (Continued)

Parameter	Type	Description
—	linkState	The link state of the interface, which is either enabled or disabled. An interface is considered active if the link is up, and the interface is in a forwarding state.
—	mtuSize	Indicates Maximum Transmission Unit (MTU) of the interface. The actual frame size is calculated by adding Ethernet header size in MTU.
—	participation	Participation mode is shown as 'included', 'excluded', or 'auto' for the VLAN specified by the vlan argument. If the vlan argument is omitted, then it is an array of VLANs that this interface participates in.
—	physicalAddress	The burned-in physical address of the interface. The format is six two-digit hexadecimal numbers separated by colons, for example 00:06:29:32:81:40.
—	pVid	If the given interface is assigned as the port VLAN ID.
—	speed	The physical link data rate of the interface.
—	tagging	Identified as 'tagged' or 'untagged' for the VLAN specified by the vlan argument. If vlan is not specified, it is an array of VLANs that are tagged for this interface.
—	type	Interface type, such as physical, VLAN, LAG, and so on.
—	lastChange	The time of last link state change in seconds.
—	inOctets	The total number of octets received on the interface including framing characters.
—	inUcastPkts	The number of packets, delivered by this sub-layer to a higher sub-layer.
—	inDiscards	The number of inbound packets which were chosen to be discarded.
—	inErrors	The current value of the errors counter.
—	inUnknownProtos	The number of packets received via the interface which were discarded because of an unknown or unsupported protocol.

Table 4-44. Interface READ Request Details (Continued)

Parameter	Type	Description
—	outOctets	The total number of octets transmitted out of the interface, including framing characters.
—	outUcastPkts	The total number of packets that higher-level protocols requested be transmitted.
—	outDiscards	The number of outbound packets which were chosen to be discarded.
—	outErrors	For packet-oriented interfaces, the number of outbound packets that could not be transmitted because of errors.
—	interfaceMax	Ancillary request parameter to get the maximum number of interfaces possible for the given platform, i.e., CPU slots, LAG, VLAN, Loopback, Tunnel, Stack, and Service Port interfaces.
—	descriptionMax	Ancillary request parameter to get the maximum number of characters allowed for this interface description.
offset	integer	Denotes beginning index for pagination. Default is 1. This parameter is not valid for single interface retrievals.
limit	integer	Denotes total retrieval count. Default is 10. This parameter is not valid for single interface retrievals.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, 404, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-45. Interface READ Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.

Table 4-45. Interface READ Internal Error Codes (Continued)

Error Code	Code Name	Description
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Operation Fail.
-29	OPEN_E_NOT_FOUND	Parameter not found.

Multiple Interface Request Example

The following example shows a multiple interface retrieval:

```
curl -GET 10.10.10.1:8080/open/v1/interfaces
browser
http://10.10.10.1:8080/open/v1/interfaces
```

NOTE: Pagination is implicit.

Response Example

```
{
  "interfaces": [
    {
      "adminMode": "enabled",
      "autoNegotiate": "enabled",
      "description": "",
      "physicalAddress": "00:01:85:48:E0:11",
      "duplexMode": "half",
      "pvId": 1,
      "type": "physical",
      "linkState": "down",
      "acceptFrame": "all",
      "speed": "unknown",
      "mtuSize": 1500,
      "interface": "0/1",
      "lastChange": 1301,
      "inOctets": 3,
      "inUcastPkts": 2,
      "inDiscards": 1,
```

```

    "inErrors": 0,
    "inUnknownProtos": 0,
    "outOctets": 3,
    "outUcastPkts": 2,
    "outDiscards": 0,
    "outErrors": 0,
    "tagging": [1, 70],
    "participation" : [1, 70],

  },
  {
    "adminMode": "enabled",
    "autoNegotiate": "enabled",
    "description": "",
    "physicalAddress": "00:01:85:48:E0:11",
    "duplexMode": "half",
    "pvId": 1,
    "type": "physical",
    "linkState": "down",
    "acceptFrame": "all",
    "speed": "unknown",
    "mtuSize": 1500,
    "interface": "0/2",
    "lastChange": 1402,
    "inOctets": 5,
    "inUcastPkts": 1,
    "inDiscards": 0,
    "inErrors": 0,
    "inUnknownProtos": 0,
    "outOctets": 5,
    "outUcastPkts": 1,
    "outDiscards": 0,
    "outErrors": 0
    "tagging": [1, 75],
    "participation" : [1, 75],

  }
  , ...

```

```

    {
      "adminMode": "enabled",
      "autoNegotiate": "enabled",
      "description": "",
      "physicalAddress": "00:01:85:48:E0:11",
      "duplexMode": "half",
      "pvId": 1,
      "type": "physical",
      "linkState": "down",
      "acceptFrame": "all",
      "speed": "unknown",
      "mtuSize": 1500,
      "interface": "0/10",
      "lastChange": 1531,
      "inOctets": 0,
      "inUcastPkts": 0,
      "inDiscards": 0,
      "inErrors": 0,
      "inUnknownProtos": 0,
      "outOctets": 0,
      "outUcastPkts": 0,
      "outDiscards": 0,
      "outErrors": 0
      "tagging": [1, 80],
      "participation" : [1, 80],

    }
  ]
,
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": "0"
  }
}

```

Multiple Interface Partial Request Example

The following is an example of a request that limits the output to the first three interfaces:

```
curl -GET -d "limit=3" 10.10.10.1:8080/open/v1/interfaces
```

browser

```
http://10.10.10.1:8080/open/v1/interfaces?limit=3
```

Response Example

```
{
  "interfaces": [
    {
      "adminMode": "enabled",
      "autoNegotiate": "enabled",
      "description": "",
      "physicalAddress": "00:01:85:48:E0:11",
      "duplexMode": "half",
      "pvId": 1,
      "type": "physical",
      "linkState": "down",
      "acceptFrame": "all",
      "speed": "unknown",
      "mtuSize": 1500,
      "interface": "0/1",
      "lastChange": 1301,
      "inOctets": 3,
      "inUcastPkts": 2,
      "inDiscards": 1,
      "inErrors": 0,
      "inUnknownProtos": 0,
      "outOctets": 3,
      "outUcastPkts": 2,
      "outDiscards": 0,
      "outErrors": 0,
      "tagging": [1, 70],
      "participation" : [1, 70],
    }
  ]
}
```

```

,
{
  "adminMode": "enabled",
  "autoNegotiate": "enabled",
  "description": "",
  "physicalAddress": "00:01:85:48:E0:11",
  "duplexMode": "half",
  "pvId": 1,
  "type": "physical",
  "linkState": "down",
  "acceptFrame": "all",
  "speed": "unknown",
  "mtuSize": 1500,
  "interface": "0/2",
  "lastChange": 1402,
  "inOctets": 5,
  "inUcastPkts": 1,
  "inDiscards": 0,
  "inErrors": 0,
  "inUnknownProtos": 0,
  "outOctets": 5,
  "outUcastPkts": 1,
  "outDiscards": 0,
  "outErrors": 0
  "tagging": [1, 75],
  "participation" : [1, 75],

}
,
{
  "adminMode": "enabled",
  "autoNegotiate": "enabled",
  "description": "",
  "physicalAddress": "00:01:85:48:E0:11",
  "duplexMode": "half",
  "pvId": 1,
  "type": "physical",
  "linkState": "down",
  "acceptFrame": "all",

```

```

        "speed": "unknown",
        "mtuSize": 1500,
        "interface": "0/3",
        "lastChange": 1102,
        "inOctets": 2,
        "inUcastPkts": 1,
        "inDiscards": 0,
        "inErrors": 0,
        "inUnknownProtos": 0,
        "outOctets": 2,
        "outUcastPkts": 1,
        "outDiscards": 0,
        "outErrors": 0
        "tagging": [1, 80],
        "participation" : [1, 80],
    }
]
,
"status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": "0"
}
}

```

Single Interface Request Example

Single interface retrievals are specified by using the **name** parameter. All the available interface resources are retrieved by default.

The following is an example of a specific interface request located in slot 0, port 34:

```
curl -GET -d "interface=0/34"
10.10.10.1:8080/open/v1/interfaces
```

browser:

```
http://10.10.10.1:8080/open/v1/interfaces?interface=0/34
```

Response Example

```
{
  "interfaces": [
    {
      "adminMode": "enabled",
      "autoNegotiate": "enabled",
      "description": "Engineering Dept.",
      "physicalAddress": "00:01:85:48:E0:11",
      "duplexMode": "full",
      "pvId": 1,
      "type": "physical",
      "linkState": "up",
      "acceptFrame": "all",
      "speed": "1000-full-duplex",
      "mtuSize": 1500,
      "interface": "0/34",
      "lastChange": 1102,
      "inOctets": 2,
      "inUcastPkts": 1,
      "inDiscards": 0,
      "inErrors": 0,
      "inUnknownProtos": 0,
      "outOctets": 2,
      "outUcastPkts": 1,
      "outDiscards": 0,
      "outErrors": 0
      "tagging": [1, 80],
      "participation" : [1, 80],
    }
  ]
  ,
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": "0"
  }
}
```

```
}
```

Single Interface Partial Request Example

The response can be reduced by using the partial response syntax as denoted by the **fields** parameter. Following is an example of a specific interface, while limiting the results to VLAN related attributes.

```
curl -GET -d "interface=0/34fields=
participation,acceptFrame,tagging"
10.10.10.1:8080/open/v1/interfaces
```

browser

```
http://10.10.10.1:8080/open/v1/interfaces?interface=
0/34&fields=participation,acceptFrame
```

Response Example

```
"interfaces": [
    {
        "acceptFrame": "all",
        "participation": "included",
        "interface": "0/34"
        "tagging": "untagged"
    }
]
,
"status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": "0"
}
}
```

Ancillary Request Example

The following is an example using an ancillary request parameter to determine the maximum number of characters allowed for the given interface description.

- `curl -GET -d "interface=0/1&fields=descriptionMax" 10.27.21.31/open/v1/interfaces`
- `curl -GET -d "fields=descriptionMax" 10.10.10.1:8080/open/v1/interfaces`
- browser
`http://10.10.10.1:8080/open/v1/interfaces?fields=descriptionMax`

Response Example

```
{
  "interfaces": [
    {
      "descriptionMax": 64,
      "interface": "0/1"
    }
  ],
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": "0"
  }
}
```

Interface UPDATE

```
PUT /open/v1/interfaces
```

Use this call to update an existing interface on the switch. The interface name is mandatory.

Request Details

Table 4-46. Interface UPDATE Request Details

Parameter	Type	Description
acceptFrame	all, tagged, untagged	Indicates how the interface handles untagged and priority tagged frames.

Table 4-46. Interface UPDATE Request Details (Continued)

Parameter	Type	Description
adminMode	enabled, disabled	The administrative state of the interface, which is either enabled or disabled.
autoNegotiate	enabled, disabled	Indicates the auto-negotiation mode for physical interfaces.
description	string	Description of the interface for identification purposes.
interface	slot/port	Interface name. Identifies the port or LAG. The interface name is mandatory for all updates.
mtuSize	integer	The Maximum Transmit Unit (MTU) size of the interface.
participation	auto, included, excluded	VLAN participation mode. Interface name and VLAN id is mandatory.
pvId	port VLAN id	Assign this port to the associated VLAN. Interface name and VLAN ID is mandatory.
speed	See Available Speeds	The physical link data rate of the interface.
tagging	enabled, disabled	The interface tagging mode for the given VLAN. Interface name and VLAN id is mandatory.
vlan	integer	VLAN ID. Mandatory for participation, tagging, and pvId.

Available Speeds

- auto
- 10-full-duplex
- 10-half-duplex
- 100-full-duplex
- 100-half-duplex
- 100-fx-full-duplex

- 1000-full-duplex
- 10g-full-duplex
- 20g-full-duplex
- 25g-full-duplex
- 40g-full-duplex
- 50g-full-duplex
- 100g-full-duplex
- 155-aal5
- 2.5g-fd

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, 404, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-47. Interface UPDATE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Operation Fail.
-28	OPEN_E_UNAVAIL	Parameter or feature is not supported.
-29	OPEN_E_NOT_FOUND	Parameter not found.

Update Interface Request Example

The following is an example to update the administration mode and description for interface located in slot 0, port 1:

```
curl -GET -XPUT -d "interface=0/34&adminMode=enabled&speed=100-full-duplex" 10.10.10.1:8080/open/v1/interfaces
```

browser:

```
http://10.10.10.1:8080/open/v1/interfaces?method=
put&interface=0/34&adminMode=enabled&speed=1000-full-duplex
```

Response Example

```
{
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": "0"
  }
}
```

Interface DELETE

NOTE: Interfaces cannot be deleted. However, you can reconfigure or UPDATE their parameters using PUT commands.

VLANs

VLAN CREATE

```
POST /open/v1/vlans
```

Use this call to create a VLAN with the ID specified on the switch. The VLAN ID is mandatory. VLAN name is optional, but recommended.

Request Details

Table 4-48. VLAN CREATE Request Details

Parameter	Type	Description
id	integer	VLAN ID. Refer to the ancillary vlanMax GET request for VLAN limits.
name	string	VLAN name. Up to 32 alpha-numeric characters.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, 404, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-49. VLAN CREATE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-24	OPEN_E_EXISTS	Already exists.
-26	OPEN_E_FAIL	Operation fail.

Request Example

The following is a VLAN CREATE request example:

```
curl -POST -d "id=2&name=test" 10.10.10.1:8080/open/v1/vlans
```

browser:

```
http://10.10.10.1:8080/open/v1/vlans?method=post&id=2&name=test
```

Response Example

Returns an appropriate status message.

```
{
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

VLAN READ

GET /open/v1/vlans

Retrieves one or all available VLANs.

Request Details

Table 4-50. VLAN READ Request Details

Parameter	Type	Description
id	integer	VLAN ID. Always returned.
fields		Partial response syntax.
fields	id	VLAN ID.
	name	VLAN name.
	type	VLAN type is 'static' or 'default'. The default VLAN type is identified as default.
	interface	Interface retrieval can be bypassed.
	vlanMax	Ancillary request parameter to determine the maximum number of allowed VLANs for the given platform.

Table 4-50. VLAN READ Request Details (Continued)

Parameter	Type	Description
offset	integer	Denotes beginning index for pagination. Default is 1. This parameter is not valid for single VLAN retrievals.
limit	integer	Denotes total retrieval count. Default is 10. This parameter is not valid for single VLAN retrievals.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, 404, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-51. VLAN READ Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Operation Fail.
-29	OPEN_E_NOT_FOUND	Parameter not found.

Multiple VLAN Request Example

The following example shows a multiple VLAN retrieval. Note that pagination is implicit.

```
curl -GET 10.10.10.1:8080/open/v1/vlans
browser:
http://10.10.10.1:8080/open/v1/vlans
```

Response Example

```
{
```

```

"vlans": [
{
  "id": 1,
  "type": "default",
  "name": "default"
},
{
  "id": 2,
  "type": "static",
  "name": "vlan2"
},
{
  "id": 10,
  "type": "static",
  "name": "vlan10"
},
{
  "id": 14,
  "type": "static",
  "name": "vlan14"
},

```

Multiple VLAN Partial Request Example

An example of a request that limits the VLAN output to 3.

```
curl -GET -d "limit=3" 10.10.10.1:8080/open/v1/vlans
```

browser:

```
http://10.10.10.1:8080/open/v1/vlans?limit=3
```

Response Example

```

{
  "vlans": [
  {
    "id": 1,
    "type": "default",
    "name": "default"
  },

```

```
{
  "id": 2,
  "type": "static",
  "name": "vlan2"
},
{
  "id": 10,
  "type": "static",
  "name": "vlan10"
},
],
"status": {
  "response_code": 200,
  "msg": "OK",
  "more_info": "",
  "error_code": 0
}
}
```

Single VLAN Request Example

Single VLAN retrievals are specified by using the `id` parameter.

The interfaces are included for single VLAN retrievals. As such, all the available interfaces are scanned and if the associated interface is assigned to the given VLAN, then the interface is included in the response.

All the available VLAN resources are retrieved by default.

```
curl -GET -d "id=2" 10.10.10.1:8080/open/v1/vlans
```

browser:

```
http://10.10.10.1:8080/open/v1/vlans?id=2
```

Response Example

```
{
  "vlans": [
    {
      "id": 2,
      "type": "static",
```

```

    "name": "vlan2",
    "interfaces": [
      {
        "pvid": 2,
        "participation": "include",
        "tagging": "untagged",
        "interface": "0/4"
      }
    ]
  },
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}

```

Single VLAN Partial Request Example

The following is an example of a specific VLAN request excluding the interfaces.

```

curl -GET -d "id=2&fields=name,type"
10.10.10.1:8080/open/v1/vlans

```

browser:

```

http://10.10.10.1:8080/open/v1/vlans?id=2&fields=name,type

```

Response Example

```

{
  "vlans": [
    {
      "id": 2,
      "type": "static",
      "name": "vlan2_updated"
    }
  ],
}

```

```

    "status": {
      "response_code": 200,
      "msg": "OK",
      "more_info": "",
      "error_code": 0
    }
  }
}

```

Ancillary Request Example

The following is an example using an ancillary request parameter to determine the maximum number of allowed VLANs for the given platform.

```

curl -GET -d "fields=vlanMax" 10.10.10.1:8080/open/v1/vlans

```

browser:

```

http://10.10.10.1:8080/open/v1/vlans?fields=vlanMax

```

Response Example

```

{
  "vlanMax": 4093,
  "vlans": [
    {
      "id": 1
    }
  ],
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}

{
  "id": 15,
  "type": "static",
  "name": "vlan15"
}

```

```

],
"status": {
  "response_code": 200,
  "msg": "OK",
  "more_info": "",
  "error_code": 0
}
}

```

Multiple VLAN Partial Request Example

An example of a request that limits the VLAN output to 3.

```
curl -GET -d "limit=3" 10.10.10.1:8080/open/v1/vlans
```

browser:

```
http://10.10.10.1:8080/open/v1/vlans?limit=3
```

Response Example

```

{
  "vlans": [
    {
      "id": 1,
      "type": "default",
      "name": "default"
    },
    {
      "id": 2,
      "type": "static",
      "name": "vlan2"
    },
    {
      "id": 10,
      "type": "static",
      "name": "vlan10"
    }
  ],
  "status": {
    "response_code": 200,

```

```

    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}

```

Single VLAN Request Example

Single VLAN retrievals are specified by using the `id` parameter.

The interfaces are included for single VLAN retrievals. As such, all the available interfaces are scanned and if the associated interface is assigned to the given VLAN, then the interface is included in the response.

All the available VLAN resources are retrieved by default.

```
curl -GET -d "id=2" 10.10.10.1:8080/open/v1/vlans
```

browser:

```
http://10.10.10.1:8080/open/v1/vlans?id=2
```

Response Example

```

{
  "vlans": [
    {
      "id": 2,
      "type": "static",
      "name": "vlan2",
      "interfaces": [
        {
          "pvid": 2,
          "participation": "include",
          "tagging": "untagged",
          "interface": "0/4"
        }
      ]
    }
  ],
  "status": {
    "response_code": 200,

```

```

    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}

```

Single VLAN Partial Request Example

The following is an example of a specific VLAN request excluding the interfaces.

```

curl -GET -d "id=2&fields=name,type"
10.10.10.1:8080/open/v1/vlans

```

browser:

```

http://10.10.10.1:8080/open/v1/vlans?id=2&fields=name,type

```

Response Example

```

{
  "vlans": [
    {
      "id": 2,
      "type": "static",
      "name": "vlan2_updated"
    }
  ],
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}

```

Ancillary Request Example

The following is an example using an ancillary request parameter to determine the maximum number of allowed VLANs for the given platform.

```

curl -GET -d "fields=vlanMax" 10.10.10.1:8080/open/v1/vlans

```

browser:

```
http://10.10.10.1:8080/open/v1/vlans?fields=vlanMax
```

Response Example

```
{
  "vlanMax": 4093,
  "vlans": [
    {
      "id": 1
    }
  ],
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

VLAN UPDATE

```
PUT /open/v1/vlans
```

Use this call to update an existing VLAN on the switch. The VLAN ID is mandatory.

Request Details

Table 4-52. VLAN UPDATE Request Details

Parameter	Type	Description
id	integer	VLAN ID.
name	string	VLAN name. Up to 32 alpha-numeric characters.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, 404, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-53. VLAN UPDATE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Operation fail.

VLAN Request Example

The following is a VLAN UPDATE request example.

```
curl -GET -XPUT -d "id=2&name=vlan2_update1"
10.10.10.1:8080/open/v1/vlans
```

browser:

```
http://10.10.10.1:8080/open/v1/vlans?method=put&id=2&name=
vlan2_update1
```

Response Example

```
{
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

VLAN DELETE

```
DELETE /open/v1/vlans
```

Use this call to delete an existing VLAN on the switch. The VLAN ID is mandatory.

Request Example

The following is a VLAN DETETE request example:

```
curl -GET -XDELETE -d "id=3" 10.10.10.1:8080/open/v1/vlans
browser:
http://10.10.10.1:8080/open/v1/vlans?method=delete&id=3
```

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, 404, and 500.

Request Details

Table 4-54. VLAN DELETE Request Details

Parameter	Type	Description
id	integer	VLAN ID.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-55. VLAN DELETE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Operation fail.

Response Example

```
{
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",

```

```
    "error_code": 0
  }
}
```

Overlay Networking

IP

ip CREATE

This component does not support create semantics.

ip READ

GET /open/v1/ip

Retrieves one or more available ip.

Request Details

Table 4-56. ip READ Request Details

Parameter	Type	Description
ipAddr	string	IP Address entry. Always returned.
fields		Partial response syntax.
	ipAddr	IP Address entry.
	ifIndex	The index value which uniquely identifies the interface to which this entry is applicable.
	netMask	The subnet mask associated with the IP address of this entry.
	bCast	The value of the least-significant bit in the IP broadcast address used for sending datagrams on the (logical) interface associated with the IP address of this entry.
	reasmMaxSize	Size of the largest IP datagram which this entity can re-assemble from incoming IP fragmented datagrams received on this interface.
offset	integer	Denotes beginning index for pagination. Default is 1. This parameter is not valid for single IP retrievals.

Table 4-56. ip READ Request Details (Continued)

Parameter	Type	Description
limit	integer	Denotes total retrieval count. Default is 10. This parameter is not valid for single IP retrievals.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-57. ip READ Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Operation fail.

Multiple ip Request Example

The following example shows a multiple ip retrieval. Note that pagination is implicit.

```
curl -GET 10.10.10.1:8080/open/v1/ip
```

browser:

```
http://10.10.10.1:8080/open/v1/ip
```

Response Example

```
{
  "ip": [
    {
      "ipAddr": "10.11.12.11",
      "ifIndex": "0/1",
```

```

        "netMask": "255.255.255.128",
        "bCast": 1,
        "reasmMaxSize": 120
      },
      {
        "ipAddr": "10.11.12.12",
        "ifIndex": "0/2",
        "netMask": "255.255.255.128",
        "bCast": 1,
        "reasmMaxSize": 120
      },
      ...
      {
        "ipAddr": "10.11.12.19",
        "ifIndex": "0/9",
        "netMask": "255.255.255.128",
        "bCast": 1,
        "reasmMaxSize": 120
      }
    ],
    "status": {
      "response_code": 200,
      "msg": "OK",
      "more_info": "",
      "error_code": "0_"
    }
  }
}

```

Multiple ip Partial Request Example

```

curl -GET -d "limit=2" 10.10.10.1:8080/open/v1/ip

```

browser:

```

http://10.10.10.1:8080/open/v1/ip?limit=2

```

Response Example

```

{
  "ip": [
    {

```

```

        "ipAddr": "10.11.12.11",
        "ifIndex": "0/1",
        "netMask": "255.255.255.128",
        "bCast": 1,
        "reasmMaxSize": 120
    },
    {
        "ipAddr": "10.11.12.12",
        "ifIndex": "0/2",
        "netMask": "255.255.255.128",
        "bCast": 1,
        "reasmMaxSize": 120
    },
],
"status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": "0"
}

```

Single ip Request Example

curl -GET -d "ipAddr=10.11.12.11" 10.10.10.1:8080/open/v1/ip
 browser:
<http://10.10.10.1:8080/open/v1/ip?ipAddr=10.11.12.11>

Response Example

```

{
    "ip": [
        {
            "ipAddr": "10.11.12.11",
            "ifIndex": "0/1",
            "netMask": "255.255.255.128",
            "bCast": 1,
            "reasmMaxSize": 120
        }
    ],
    "status": {

```

```

        "response_code": 200,
        "msg": "OK",
        "more_info": "",
        "error_code": "0"
    }
}

```

Single ip Partial Request Example

The response can be reduced by using the partial response syntax as denoted by the *fields* parameter.

```

"curl -GET -d "ipAddr=10.11.12.11&fields=ifIndex"
10.10.10.1:8080/open/v1/ip

```

browser

```

http://10.10.10.1:8080/open/v1/ip?ipAddr=10.11.12.11&fields=
ifIndex

```

Response Example

```

{
    "ip": [
        {
            "ipAddr": "10.11.12.11",
            "ifIndex": "0/1",
        }
    ],
    "status": {
        "response_code": 200,
        "msg": "OK",
        "more_info": "",
        "error_code": "0"
    }
}

```

ip UPDATE

This component does not support update semantics.

ip DELETE

This component does not support delete semantics.

lagports

lagports implements a RESTful API for the OpEN LAG API.

LAG PORT CREATE

`POST /open/v1/lagports`

Use this call to set port as a member of a specified LAG. All fields are mandatory.

Request Details

Table 4-58. LAG PORT CREATE Request Details

Parameter	Type	Description
lag	string	LAG identifier in slot/port format. Refer to the ancillary membersMax GET request for LAG limits.
port	string	Interface identifier in slot/port format.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, 404, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-59. LAG PORT CREATE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_Internal	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-24	OPEN_E_EXISTS	Already exists.
-26	OPEN_E_FAIL	Operation fail.

Request Example

```
"curl -POST -d "lag=1/1&port=0/1"  
10.10.10.1:8080/open/v1/lagports
```

browser

```
http://10.10.10.1:8080/open/v1/lagports?method=post&lag=  
1/1&port=0/1
```

Response Example

Returns an appropriate status message.

```
{  
  "status": {  
    "response_code": 200,  
    "msg": "OK",  
    "more_info": "",  
    "error_code": 0  
  }  
}
```

LAG PORT READ

lagports does not support read semantics. 'lags' resource should be used to retrieve LAGs configured.

LAG PORT UPDATE

lagports does not support update semantics.

LAG PORT DELETE

```
DELETE /open/v1/lagports
```

Use this call to remove port from LAG. All fields are mandatory.

Request Example

```
curl -GET -XDELETE -d "lag=1/1&port=0/1"  
10.10.10.1:8080/open/v1/lagports
```

browser:

```
http://10.10.10.1:8080/open/v1/lagports?method=delete&lag=1/1&port=0/1
```

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, 404, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-60. LAG PORT DELETE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Operation Fail.

Request Details

Table 4-61. LAG PORT DELETE Request Details

Parameter	Type	Description
lag	string	LAG identifier in slot/port format.
port	string	Interface identifier in slot/port format.

Response Example

```
{
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-62. LAG PORT DELETE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Operation Fail.

Request Details

Table 4-63. LAG PORT DELETE Request Details

Parameter	Type	Description
lag	string	LAG identifier in slot/port format.
port	string	Interface identifier in slot/port format.

Response Example

```
{
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

lags

lags implements a RESTful API for the OpEN LAG API.

LAG CREATE

Note that lags cannot be created. You can however reconfigure or UPDATE their parameters using PUT commands.

LAG READ

```
GET /open/v1/lags
```

Retrieves one or more available LAGs.

Request Details

Table 4-64. LAG READ Request Details

Parameter	Type	Description
lag	slot/port	LAG identifier in slot/port format. Identifies the LAG. LAG name is mandatory.
fields	—	Partial response syntax.
—	loadBalanceMode	Indicates the LAG load balance mode.
—	name	The current name, if any, associated with the lag to help identify it.
—	staticMode	Indicates the static mode for LAG which is either Enable or Disable.
—	members	List of LAG members.
—	activeMembers	List of active LAG members.
—	membersMax	Ancillary request parameter to determine the maximum number of LAG members possible for the given platform.
—	minUpLinks	Ancillary request parameter to determine the minimum number of Up links.
—	nameMax	Ancillary request parameter to get the maximum number of characters allowed for this LAG name.
—	thresholdMinMax	Ancillary request parameter to get the minimum and maximum threshold of LAG members for a platform.

Table 4-64. LAG READ Request Details (Continued)

Parameter	Type	Description
offset	integer	Denotes beginning index for pagination. Default is 1. This parameter is not valid for single interface retrievals.
limit	integer	Denotes total retrieval count. Default is 10. This parameter is not valid for single interface retrievals.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, 404, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-65. LAG READ Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Operation Fail.

The following example shows a multiple LAGs retrieval. Note that pagination is implicit.

```
curl -GET 10.10.10.1:8080/open/v1/lags
browser
http://10.10.10.1:8080/open/v1/lags
```

Response Example

```
{
  "lags": [
    {
```

```

"nameMax": 16,
"staticMode": "enabled",
    "lag": "3/1",
    "loadBalanceMode": "sda_vlan",
    "name": "lag 1",
"membersMax": 32,
"minUpLinks": 1,
"thresholdMax": 32,
"thresholdMin": 1,
    "members": [
        "0/74", "0/75"
    ],
    "activeMembers": [
        "0/74"
    ]
},
... /** more records here */
{
"nameMax": 16,
"staticMode": "enabled",
    "lag": "3/9",
    "loadBalanceMode": "sda_vlan",
    "name": "lag 1",
"membersMax": 32,
"minUpLinks": 1,
"thresholdMax": 32,
"thresholdMin": 1,
    "members": [
        "0/13"
    ],
    "activeMembers": [
        "0/13"
    ]
},
{
"nameMax": 16,
"staticMode": "enabled",
    "lag": "3/10",
    "loadBalanceMode": "sda_vlan",
    "name": "lag 1",
"membersMax": 32,
"minUpLinks": 1,
"thresholdMax": 32,
"thresholdMin": 1,
    "members": [
        "0/3", "0/5"
    ]
}

```

```

        ],
        "activeMembers": [ ]
    }
],
_____ "status":
    {
        _____ "response_code": 200,
        "msg": "OK",
        "more_info": "",
        "error_code": 0
    }
}

```

Multiple LAG Partial Request Example

The following example request limits the output to the first two LAGs.

```

curl -GET -d "limit=2" 10.10.10.1:8080/open/v1/lags
browser
http://10.10.10.1:8080/open/v1/lags?limit=2

```

Response Example

```

{
    "lags": [
        {
            "nameMax": 16,
            "staticMode": "enabled",
            "lag": "3/1",
            "loadBalanceMode": "sda_vlan",
            "name": "lag 1",
            "membersMax": 32,
            "minUpLinks": 1,
            "thresholdMax": 32,
            "thresholdMin": 1,
            "members": [
                "0/74", "0/75"
            ],
            "activeMembers": [
                "0/74"
            ]
        }
    ]
}

```

```

    },
    {
      "nameMax": 16,
      "staticMode": "enabled",
      "lag": "3/2",
      "loadBalanceMode": "sda_vlan",
      "name": "lag 1",
      "membersMax": 32,
      "minUpLinks": 1,
      "thresholdMax": 32,
      "thresholdMin": 1,
      "members": [
        "0/3"
      ],
      "activeMembers": [
        "0/3"
      ]
    },
  ],
  "status":
  {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}

```

Single LAG Request Example

Single LAG retrievals are specified by using the name parameter. All the available LAG resources are retrieved by default.

The following example is for a specific LAG request located in slot 1, port 1.

```

curl -GET -d "lag=3/1" 10.10.10.1:8080/open/v1/lags
browser
http://10.10.10.1:8080/open/v1/lags?lag=3/1

```

Response Example

```
{
  "lags": [
    {
      "nameMax": 16,
      "staticMode": "enabled",
      "lag": "3/1",
      "loadBalanceMode": "sda_vlan",
      "name": "lag 1",
      "membersMax": 32,
      "minUpLinks": 1,
      "thresholdMax": 32,
      "thresholdMin": 1,
      "members": [
        "0/74", "0/75"
      ],
      "activeMembers": [
        "0/74"
      ]
    },
  ],
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": "0"
  }
}
```

Single LAG Partial Request Example

The response can be reduced by using the partial response syntax as denoted by the *fields* parameter.

The following is an example of a specific LAG, while limiting the results to LACP enabled.

```
curl -GET -d "lag=3/1&fields=name,loadBalanceMode"
10.10.10.1:8080/open/v1/lags
```

browser

```
http://10.10.10.1:8080/open/v1/lags?lag=3/1&fields=
name,loadBalanceMode
```

Response Example

```
{
  "lags": [
    {
      "lag": "3/1",
      "loadBalanceMode": "sda_vlan",
      "name": "lag 100",
    }
  ],
  "status":
  {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Ancillary Request Example

The following example uses an ancillary request parameter to determine the maximum number of allowed LAG members for the given platform.

```
curl -GET -d "fields=membersMax" 10.10.10.1:8080/open/v1/lags
browser
http://10.10.10.1:8080/open/v1/lags?fields=membersMax
```

Response Example

```
{
  "lags": [
    {
      "lag": "3/1",
      "membersMax": 16
    }
  ],
  "status":
  {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

```
}  
}
```

Error Example

```
curl -GET -d "lag=2/1" 10.10.10.1:8080/open/v1/lags  
browser  
http://10.10.10.1:8080/open/v1/lags?method=get&lag=2/1
```

Response Example

```
{  
  "status": {  
    "response_code": 404,  
    "msg": "Not Found",  
    "more_info": "",  
    "error_code": -29  
  }  
}
```

LAG UPDATE

```
PUT /open/v1/lags
```

Use this call to update an existing LAG on the switch. The LAG name is mandatory.

Request Details

Table 4-66. LAG UPDATE Request Details

Parameter	Type	Description
lag	string	LAG name in slot/port format. Identifies the LAG. The LAG name is mandatory for all updates.
loadBalanceMode	string (see Load Balance)	Indicates the LAG load balance mode.
minUpLinks	string	Minimum number of Up links for a LAG.
name	string	Name associated with the lag to help identify it.

Table 4-66. LAG UPDATE Request Details (Continued)

Parameter	Type	Description
staticMode	string	Static mode for LAG (enabled/disabled).

Load Balance

- sa_vlan
- da_vlan
- sda_vlan
- sip_sport
- dip_dport
- sdip_dport
- enhanced

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, 404, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-67. LAG UPDATE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Operation Fail.
-28	OPEN_E_UNAVAIL	Parameter or feature is not supported.
-29	OPEN_E_NOT_FOUND	Parameter not found.

Update LAG Request Example

The following example enables the static mode and sets name to *lag 100* for LAG located in slot 1, port 1.

```
curl -GET -XPUT -d "lag=1/1&staticMode=enabled&name=lag%20100"  
10.10.10.1:8080/open/v1/lags
```

browser

```
http://10.10.10.1:8080/open/v1/lags?method=put&lag=  
1/1&staticMode=enabled&name=lag%20100
```

Response Example

```
{  
  "status": {  
    "response_code": 200,  
    "msg": "OK",  
    "more_info": "",  
    "error_code": 0  
  }  
}
```

LAG DELETE

Note that LAGs cannot be deleted. You can however reconfigure or UPDATE their parameters using PUT commands.

mfdb

MFDB CREATE

This component does not support create semantics.

MFDB READ

```
GET /open/v1/mfdb
```

Retrieves all of the variables that are associated with the MFDB object.

Request Details

Table 4-68. MFDB READ Request Details

Parameter	Type	Description
mac	string	MAC address of the entry (Requires <i>vlanId</i>).
vlanId	integer	VLAN ID of the entry (Requires <i>mac</i>).
fields	—	Partial response syntax.
—	mac	MAC address of the entry.
—	vlanId	VLAN ID of the entry.
—	source	The protocol type.
—	type	The entry type.
—	description	The entry description.
—	frwdIntMask	The forwarding interfaces. Comma-separated list of interfaces. Example: <i>0/1,0/23</i> .
—	fltIntMask	The filtering interfaces. Comma-separated list of interfaces. Example: <i>0/11,0/14</i> .
—	absPortMask	The interfaces of the entry. Comma-separated list of interfaces. Example: <i>0/11,0/12</i> .
—	mfdbCurrEntries	The current number of entries in the MFDB table.
—	mfdbMaxTableEntries	Maximum number of entries that the SSM MFDB table can hold for a given family.
—	mfdbMostEntries	Most number of entries ever in the SSM MFDB table for a given family.

Table 4-68. MFDB READ Request Details (Continued)

Parameter	Type	Description
offset	integer	Denotes beginning index for pagination. Default is 1.
limit	integer	Denotes total retrieval count. Default is 10.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-69. MFDB READ Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-22	OPEN_E_PARAM	Invalid request.
-26	OPEN_E_FAIL	Operation failed.
-28	OPEN_E_UNAVAIL	Feature is not supported.

Multiple MFDB Entries Request Example

```
curl -GET 10.10.10.1:8080/open/v1/mfdbs
browser
http://10.10.10.1:8080/open/v1/mfdbs
```

Response Example

```
{
  "mfdbCurrEntries": 3,
  "mfdbMaxTableEntries": 100,
  "mfdbMostEntries": 30,
  "mfdbEntries": [
    {
      "vlanId": 1,
      "mac": "11:11:11:11:11:11",

```

```

        "source": "Filter",
        "type": "Static",
        "description": "Mgmt Config",
        "absPortMask": "0/1",
        "frwdIntMask": "0/2,0/3",
        "fltIntMask": ""
    },
    {
        "vlanId": 3,
        "mac": "33:33:33:33:33:33",
        "source": "Filter",
        "type": "Static",
        "description": "Mgmt Config",
        "absPortMask": "",
        "frwdIntMask": "",
        "fltIntMask": ""
    },
    {
        "vlanId": 5,
        "mac": "55:55:55:55:55:55",
        "source": "Filter",
        "type": "Static",
        "description": "Mgmt Config",
        "absPortMask": "",
        "frwdIntMask": "",
        "fltIntMask": ""
    },
],
"status":
{
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
}
}

```

Multiple MFDB Entry Partial Request Example

```

curl -GET -d "fields=source&limit=2"
10.10.10.1:8080/open/v1/mfdbs

```

browser

```

http://10.10.10.1:8080/open/v1/mfdbs?fields=source&limit=2

```

Response Example

```
{
  "mfdbEntries": [
    {
      "source": "Filter"
    },
    {
      "source": "IGMP"
    }
  ],
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Single MFDB Entry Partial Request Example

```
curl -GET -d "mac=33:33:33:33:33:33&vlanId=3"
10.10.10.1:8080/open/v1/mfdbs
```

browser

```
http://10.10.10.1:8080/open/v1/mfdbs?mac=
33:33:33:33:33:33&vlanId=3
```

Response Example

```
{
  "mfdbEntries": [
    {
      "vlanId": 3,
      "mac": "33:33:33:33:33:33",
      "source": "Filter",
      "type": "Static",
      "description": "Mgmt Config",
      "absPortMask": "",
      "frwdIntMask": "",
      "fltIntMask": ""
    }
  ],
  "status":
```

```

    {
        "response_code": 200,
        "msg": "OK",
        "more_info": "",
        "error_code": 0
    }
}

```

Error Example

```

curl -GET -d "x=-99" 10.10.10.1:8080/open/v1/mfdbs
browser
http://10.10.10.1:8080/open/v1/mfdbs?method=get&x=-99

```

Response Example

Returns an appropriate status message.

```

{
    "status":
    {
        "response_code": 400,
        "msg": "Bad Request",
        "more_info": "Invalid arguments",
        "error_code": -22
    }
}

```

MFDB UPDATE

This component does not support update semantics.

MFDB DELETE

This component does not support delete semantics.

System

Registry

The registry resource represents a RESTful API that exposes various data about the switch via its system registry.

Registry CREATE

This component does not support create semantics.

Registry READ

```
GET /open/v1/registry
```

Retrieves all of the data available from the system registry.

Request Details

Table 4-70. Registry READ Request Details

Parameter	Type	Description
serialNumber	string	Serial number of the unit.
machineType	string	Model description.
machineModel	string	Model number.

Response Details

HTTP Response Codes

The possible HTTP response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-71. Registry READ Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-22	OPEN_E_PARAM	Invalid request.

Table 4-71. Registry READ Internal Error Codes (Continued)

Error Code	Code Name	Description
-26	OPEN_E_FAIL	Operation failed.

Request Example

```
curl -GET 10.10.10.1:8080/open/v1/registry
```

browser:

```
http://10.10.10.1:8080/open/v1/registry
```

Response Example

```
{
  "machineType": "type",
  "machineModel": "model",
  "serialNumber": "serial number",
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Error Example

```
curl -GET -d "x=-99" 10.10.10.1:8080/open/v1/registry
```

browser:

```
http://10.10.10.1:8080/open/v1/registry?method=get&x=-99
```

Response Example

Returns an appropriate status message.

```
{
  "status": {
    "response_code": 400,
    "msg": "Bad Request",
  }
}
```

```
        "more_info": "Invalid arguments",  
        "error_code": -22  
    }  
}
```

Registry UPDATE

The registry resource does not support update semantics.

Registry DELETE

This registry resource does not support delete semantics.

serviceport

serviceport implements a RESTful API for the OpEN API. This version of the API only supports IPV4.

Service port CREATE

Create semantics are not supported by this resource

Service port READ

GET /open/v1/serviceport

Retrieves various data associated with the device's serviceport.

Request Details

Table 4-72. Service port READ Request Details

Parameter	Type	Description
ipv4addr	string	IPV4 dot notation address (required if ipv4netmask present).
ipv4netmask	string	IPV4 dot notation netmask (required if ipv4addr present).
ipv4gateway	string	IPV4 dot notation gateway address (both ipv4addr and ipv4netmask must be specified when present).
ipv4protocolMode	integer	integer(optional) • 1 = Protocol mode none • 2 = Protocol mode BOOTP • 3 = Protocol mode DHCP

NOTE: If none of the preceding parameters are specified, an error will be reported.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-73. Service port READ Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Request failed.

Request Example

```
curl -GET 10.10.10.1:8080/open/v1/serviceport
browser
http://10.10.10.1:8080/open/v1/serviceport
```

Response Example

```
{
  "IPv4" :
  {
    "addr" : "10.14.240.47",
    "netmask": "255.255.255.0",
    "gateway" : "255.255.255.0",
    "protocolMode" : 3
  }
  "status":
  {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Error Example

```
curl -GET -d "tenant=-99" 10.10.10.1:8080/open/v1/serviceport
browser
http://10.10.10.1:8080/open/v1/serviceport?method=get&tenant=-99
```

Response Example

Returns an appropriate status message.

```
{
  "status":
  {
    "response_code": 400,
    "msg": "Bad Request",
    "more_info": "Invalid parameters"
    "error_code": -22
  }
}
```

Service port UPDATE

```
PUT /open/v1/serviceport
```

Use this call to update the service port parameters.

Caution! This request is issued over a connection to the serviceport, the results are undefined. Likely, the HTTP request will not complete. It is advised that other means for configuring the serviceport be used in this scenario.

Request Details

Table 4-74. Service port UPDATE Request Details

Parameter	Type	Description
ipv4addr	string	IPv4 dot notation address (required if ipv4netmask present).
ipv4netmask	string	IPv4 dot notation netmask (required if ipv4addr present).

Table 4-74. Service port UPDATE Request Details (Continued)

Parameter	Type	Description
ipv4gateway	string	IPv4 dot notation gateway address (both ipv4addr and ipv4netmask must be specified when present).
ipv4protocolMode	integer	integer(optional) • 1 = Protocol mode none • 2 = Protocol mode BOOTP • 3 = Protocol mode DHCP

NOTE: If none of the preceding parameters are specified, an error will be reported.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-75. Service port UPDATE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Request was successful.
-22	OPEN_E_PARAM	Error in parameters passed, or none of the optional parameters was specified.
-26	OPEN_E_FAIL	Update failed.
-31	OPEN_E_UNAVAIL	Feature is not supported.

Request Example

```
curl -GET -XPUT -d "ipv4addr=10.14.240.45&ipv4netmask=
255.255.255.0&ipv4gateway=10.14.240.1"
10.10.10.1:8080/open/v1/serviceport

browser

http://10.10.10.1:8080/open/v1/serviceport?method=
put&ipv4addr=10.14.240.45&ipv4netmask=
255.255.255.0&ipv4gateway=10.14.240.1
```

Response Example

```
{
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Error Example

```
curl -GET -XPUT -d "vlanId=2"
10.10.10.1:8080/open/v1/serviceport
```

browser

```
http://10.10.10.1:8080/open/v1/serviceport?method=put&vlanId=2
```

Response Example

```
{
  "status": {
    "response_code": 400,
    "msg": "Bad Request",
    "more_info": "Invalid arguments"
    "error_code": -22
  }
}
```

Service port DELETE

This resource does not support delete semantics.

Snooping Configuration

Snooping configuration CREATE

This component does not support create semantics.

Snooping configuration READ

```
GET /open/v1/snoopingconfig
```

Retrieves all of the variables that are associated with the snooping configuration.

Request Details

Table 4-76. Snooping configuration READ Request Details

Parameter	Type	Description
family	igmp/mld	Snooping family. The value is mandatory.
fields	—	Partial response syntax.
fields	adminMode	IGMP/MLD Admin mode (enabled/disabled).
—	proxyQuerierAdminMode	IGMP/MLD Proxy Querier Admin mode (enabled/disabled).
—	controlFrames	Number of multicast control frames processed by the CPU.
—	forwardedFrames	Number of multicast data frames forwarded by the CPU.
—	floodAllUnknownPort	Configured mode for flooding unknown multicast traffic to all ports (true/false).

Table 4-76. Snooping configuration READ Request Details (Continued)

Parameter	Type	Description
—	*isFamilyTypeSupported*	Ancillary request parameter to determine if the passed family type snooping is supported or not (true/false).

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-77. Snooping configuration READ Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-22	OPEN_E_PARAM	Invalid request.
-26	OPEN_E_FAIL	Operation failed.

Request Example

```
curl -GET -d "family=igmp"
10.10.10.1:8080/open/v1/snoopingconfig
browser
http://10.10.10.1:8080/open/v1/snoopingconfig?family=igmp
```

Response Example

```
{
  "adminMode": "enabled",
  "proxyQuerierAdminMode": "enabled",
  "family": "igmp",
  "controlFrames": 1,
  "forwardedFrames": 5,
```

```

    "floodAllUnknownPort": "false",
    "status":
    {
        "response_code": 200,
        "msg": "OK",
        "more_info": "",
        "error_code": 0
    }
}

```

Ancillary Request Example

An example using an ancillary request parameter to determine if the passed family type is snooping supported or not.

```

curl -GET -d "family=igmp&fields=isFamilyTypeSupported"
10.10.10.1:8080/open/v1/snoopingconfig

```

browser

```

http://10.10.10.1:8080/open/v1/snoopingconfig?fields=
isFamilyTypeSupported,adminMode&family=igmp

```

```

{
    "adminMode": "enabled",
    "family": "igmp",
    "isFamilyTypeSupported": "true",
    "status":
    {
        "response_code": 200,
        "msg": "OK",
        "more_info": "",
        "error_code": 0
    }
}

```

Error Request Example

```

curl -GET -d "fields=isFamilyTypeSupported"
10.10.10.1:8080/open/v1/snoopingconfig

```

browser

```

http://10.10.10.1:8080/open/v1/snoopingconfig?fields=
isFamilyTypeSupported,adminMode

```

Response Example

```
{
  "status":
  {
    "response_code": 400,
    "msg": "Bad Request",
    "more_info": "Invalid parameter",
    "error_code": -22
  }
}
```

Snooping configuration UPDATE

PUT /open/v1/snoopingconfig

Use this call to update various properties of the switch snooping configuration.

Request Details

Table 4-78. Snooping configuration UPDATE Request Details

Parameter	Type	Description
family	string	Snooping family (igmp/mld). The value is mandatory.
adminMode	string	IGMP/MLD Admin mode (enabled/disabled).
proxyQuerierAdminMode	string	IGMP/MLD Proxy Querier Admin mode (enabled/disabled).
floodAllUnknownPort	string	Flood unknown multicast traffic to all ports (true/false).
entriesFlush	—	Request to flush all Snooping entries and multicast router entries.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200 and 400.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-79. Snooping configuration UPDATE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Config update successful.
-22	OPEN_E_PARAM	Error in parameters passed.
-26	OPEN_E_FAIL	Config update failed.

Config Request Example

Enable IGMP Admin mode

```
curl -GET -XPUT -d "family=igmp&adminMode=enabled"
10.10.10.1:8080/open/v1/snoopingconfig
```

browser

```
http://10.10.10.1:8080/open/v1/snoopingconfig?method=
put&adminMode=enabled&family=igmp
```

Response Example

```
{
  "status":
  {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Error Example

```
curl -GET -XPUT -d "adminMode=enabled"
10.10.10.1:8080/open/v1/snoopingconfig
```

browser

```
http://10.10.10.1:8080/open/v1/snoopingconfig?method=
put&adminMode=enabled
```

Response Example

```
{
  "status":
  {
    "response_code": 400,
    "msg": "Bad Request",
    "more_info": "Invalid arguments",
    "error_code": -22
  }
}
```

Snooping configuration DELETE

Snooping configuration does not support delete semantics.

Snooping Interfaces

Snooping Interface CREATE

Note that Snooping Interface cannot be created.

Snooping Interface GET

```
GET /open/v1/snoopinginterfaces
```

Retrieves Snooping Interfaces.

Request Details

Table 4-80. Snooping Interface GET Request Details

Parameter	Type	Description
interface	slot/port	Interface name. Identifies the port.
family	igmp/mld	Snooping family. The value is mandatory.
fields	—	Partial response syntax.
fields	fastLeaveAdminMode	Fast-Leave Admin mode for the specified interface.
—	groupMembershipInterval	IGMP group membership interval for the specified interface.
—	intfMode	IGMP Snooping mode for the specified interface.
—	proxyQuerierMode	Proxy Querier Admin mode for the specified interface.
—	responseTime	IGMP query response time for the specified interface.
—	IsValidIntf	Ancillary request parameter to determine is interface a valid IGMP snooping Interface.
offset	integer	Denotes beginning index for pagination. Default is 1. This parameter is not valid for single interface retrievals.
limit	integer	Denotes total retrieval count. Default is 10. This parameter is not valid for single interface retrievals.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, 404, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-81. Snooping Interface GET Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Operation Fail.

Request Example

```
curl -GET -d "family=igmp"
10.10.10.1:8080/open/v1/snoopinginterfaces

browser
http://10.10.10.1:8080/open/v1/snoopinginterfaces?family=igmp
```

Response Example

```
{
  "snoopinginterfaces": [
    {
      "interface": "0/1",
      "family": "igmp",
      "fastLeaveAdminMode": "enabled",
      "groupMembershipInterval": 10,
      "intfMode": "enabled",
      "proxyQuerierMode": "enabled",
      "responseTime": 10
    },
    ... /** more records here **/
    {
      "interface": "0/64",
      "family": "igmp",
```

```

        "fastLeaveAdminMode": "enabled",
        "groupMembershipInterval": 10,
        "intfMode": "enabled",
        "proxyQuerierMode": "enabled",
        "responseTime": 10
    }
],
"status":
{
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
}
}

```

Multiple Snooping Interface Partial Request Example

The following example request limits the output to the first interface.

```

curl -GET -d "limit=1&family=igmp"
10.10.10.1:8080/open/v1/snoopinginterfaces

```

browser

```

http://10.10.10.1:8080/open/v1/snoopinginterfaces?limit=
1&family=igmp

```

Response Example

```

{
    "snoopinginterfaces": [
        {
            "interface": "0/1",
            "family": "igmp",
            "fastLeaveAdminMode": "enabled",
            "groupMembershipInterval": 10,
            "intfMode": "enabled",
            "proxyQuerierMode": "enabled",
            "responseTime": 10
        },
    ],
    "status":
    {
        "response_code": 200,
        "msg": "OK",
        "more_info": "",
    }
}

```

```

        "error_code": 0
    }
}

```

Single Snooping Interface Request Example

Single Snooping Interface retrievals are specified by using the interface parameter. All the available Snooping Interface resources are retrieved by default.

The following example requests Snooping Interface located in slot 0, port 9.

```

curl -GET -d "interface=0/9&family=igmp"
10.10.10.1:8080/open/v1/snoopinginterfaces

```

browser

```

http://10.10.10.1:8080/open/v1/snoopinginterfaces?interface=
0/9&family=igmp

```

Response Example

```

{
  "snoopinginterfaces": [
    {
      "interface": "0/9",
      "family": "igmp",
      "fastLeaveAdminMode": "enabled",
      "groupMembershipInterval": 10,
      "intfMode": "enabled",
      "proxyQuerierMode": "enabled",
      "responseTime": 10
    },
  ],
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}

```

Single Snooping Interface Partial Request Example

The response can be reduced by using the partial response syntax as denoted by the 'fields' parameter.

```
curl -GET -d "interface=0/1&fields=
family,groupMembershipInterval,responseTime"
10.10.10.1:8080/open/v1/snoopinginterfaces
```

browser

```
http://10.10.10.1:8080/open/v1/snoopinginterfaces?interface=
0/1&family=igmp&fields=groupMembershipInterval,responseTime
```

Response Example

```
{
  "snoopinginterfaces": [
    {
      "interface": "0/1",
      "family": "igmp",
      "groupMembershipInterval": 10,
      "responseTime": 10
    },
  ],
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Error Example

```
curl -GET -d "interface=5/1&family=igmp"
10.10.10.1:8080/open/v1/snoopinginterfaces
```

browser

```
http://10.10.10.1:8080/open/v1/snoopinginterfaces?method=
get&interface=5/1&family=igmp
```

Response Example

```
{
  "status": {
    "response_code": 404,
    "msg": "Not Found",
    "more_info": "",
    "error_code": -29
  }
}
```

Snooping Interface UPDATE

```
PUT /open/v1/snoopinginterfaces
```

Use this call to update an existing Snooping Interface configuration on the switch. The Interface name is mandatory.

Request Details

Table 4-82. Snooping Interface UPDATE Request Details

Parameter	Type	Description
interface	string	Interface name in slot/port format. Identifies the interface. The interface name is mandatory for all updates.
family	string	IGMP Snooping Address family (igmp/mld). The family type is mandatory for all updates.
fastLeaveAdminMode	string	Fast-Leave Admin mode for the specified interface (enabled/disabled).
groupMembershipInterval	integer	IGMP group membership interval for the specified interface.
intfMode	string	IGMP Snooping mode for the specified interface (enabled/disabled).
proxyQuerierMode	string	Proxy Querier Admin mode for the specified interface (enabled/disabled).
responseTime	integer	IGMP query response time for the specified interface.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, 404, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-83. Snooping Interface UPDATE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Operation Fail.
-29	OPEN_E_NOT_FOUND	Parameter not found.

Update Snooping Interface Request Example

The following example enables the snooping mode and sets response time to 100 for Interface located in slot 0, port 1.

```
curl -GET -XPUT -d "interface=0/1&intfMode=
enabled&responseTime=100&family=igmp"
10.10.10.1:8080/open/v1/snoopinginterfaces

browser

http://10.10.10.1:8080/open/v1/snoopinginterfaces?method=
put&interface=0/1&intfMode=enabled&responseTime=100&family=
igmp
```

Response Example

```
{
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

```
}
```

Error Example

```
curl -GET -XPUT -d "interface=0/1&intfMode=
enabled&responseTime=100"
10.10.10.1:8080/open/v1/snoopinginterfaces
```

browser

```
http://10.10.10.1:8080/open/v1/snoopinginterfaces?method=
put&interface=0/1&intfMode=enabled&responseTime=100
```

Response Example

```
{
  "status":
  {
    "response_code": 400,
    "msg": "Bad Request",
    "more_info": "Invalid arguments",
    "error_code": -22
  }
}
```

Snooping Interface DELETE

Note that Snooping Interface cannot be deleted. You can, however, reconfigure or UPDATE their parameters using PUT commands.

snoopingqueriers

snoopingquerier CREATE

This component does not support create semantics

snoopingquerier READ

```
GET /open/v1/snoopingqueriers
```

Retrieves all of the variables that are associated with the snooping querier.

Request Details

Table 4-84. snoopingquerier READ Request Details

Parameter	Type	Description
family	igmp/mld	Snooping Address family (mandatory parameter).
vlanId	integer	VLAN ID (mandatory parameter).
fields	—	Partial response syntax.
fields	address	Snooping querier configured address of a snoop instance.
—	adminMode	Snooping querier admin mode of a snoop instance.
—	expiryInterval	Expiry interval of a snoop instance.
—	lastQuerierAddress	The detected last querier address for specified VLAN ID of a snoop instance.
—	lastQuerierVersion	The detected last querier's version for specified VLAN ID of a snoop instance. This field can be empty.
—	operMaxRespTime	The operational value of max response time for specified VLAN ID of a snoop instance. This field can be empty.
—	operState	The operational state of querier for specified VLAN ID of a snoop instance.
—	operVersion	The operational version of querier for specified VLAN ID of a snoop instance.
—	queryInterval	Snooping querier query interval of a snoop instance.

Table 4-84. snoopingquerier READ Request Details (Continued)

Parameter	Type	Description
—	querierVersion	Snooping querier configured version of a snoop instance.
—	vlanAddress	The snooping querier configured address for specified VLAN ID of a snoop instance.
—	vlanElectionMode	Configured snooping querier election mode for the specified VLAN ID of a snoop instance.
—	vlanMode	Configured snooping querier mode for the specified VLAN of a snoop instance.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-85. snoopingquerier READ Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error
-22	OPEN_E_PARAM	Invalid request.
-26	OPEN_E_FAIL	Operation failed

Request Example

```
curl -GET -d "family=igmp&vlanId=1"
10.10.10.1:8080/open/v1/snoopingqueriers

browser

http://10.10.10.1:8080/open/v1/snoopingqueriers?family=
igmp&vlanId=1
```

Response Example

```
{
  "family": "igmp",
  "vlanId": 1,
  "address": "10.11.12.13",
  "adminMode": "enabled",
  "family": "igmp",
  "expiryInterval": 20,
  "lastQuerierAddress": "10.11.12.14",
  "lastQuerierVersion": 3,
  "operMaxRespTime": 35,
  "operState": "querier",
  "operVersion": 5,
  "queryInterval": 15,
  "querierVersion": 3,
  "vlanAddress": "10.11.12.15",
  "vlanElectionMode": "enabled",
  "vlanMode": "enabled",
  "status":
  {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Partial Request Example

```
curl -GET -d "family=igmp&vlanId=1&fields=address,adminMode"
10.10.10.1:8080/open/v1/snoopingqueriers
```

browser

```
http://10.10.10.1:8080/open/v1/snoopingqueriers?family=
igmp&vlanId=1&fields=address,adminMode
```

```
{
  "family": "igmp",
  "vlanId": 1,
  "address": "10.11.12.13",
  "adminMode": "enabled",
  "status":
  {
    "response_code": 200,
    "msg": "OK",
```

```

        "more_info": "",
        "error_code": 0
    }
}

```

Error Request Example

```

curl -GET 10.10.10.1:8080/open/v1/snoopingqueriers
browser
http://10.10.10.1:8080/open/v1/snoopingqueriers

```

Response Example

```

{
    "status":
    {
        "response_code": 400,
        "msg": "Bad Request",
        "more_info": "Invalid parameter",
        "error_code": -22
    }
}

```

snoopingquerier UPDATE

```
PUT /open/v1/snoopingqueriers
```

Use this call to update various properties of the switch.

Request Details

Table 4-86. snoopingquerier UPDATE Request Details

Parameter	Type	Description
family	string	IGMP Snooping Address family (igmp/mld).
vlanId	integer	VLAN ID.
address	string	Snooping querier configured address of a snoop instance.
adminMode	string	Snooping querier admin mode of a snoop instance (enabled/disabled).
expiryInterval	integer	Expiry interval of a snoop instance.

Table 4-86. snoopingquerier UPDATE Request Details (Continued)

Parameter	Type	Description
queryInterval	integer	Snooping querier query interval of a snoop instance.
querierVersion	integer	Snooping querier configured version of a snoop instance.
vlanAddress	string	The snooping querier configured address for specified VLAN ID of a snoop instance.
vlanElectionMode	string	Configured snooping querier election mode for the specified VLAN ID of a snoop instance.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-87. snoopingquerier UPDATE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Config update successful.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Error in parameters passed.
-26	OPEN_E_FAIL	Config update failed.

Config Request Example

Enable IGMP Admin mode

```
curl -GET -XPUT -d "family=igmp&vlanId=1&adminMode=enabled"
10.10.10.1:8080/open/v1/snoopingqueriers

browser

http://10.10.10.1:8080/open/v1/snoopingqueriers?method=
put&adminMode=enabled&vlanId=1&family=igmp
```

Response Example

```
{
  "status":
  {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Error Example

```
curl -GET -XPUT -d "adminMode=enabled"
10.10.10.1:8080/open/v1/snoopingqueriers
```

browser

```
http://10.10.10.1:8080/open/v1/snoopingqueriers?method=
put&adminMode=enabled
```

Response Example

```
{
  "status":
  {
    "response_code": 400,
    "msg": "Bad Request",
    "more_info": "Invalid arguments",
    "error_code": -22
  }
}
```

snoopingquerier DELETE

This component does not support delete semantics.

snoopingssms

snoopingssm CREATE

This component does not support create semantics.

snoopingssm READ

```
GET /open/v1/snoopingssms
```

Retrieves all available snoopingssm entries.

Request Details

Table 4-88. snoopingssm READ Request Details

Parameter	Type	Description
family	igmp/mld	Snooping Address family (mandatory parameter).
entryType	mfdb/group/srclist	Type of query (mandatory parameter).
fields		Partial response syntax.
fields	mfdbCurrEntries	The current number of entries in the MFDB table.
	mfdbMaxTableEntries	Maximum number of entries that the SSM MFDB table can hold for a given family.
	mfdbMostEntries	Most number of entries ever in the SSM MFDB table for a given family.
	incPortMask	Include port list.
	excPortMask	Exclude port list.
offset	integer	Denotes beginning index for pagination. Default is 1.
limit	integer	Denotes total retrieval count. Default is 10.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-89. snoopingssm READ Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Operation Fail.

Multiple Request Example

Note that pagination is implicit.

```
curl -GET -d "family=igmp&entryType=mfdb"
10.10.10.1:8080/open/v1/snoopingssms
```

browser

```
http://10.10.10.1:8080/open/v1/snoopingssms?family=
igmp&entryType=mfdb
```

Response Example

```
{
  "family": "igmp",
  "mfdbCurrEntries": 3,
  "mfdbMaxTableEntries": 100,
  "mfdbMostEntries": 30,
  "mfdbEntry": [
    {
      "excPortMask": "0/1, 0/2",
      "incPortMask": "0/3",
      "vlanId": 1
    },
    {
      "excPortMask": "1/1, 1/2",
      "incPortMask": "1/3",
      "vlanId": 2
    },
    {
      "excPortMask": "2/1, 2/2",
```

```

        "incPortMask": "2/3",
        "vlanId": 3
    },
    ],
    "status": {
        "response_code": 200,
        "msg": "OK",
        "more_info": "",
        "error_code": 0
    }
}

```

Partial Request Example

The following example request limits the output to 2.

```

curl -GET -d "family=igmp&entryType=mfdb&limit=2"
10.10.10.1:8080/open/v1/snoopingssms

```

browser

```

http://10.10.10.1:8080/open/v1/snoopingssms?family=
igmp&entryType=mfdb&limit=2

```

Response Example

```

{
    "family": "igmp",
    "mfdbCurrEntries": 3,
    "mfdbMaxTableEntries": 100,
    "mfdbMostEntries": 30,
    "mfdbEntry": [
        {
            "excPortMask": "0/1, 0/2",
            "incPortMask": "0/3",
            "vlanId": 1
        },
        {
            "excPortMask": "1/1, 1/2",
            "incPortMask": "1/3",
            "vlanId": 2
        }
    ],
    "status": {
        "response_code": 200,
        "msg": "OK",
        "more_info": "",

```

```
        "error_code": 0
    }
}
```

Partial Request Example

```
curl -GET -d "family=igmp&entryType=mfdb&fields=
mfdbCurrEntries" 10.10.10.1:8080/open/v1/snoopingssms
```

browser

```
http://10.10.10.1:8080/open/v1/snoopingssms?family=
igmp&entryType=mfdb&limit=2&fields=mfdbCurrEntries
```

Response Example

```
{
  "mfdbCurrEntries": 3,
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Error Request Example

```
curl -GET -d "entryType=mfdb"
10.10.10.1:8080/open/v1/snoopingssms
```

browser

```
http://10.10.10.1:8080/open/v1/snoopingssms?entryType=mfdb
```

Response Example

```
{
  "status":
  {
    "response_code": 400,
    "msg": "Bad Request",
    "more_info": "Invalid arguments",
    "error_code": -22
  }
}
```

snoopingssm UPDATE

This component does not support update semantics.

snoopingssm DELETE

This component does not support delete semantics.

Snooping VLANs

Snooping VLAN CREATE

Note that Snooping VLAN cannot be created.

Snooping VLAN GET

`GET /open/v1/snoopingvlans`

Retrieves Snooping VLANs.

Request Details

Table 4-90. Snooping VLAN GET Request Details

Parameter	Type	Description
vlanId	integer	VLAN ID. VLAN ID is mandatory.
family	igmp/mld	Snooping family. The family type is mandatory.
fields	—	Partial response syntax.
fields	fastLeaveMode	IGMP/MLD snooping fast leave mode for the specified VLAN.
—	groupMembershipInterval	IGMP/MLD snooping group membership interval for the specified VLAN.
—	maxResponseTime	IGMP/MLD snooping maximum response time for the specified VLAN.
—	vlanMode	IGMP/MLD Snooping mode for the specified VLAN.
—	proxyQuerierMode	Proxy Querier Admin mode for the specified VLAN.
—	reportSuppMode	Snooping report suppression mode for the specified VLAN.
—	IsValidVlan	Ancillary request parameter to determine is VLAN id eligible for snooping configuration.
offset	integer	Denotes beginning index for pagination. Default is 1. This parameter is not valid for single VLAN retrievals.

Table 4-90. Snooping VLAN GET Request Details (Continued)

Parameter	Type	Description
limit	integer	Denotes total retrieval count. Default is 10. This parameter is not valid for single VLAN retrievals.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, 404, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-91. Snooping VLAN GET Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Operation fail.

Request Example

```
curl -GET -d "family=igmp"
10.10.10.1:8080/open/v1/snoopingvlans

browser

http://10.10.10.1:8080/open/v1/snoopingvlans?family=igmp
```

Response Example

```
{
  "snoopingvlans": [
    {
      "vlanId": "1",
      "family": "igmp",
      "fastLeaveMode": "enabled",
      "vlanMode": "enabled",
```

```

        "reportSuppMode": "enabled",
        "proxyQuerierMode": "enabled",
        "groupMembershipInterval": 10,
        "maxResponseTime": 10
    },
    ... /** more records here **/
    {
        "vlanId": "10",
        "family": "igmp",
        "fastLeaveMode": "enabled",
        "vlanMode": "enabled",
        "reportSuppMode": "enabled",
        "proxyQuerierMode": "enabled",
        "groupMembershipInterval": 10,
        "maxResponseTime": 10
    }
],
"status":
{
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
}
}

```

Single Snooping VLAN Request Example

The following example request limits the output to the first VLAN.

```

curl -GET -d "family=igmp&limit=1"
10.10.10.1:8080/open/v1/snoopingvlans

```

browser

```

http://10.10.10.1:8080/open/v1/snoopingvlans?family=
igmp&limit=1

```

Response Example

```

{
    "snoopingvlans": [
        {
            "vlanId": "1",
            "family": "igmp",
            "fastLeaveMode": "enabled",
            "vlanMode": "enabled",

```

```

        "reportSuppMode": "enabled",
        "proxyQuerierMode": "enabled",
        "groupMembershipInterval": 10,
        "maxResponseTime": 10
    },
],
"status":
{
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
}
}

```

Single Snooping VLAN Request Example

Single Snooping VLAN retrievals are specified by using the VLAN parameter. All the available Snooping VLAN resources are retrieved by default.

The following is an example request Snooping VLAN 2.

```

curl -GET -d "family=igmp&vlanId=2"
10.10.10.1:8080/open/v1/snoopingvlangs

```

browser

```

http://10.10.10.1:8080/open/v1/snoopingvlangs?family=
igmp&vlanId=2

```

Response Example

```

{
    "snoopingvlangs": [
        {
            "vlanId": "2",
            "family": "igmp",
            "fastLeaveMode": "enabled",
            "vlanMode": "enabled",
            "reportSuppMode": "enabled",
            "proxyQuerierMode": "enabled",
            "groupMembershipInterval": 10,
            "maxResponseTime": 10
        }
    ],
    "status":
    {
        "response_code": 200,

```

```

        "msg": "OK",
        "more_info": "",
        "error_code": 0
    }
}

```

Single Snooping VLAN Partial Request Example

The response can be reduced by using the partial response syntax as denoted by the 'fields' parameter.

```

curl -GET -d "family=igmp&vlanId=1&fields=
vlanMode,maxResponseTime"
10.10.10.1:8080/open/v1/snoopingvlans

```

browser

```

http://10.10.10.1:8080/open/v1/snoopingvlans?family=
igmp&vlanId=1&fields=vlanMode,maxResponseTime

```

Response Example

```

{
  "snoopingvlans": [
    {
      "vlanId": "1",
      "family": "igmp",
      "vlanMode": "enabled",
      "maxResponseTime": 10
    },
  ],
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}

```

Error Example

```

curl -GET -d "vlanId=2" 10.10.10.1:8080/open/v1/snoopingvlans

```

browser

```

http://10.10.10.1:8080/open/v1/snoopingvlans?vlanId=2

```

Response Example

```
{
  "status": {
    "response_code": 400,
    "msg": "Bad Request",
    "more_info": "Invalid parameter",
    "error_code": -22
  }
}
```

Snooping VLAN UPDATE

```
PUT /open/vl/snoopingvlans
```

Use this call to update an existing Snooping VLAN configuration on the switch. The VLAN ID and family are mandatory.

Request Details

Table 4-92. Snooping VLAN UPDATE Request Details

Parameter	Type	Description
vlanId	integer	VLAN ID. The VLAN ID is mandatory for all updates.
family	string	Snooping family (igmp/mld). The family type is mandatory for all updates.
fastLeaveMode	string	Snooping fast leave mode for the specified VLAN (enabled/disabled).
vlanMode	string	Snooping mode for the specified VLAN (enabled/disabled).
proxyQuerierMode	string	Proxy Querier Admin mode for the specified VLAN (enabled/disabled).
reportSuppMode	string	Report suppression mode for the specified VLAN (enabled/disabled).
groupMembershipInterval	integer	Snooping group membership interval for the specified VLAN.
maxResponseTime	integer	Snooping maximum response time for the specified VLAN.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, 404, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-93. Snooping VLAN UPDATE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Operation Fail.
-29	OPEN_E_NOT_FOUND	Parameter not found.

Update Snooping VLAN Request Example

The following example enables the snooping mode and sets maximum response time to 100 for VLAN 1.

```
curl -GET -XPUT -d "vlanId=1&vlanMode=enabled&maxResponseTime=100&family=igmp" 10.10.10.1:8080/open/v1/snoopingvlans  
  
browser  
  
http://10.10.10.1:8080/open/v1/snoopingvlans?method=put&vlanId=1&vlanMode=enabled&maxResponseTime=100&family=igmp
```

Response Example

```
{  
  "status": {  
    "response_code": 200,  
    "msg": "OK",  
    "more_info": "",  
    "error_code": 0  
  }  
}
```

Error Example

```
curl -GET -XPUT -d "vlanMode=enabled&maxResponseTime=100&family=igmp" 10.10.10.1:8080/open/v1/snoopingvlans
```

browser

```
http://10.10.10.1:8080/open/v1/snoopingvlans?method=put&vlanMode=enabled&maxResponseTime=100&family=igmp
```

Response Example

```
{
  "status":
  {
    "response_code": 400,
    "msg": "Bad Request",
    "more_info": "Invalid arguments",
    "error_code": -22
  }
}
```

Snooping VLAN DELETE

Note that Snooping VLAN cannot be deleted. You can, however, reconfigure or UPDATE their parameters using PUT commands.

stormcontrolports

stormcontrolports implements a RESTful API for the OpEN Overlay API.

Storm Control Interface CREATE

Note that Storm Control Interface cannot be created. You can, however, reconfigure or UPDATE their parameters using PUT commands.

Storm Control Interface GET

```
GET /open/v1/stormcontrolports
```

Retrieves Storm Control Interface.

Request Details

Table 4-94. Storm Control Interface GET Request Details

Parameter	Type	Description
interface	string	Interface identifier in slot/port format.
fields	—	Partial response syntax.
fields	modeUnicast	Storm control mode settings (enabled/disabled) for interface.
—	modeBroadcast	Broadcast storm control mode settings (enabled/disabled) for interface.
—	modeMulticast	Multicast storm control mode settings (enabled/disabled) for interface.
—	thresholdUnicast	Unicast Storm Control mode threshold.
—	thresholdBroadcast	Broadcast Storm Control mode threshold.
—	thresholdMulticast	Multicast Storm Control mode threshold.
—	rateUnitUnicast	Unicast rate unit for threshold (pps/percent).
—	rateUnitBroadcast	Broadcast rate unit for threshold (pps/percent).
—	rateUnitMulticast	Multicast rate unit for threshold (pps/percent).
offset	integer	Denotes beginning index for pagination. Default is 1. This parameter is not valid for single interface retrievals.

Table 4-94. Storm Control Interface GET Request Details (Continued)

Parameter	Type	Description
limit	integer	Denotes total retrieval count. Default is 10. This parameter is not valid for single interface retrievals.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, 404, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-95. Storm Control Interface GET Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error.
-22	OPEN_E_PARAM	Invalid parameter.
-24	OPEN_E_EXISTS	Already exists.
-26	OPEN_E_FAIL	Operation Fail.

Request Example

```
curl -GET 10.10.10.1:8080/open/v1/stormcontrolports
browser
http://10.10.10.1:8080/open/v1/stormcontrolports
```

Response Example

```
{
  "stormcontrolports": [
    {
      "interface": "0/1",
      "modeUnicast": "enabled",
      "modeBroadcast": "enabled",
      "modeMulticast": "enabled",
```

```

        "thresholdUnicast": 7440000,
        "thresholdBroadcast": 7440000,
        "thresholdMulticast": 7440000,
        "rateUnitUnicast": "pps"
        "rateUnitBroadcast": "pps"
        "rateUnitMulticast": "pps"
    },
    ... /** more records here **/
    {
        "interface": "0/64",
        "modeUnicast": "enabled",
        "modeBroadcast": "enabled",
        "modeMulticast": "enabled",
        "thresholdUnicast": 7440000,
        "thresholdBroadcast": 7440000,
        "thresholdMulticast": 7440000,
        "rateUnitUnicast": "pps"
        "rateUnitBroadcast": "pps"
        "rateUnitMulticast": "pps"
    }
],
"status":
{
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
}
}

```

Multiple Storm Control Interface Configuration Partial Request Example

The following example request limits the output to the first two interfaces.

```

curl -GET -d "limit=2"
10.10.10.1:8080/open/v1/stormcontrolports

```

browser

```

http://10.10.10.1:8080/open/v1/stormcontrolports?limit=2

```

Response Example

```

{
    "stormcontrolports": [
        {
            "interface": "0/1",

```

```

        "modeUnicast": "enabled",
        "modeBroadcast": "enabled",
        "modeMulticast": "enabled",
        "thresholdUnicast": 7440000,
        "thresholdBroadcast": 7440000,
        "thresholdMulticast": 7440000,
        "rateUnitUnicast": "pps"
        "rateUnitBroadcast": "pps"
        "rateUnitMulticast": "pps"
    },
    {
        "interface": "0/2",
        "modeUnicast": "enabled",
        "modeBroadcast": "disabled",
        "modeMulticast": "enabled",
        "thresholdUnicast": 7440000,
        "thresholdBroadcast": 7440000,
        "thresholdMulticast": 7440000,
        "rateUnitUnicast": "pps"
        "rateUnitBroadcast": "pps"
        "rateUnitMulticast": "pps"
    }
],
"status":
{
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
}
}

```

Single Storm Control Interface Request Example

Single Storm Control Interface retrievals are specified by using the interface parameter. All the available Storm Control Interface resources are retrieved by default.

The following is an example of a request Storm Control Interface located in slot 0, port 9.

```

curl -GET -d "interface=0/9"
10.10.10.1:8080/open/v1/stormcontrolports

```

browser

```

http://10.10.10.1:8080/open/v1/stormcontrolports?interface=0/9

```

Response Example

```
{
  "stormcontrolports": [
    {
      "interface": "0/9",
      "modeUnicast": "enabled",
      "modeBroadcast": "enabled",
      "modeMulticast": "enabled",
      "thresholdUnicast": 7440000,
      "thresholdBroadcast": 7440000,
      "thresholdMulticast": 7440000,
      "rateUnitUnicast": "pps"
      "rateUnitBroadcast": "pps"
      "rateUnitMulticast": "pps"
    }
  ],
  "status":
  {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Single Storm Control Interface Partial Request Example

The response can be reduced by using the partial response syntax as denoted by the *fields* parameter.

In the following example, a specific Storm Control Interface request, while limiting the results to enabled multicast storm control.

```
curl -GET -d "interface=0/1&fields=
modeMulticast,thresholdMulticast,rateUnitMulticast"
10.10.10.1:8080/open/v1/stormcontrolports
```

browser

```
http://10.10.10.1:8080/open/v1/stormcontrolports?interface=
0/1&fields=modeMulticast,thresholdMulticast,rateUnitMulticast
```

Response Example

```
{
```

```

"stormcontrolports": [
  {
    "interface": "0/1",
    "modeMulticast": "enabled",
    "thresholdMulticast": 7440000,
    "rateUnitMulticast": "pps"
  }
],
"status": {
  "response_code": 200,
  "msg": "OK",
  "more_info": "",
  "error_code": 0
}
}

```

Error Example

```

curl -GET -d "interface=5/1"
10.10.10.1:8080/open/v1/stormcontrolports

```

browser

```

http://10.10.10.1:8080/open/v1/stormcontrolports?method=
get&interface=5/1

```

Response Example

```

{
  "status": {
    "response_code": 404,
    "msg": "Not Found",
    "more_info": "",
    "error_code": -29
  }
}

```

Storm Control Interface UPDATE

```

PUT /open/v1/stormcontrolports

```

Use this call to update an existing Storm Control Interface configuration on the switch. The Interface name is mandatory.

Request Details

Table 4-96. Storm Control Interface UPDATE Request Details

Parameter	Type	Description
interface	string	Interface name in slot/port format. Identifies the interface. The interface name is mandatory for all updates.
modeUnicast	string	Unicast Storm Control mode (enabled/disabled).
modeBroadcast	string	Broadcast Storm Control mode (enabled/disabled).
modeMulticast	string	Multicast Storm Control mode (enabled/disabled).
thresholdUnicast	integer	Unicast Storm Control threshold value (rateUnitUnicast is mandatory).
thresholdBroadcast	integer	Broadcast Storm Control threshold value (rateUnitBroadcast is mandatory).
thresholdMulticast	integer	Multicast Storm Control threshold value (rateUnitMulticast is mandatory).
rateUnitUnicast	string	Unicast rate unit for threshold type (pps/percent).
rateUnitBroadcast	string	Broadcast rate unit for threshold type (pps/percent).
rateUnitMulticast	string	Multicast rate unit for threshold type (pps/percent).

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, 404, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-97. Storm Control Interface UPDATE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Internal error.

Table 4-97. Storm Control Interface UPDATE Internal Error Codes (Continued)

Error Code	Code Name	Description
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Operation Fail.
-28	OPEN_E_UNAVAIL	Parameter or feature is not supported.
-29	OPEN_E_NOT_FOUND	Parameter not found.

Update Storm Control Interface Request Example

The following example enables the storm control unicast mode and sets the threshold to 50% for Interface located in slot 0, port 1.

```
curl -GET -XPUT -d "interface=0/1&thresholdUnicast=
50&rateUnitUnicast=percent&modeUnicast=enabled"
10.10.10.1:8080/open/v1/stormcontrolports
```

browser

```
http://10.10.10.1:8080/open/v1/stormcontrolports?method=
put&interface=0/1&thresholdUnicast=50&rateUnitUnicast=
percent&modeUnicast=enabled
```

Response Example

```
{
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Storm Control Interface DELETE

Note that Storm Control Interfaces cannot be deleted. You can, however, reconfigure or UPDATE their parameters using PUT commands.

swmirror

swmirror implements a RESTful API providing Port Mirroring.

SWMIRROR CREATE

POST /open/v1/swmirror

Request Details

Table 4-98. SWMIRROR CREATE Request Details

Parameter	Type	Description
remoteSpanVlan	integer	Used to add a VLAN as an RSPAN VLAN.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-99. SWMIRROR CREATE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Request failed.
-28	OPEN_E_UNAVAIL	Parameter not supported.

Request Example

```
curl -GET -XCREATE 10.10.10.1:8080/open/v1/swmirror -d
'remoteSpanVlan=7'
```

browser

```
http://10.10.10.1:8080/open/v1/swmirror?method=
post&remoteSpanVlan=7
```

Response Example

```
{
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

SWMIRROR READ

```
GET /open/v1/swmirror?session=1
```

Retrieves the various attributes supported by this resource.

Request Details

Table 4-100. SWMIRROR READ Request Details

Parameter	Type	Description
session	integer	Session number. Required for interface and reserveState.
interface	string	Required only for ancillary request to retrieve source port direction (session).
fields	reserveState	Retrieves session reserved state (enabled/disabled).
—	remoteSpanVlans	Remote span VLAN (empty list if none).
—	sourcePortDirection	Source port direction (unconfigured, bidirectional, ingress, egress, sFlow).

NOTE: If no parameters are specified, a 400 Invalid parameter will be returned.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-101. SWMIRROR READ Internal Error Code

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Request failed.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Request failed.
-28	OPEN_E_UNAVAIL	Parameter not supported.
-30	OPEN_E_EMPTY	Returned value is not configured.

Request Example

```
curl -GET 10.10.10.1:8080/open/v1/swmirror -d 'session=1'
browser
http://10.10.10.1:8080/open/v1/swmirror?session=1
```

Response Example

```
{
  "sourcePortDirection": "unconfigured",
  "reserveState" : "enabled",
  "remoteSpanVlans": [
    1,
    2,
    3
  ],
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
```

```
    "error_code": 0
  }
}
```

Error Example

```
curl -GET -d "test=0" 10.10.10.1:8080/open/v1/swmirror
browser
http://10.10.10.1:8080/open/v1/swmirror?method=get&test=0
```

Response Example

Returns an appropriate status message.

```
{
  "status": {
    "response_code": 400,
    "msg": "Bad Request",
    "more_info": "Invalid parameter",
    "error_code": -22
  }
}
```

SWMIRROR UPDATE

```
PUT /open/v1/swmirror
```

Use this call to update various mirroring settings. Use GET (see preceding example) to retrieve swmirror settings.

Request Details

Table 4-102. SWMIRROR UPDATE Request Details

Parameter	Type	Description
session	integer	Session number. Required for reserveState.
reserveState	string	Session reserve state.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-103. SWMIRROR UPDATE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Update successful.
-22	OPEN_E_PARAM	Error in parameters passed.
-26	OPEN_E_FAIL	Update failed.

Request Example

```
curl -GET -XPUT -d "reserveState=disabled&session=1"
10.10.10.1:8080/open/v1/swmirror

browser

http://10.10.10.1:8080/open/v1/swmirror?method=
put&reserveState=disabled&session=1
```

Response Example

```
{
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Error Example

```
curl -GET -XPUT -d "session=2"
10.10.10.1:8080/open/v1/swmirror

browser

http://10.10.10.1:8080/open/v1/swmirror?method=put&session=2
```

Response Example

```
{
  "status": {
```

```
    "response_code": 400,  
    "msg": "Bad Request",  
    "more_info": "Invalid argument"  
    "error_code": -22  
  }  
}
```

SWMIRROR DELETE

```
DELETE /open/v1/swmirror
```

Request Details

Table 4-104. SWMIRROR DELETE Request Details

Parameter	Type	Description
remoteSpanVlan	integer	Used to unconfigure remote SPAN VLAN.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-105. SWMIRROR DELETE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Request failed.
-28	OPEN_E_UNAVAIL	Parameter not supported.

Request Example

```
curl -GET -XDELETE 10.10.10.1:8080/open/v1/swmirror -d  
'remoteSpanVlan=7'
```

browser

```
http://10.10.10.1:8080/open/v1/swmirror?method=
delete&remoteSpanVlan=7
```

Response Example

```
{
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

swmonitor

swmonitor implements a RESTful API providing Session Monitor Configuration.

SWMONITOR CREATE

Note that swmonitor cannot be created. You can, however, reconfigure or UPDATE parameters using PUT commands.

SWMONITOR READ

```
GET /open/v1/swmonitor
```

Retrieves the various attributes supported by this resource.

Request Details

Table 4-106. SWMONITOR READ Request Details

Parameter	Type	Description
session	integer	Session number. Required for all attributes (except sessionCount).
fields	—	—
—	vlangs	List of monitored VLANs (if any - empty list).
—	srcType	Source mirror type.
—	dstType	Destination mirror type.
—	remoteSrcVlan	Remote source VLAN (or <i>unconfigured</i>).
—	remoteDstVlan	emote destination VLAN (or <i>unconfigured</i>).
—	reflectorPort	Reflector port (or <i>unconfigured</i>).
—	ipAclListNum	IP ACL list number (or <i>unavailable</i>).
—	macAclListName	MAC ACL list name (or <i>unavailable</i>).
—	sessionCount	The number of monitoring sessions.

NOTE: If no parameters are specified, 400 - Invalid parameter, will be returned.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-107. SWMONITOR READ Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Request failed.

Request Example

```
curl -GET 10.10.10.1:8080/open/v1/swmonitor -d 'session=1'
browser
http://10.10.10.1:8080/open/v1/swmonitor?method=get&session=1
```

Response Example

```
{
  "vlans" : [
    10
  ],
  "srcType" : 2,
  "dstType" : 1,
  "remoteSrcVlan": 1,
  "remoteDstVlan": 100,
  "reflectorPort": "0/5",
  "ipAclListNum": 100,
  "macAclListName" : "test",
  "sessionCount" : 4,

  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

```
}  
}
```

Error Example

```
curl -GET -d "test=0" 10.10.10.1:8080/open/v1/swmonitor  
browser  
http://10.10.10.1:8080/open/v1/swmonitor?method=get&test=0
```

Response Example

Returns an appropriate status message.

```
{  
  "status": {  
    "response_code": 400,  
    "msg": "Bad Request",  
    "more_info": "Invalid parameter",  
    "error_code": -22  
  }  
}
```

SWMONITOR UPDATE

```
PUT /open/v1/swmonitor
```

Use this call to update various mirroring settings. Use GET (see preceding example) to retrieve swmonitor settings.

Request Details

Table 4-108. SWMONITOR UPDATE Request Details

Parameter	Type	Description
session	integer	Session number. Required.
vlan	string	List of monitored VLANs (comma separated).
remoteSrcVlan	integer	Remote source VLAN.
remoteDstVlan	integer	Remote destination VLAN.
reflectorPort	string	Reflector port in slot/port format.
ipAclListNum	integer	IP ACL list number.
macAclListName	string	MAC ACL list name.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-109. SWMONITOR UPDATE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Update successful.
-21	OPEN_E_INTERNAL	Request failed.
-22	OPEN_E_PARAM	Error in parameters passed.
-26	OPEN_E_FAIL	Update failed.
-28	OPEN_E_UNAVAIL	Package is unavailable.
-31	OPEN_E_ERROR	ACL rule doesn't exist.

Request Example

```
curl -GET -XPUT -d "macAclListName=test&session=1"
10.10.10.1:8080/open/v1/swmonitor

browser

http://10.10.10.1:8080/open/v1/swmonitor?method=
put&macAclListName=test&session=1
```

Response Example

```
{
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Error Example

```
curl -GET -XPUT -d "session=20"
10.10.10.1:8080/open/v1/swmonitor
browser
http://10.10.10.1:8080/open/v1/swmonitor?method=put&session=20
```

Response Example

```
{
  "status": {
    "response_code": 400,
    "msg": "Bad Request",
    "more_info": "Invalid argument"
    "error_code": -22
  }
}
```

SWMONITOR DELETE

```
DELETE /open/v1/swmonitor
```

Use this call to remove various mirroring settings. Use GET (see preceding example) to retrieve swmonitor settings.

Request Details

Table 4-110. SWMONITOR DELETE Request Details

Parameter	Type	Description
session ^a	integer	Session number. Required.
vlangs	string	List of monitored VLANs (comma separated).
remoteSrcVlan	integer	Remote source VLAN.
remoteDstVlan	integer	Remote destination VLAN.
reflectorPort	string	Reflector port in slot/port format.
ipAclListNum	integer	IP ACL list number.
macAclListName	string	MAC ACL list name.

^a. Session cannot be deleted.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-111. SWMONITOR DELETE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Update successful.
-21	OPEN_E_INTERNAL	Request failed.
-22	OPEN_E_PARAM	Error in parameters passed.
-26	OPEN_E_FAIL	Update failed.
-28	OPEN_E_UNAVAIL	Package is unavailable.
-31	OPEN_E_ERROR	ACL rule doesn't exist.

Request Example

```
curl -GET -XDELETE -d "macAclListName=test&session=1"
10.10.10.1:8080/open/v1/swmonitor

browser

http://10.10.10.1:8080/open/v1/swmonitor?method=
delete&macAclListName=test&session=1
```

Response Example

```
{
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Error Example

```
curl -GET -XDELETE -d "session=20"  
10.10.10.1:8080/open/v1/swmonitor
```

browser

```
http://10.10.10.1:8080/open/v1/swmonitor?method=  
delete&session=20
```

Response Example

```
{  
  "status": {  
    "response_code": 400,  
    "msg": "Bad Request",  
    "more_info": "Invalid argument"  
    "error_code": -22  
  }  
}
```

swportmonitor

swportmonitor implements a RESTful API providing Port Monitor.

SWPORTMONITOR CREATE

POST /open/v1/swportmonitor

Request Details

Table 4-112. SWPORTMONITOR CREATE Request Details

Parameter	Type	Description
session	integer	Session number. Required.
srcPort	string	Source port in slot/port format.
type	string	Type of probe port [bidirectional/ingress/egress/sflow]. Required for source port.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-113. SWPORTMONITOR CREATE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Request failed.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Request failed.
-28	OPEN_E_UNAVAIL	Parameter not supported.

Request Example

```
curl -POST 10.10.10.1:8080/open/v1/swportmonitor -d 'session=1&srcPort=0/3&type=bidirectional'
```

browser

```
http://10.10.10.1:8080/open/v1/swportmonitor?method=post&session=1&srcPort=0/3&type=bidirectional
```

Response Example

```
{
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

SWPORTMONITOR READ

GET /open/v1/swportmonitor

Retrieves the various attributes supported by this resource.

Request Details

Table 4-114. SWPORTMONITOR READ Request Details

Parameter	Type	Description
session	integer	Session number. Required for all fields.
fields	srcPorts	List of source ports in unit/slot format (empty if any).
—	dstPort	Destination port in unit/slot format (or <i>unconfigured</i>).
—	mode	Port mirroring mode (enabled/disabled).
NOTE: If no parameters are specified, 400 - Invalid parameter will be returned.		

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-115. SWPORTMONITOR READ Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Request failed.
-28	OPEN_E_UNAVAIL	Parameter not supported.

Request Example

```
curl -GET 10.10.10.1:8080/open/v1/swportmonitor -d 'session=1'
browser
http://10.10.10.1:8080/open/v1/swportmonitor?method=
get&session=1
```

Response Example

```
{
  "srcPorts": [
    "0/1", "0/3", 0/4"
  ],
  "dstPort": "0/5",
  "mode" : "enabled"

  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Error Example

```
curl -GET -d "session=-99"
10.10.10.1:8080/open/v1/swportmonitor
browser
```

```
http://10.10.10.1:8080/open/v1/swportmonitor?method=
get&session=-99
```

Response Example

Returns an appropriate status message.

```
{
  "status": {
    "response_code": 400,
    "msg": "Bad Request",
    "more_info": "Invalid parameter",
    "error_code": -22
  }
}
```

SWPORTMONITOR UPDATE

```
PUT /open/v1/swportmonitor
```

Use this call to update various mirroring settings. Use GET (see preceding example) to retrieve swportmonitor settings.

Request Details

Table 4-116. SWPORTMONITOR UPDATE Request Details

Parameter	Type	Description
session	integer	Session number. Required (except configRemove).
mode	string	Monitor mode (enabled/disabled).
dstPort	string	Destination port in slot/port format.
configRemove	—	Request to remove current mirroring configuration.
sessionRemove	—	Request to remove current session.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-117. SWPORTMONITOR UPDATE Response Details

Error Code	Code Name	Description
0	OPEN_E_NONE	Update successful.
-22	OPEN_E_PARAM	Error in parameters passed.
-26	OPEN_E_FAIL	Update failed.

Request Example

```
curl -GET -XPUT -d "dstPort=0/5&session=1&mode=enabled"
10.10.10.1:8080/open/v1/swportmonitor

browser

http://10.10.10.1:8080/open/v1/swportmonitor?method=
put&dstPort=0/5&session=1&mode=enabled
```

Response Example

```
{
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Error Example

```
curl -GET -XPUT -d "session=20"
10.10.10.1:8080/open/v1/swportmonitor

browser

http://10.10.10.1:8080/open/v1/swportmonitor?method=
put&session=20
```

Response Example

```
{
  "status": {
    "response_code": 400,
    "msg": "Bad Request",
```

```
    "more_info": "Invalid argument"
    "error_code": -22
  }
}
```

SWPORTMONITOR DELETE

DELETE /open/v1/swportmonitor

Request Details

Table 4-118. SWPORTMONITOR DELETE Request Details

Parameter	Type	Description
session	integer	Session number. Required for source and destination port.
dstPort		Removes destination port.
srcPort	string	Source port.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-119. SWPORTMONITOR DELETE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-21	OPEN_E_INTERNAL	Request failed.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Request failed.
-28	OPEN_E_UNAVAIL	Parameter not supported.

Request Example

```
curl -GET -XDELETE 10.10.10.1:8080/open/v1/swportmonitor -d  
'dstPort=true&session=1'
```

browser

```
http://10.10.10.1:8080/open/v1/swportmonitor?method=  
delete&session=1&dstPort=true
```

Response Example

```
{  
  "status": {  
    "response_code": 200,  
    "msg": "OK",  
    "more_info": "",  
    "error_code": 0  
  }  
}
```

System1213

system1213 implements a RESTful API for the OpEN RFC1213 API. It focuses on the “system” object defined by that RFC. Note that a subset of the RFC1213 system object is defined in this version of the API.

System RFC1213 CREATE

This component does not support create semantics.

System RFC 1213 READ

```
GET /open/v1/system1213
```

Retrieves all of the variables that are associated with the system object in RFC1213.

Request Details

Table 4-120. System RFC 1213 READ Request Details

Parameter	Type	Description
sysDescr	string	System descriptor.
sysName	string	System name (optional).
sysLocation	string	System location (optional).
sysContact	string	System contact info (optional).

Response Details

HTTP Response Codes

The possible HTTP response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-121. System RFC 1213 READ Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.

Table 4-121. System RFC 1213 READ Internal Error Codes (Continued)

Error Code	Code Name	Description
-22	OPEN_E_PARAM	Invalid request.
-26	OPEN_E_FAIL	Operation failed.

Request Example

```
curl -GET 10.10.10.1:8080/open/v1/system1213
```

browser:

```
http://10.10.10.1:8080/open/v1/system1213
```

Response Example

```
{
  "sysDescr": "description",
  "sysName": "name",
  "sysLocation": "location",
  "sysContact": "contact"
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Ancillary Request Example

Ancillary request parameter to determine the maximum size of sysName string.

```
"curl -GET -d "fields=sysNameMaxSize"
10.10.10.1:8080/open/v1/system1213
```

browser

```
http://10.10.10.1:8080/open/v1/system1213?fields=
sysNameMaxSize
```

Response Example

```
{
  "sysNameMaxSize":128,
  "status":
    { "
      response_code":200,
      "msg":"OK",
      "more_info":"","
      "error_code":0
    }
}
```

Error Example

```
curl -GET -d "x=-99" 10.10.10.1:8080/open/v1/system1213
```

browser:

```
http://10.10.10.1:8080/open/v1/system1213?method=get&x=-99
```

Response Example

Returns an appropriate status message.

```
{
  "status": {
    "response_code": 400,
    "msg": "Bad Request",
    "more_info": "Invalid arguments",
    "error_code": -22
  }
}
```

System RFC 1213 UPDATE

```
PUT /open/v1/system1213
```

Use this call to update attributes of an RFC1213 system object. The following restrictions exist for strings:

- Length must be 255 characters or less.
- ASCII or UTF-8 printable characters only (depends on CLI configuration for extended ASCII codes).

Request Details

Table 4-122. System RFC 1213 UPDATE Request Details

Parameter	Type	Description
sysName	string	System name (optional).
sysLocation	string	System location (optional).
sysContact	string	System contact info (optional).

NOTE: At least one of the optional arguments must be specified, otherwise a 400 response and an OPEN_E_PARAM error code will be returned.

Response Details

HTTP Response Codes

The possible HTTP response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-123. System RFC 1213 UPDATE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Update successful.
-22	OPEN_E_PARAM	Error in parameters passed, or none of the optional parameters was specified.
-26	OPEN_E_FAIL	Update failed.

Request Example

```
curl -GET -XPUT -d "sysDescr=description&sysName=name"
10.10.10.1:8080/open/v1/system1213

browser:

http://10.10.10.1:8080/open/v1/system1213?method=put&sysDescr=
description&sysName=name
```

Response Example

```
{
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Error Example

```
curl -GET -XPUT -d "sysUndefined=2"
10.10.10.1:8080/open/v1/system1213
```

browser:

```
http://10.10.10.1:8080/open/v1/system1213?method=
put&sysUndefined=2
```

Response Example

```
{
  "status": {
    "response_code": 400,
    "msg": "Bad Request",
    "more_info": "Invalid arguments",
    "error_code": -22
  }
}
```

System RFC 1213 DELETE

This resource does not support delete semantics.

systemconfig

systemconfig implements a RESTful API for the OpEN API.

System Config CREATE

This resource does not support create semantics.

System Config READ

GET /open/v1/systemconfig

Retrieves the various attributes supported by this resource.

Request Details

Table 4-124. System Config READ Request Details

Parameter	Type	Description
sysAccessLine	string	[console telnet ssh] (optional parameter, required for sysLineTerminalLen)

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-125. System Config READ Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Request failed.
-28	OPEN_E_UNAVAIL	Parameter not supported.

Request Example

```
curl -GET 10.10.10.1:8080/open/v1/systemconfig  
browser  
http://10.10.10.1:8080/open/v1/systemconfig
```

Response Example

```
{  
  "sysSerialTimeOut" : 5,  
  "sysTelnetServerAdminMode" : "enabled",  
  "sysWarmRestartStatus" : "true",  
  "status":  
    {  
      "response_code": 200,  
      "msg": "OK",  
      "more_info": "",  
      "error_code": 0  
    }  
}
```

Request Example

```
curl -GET -d "sysAccessLine=console"  
10.10.10.1:8080/open/v1/systemconfig  
browser  
http://10.10.10.1:8080/open/v1/systemconfig?sysAccessLine=  
console
```

Response Example

```
{  
  "sysLineTerminalLen" : 128,  
  "status":  
    {  
      "response_code": 200,  
      "msg": "OK",  
      "more_info": "",  
      "error_code": 0  
    }  
}
```

Error Example

```
curl -GET -d "tenant=-99" 10.10.10.1:8080/open/v1/systemconfig
browser
http://10.10.10.1:8080/open/v1/systemconfig?method=get&tenant=-99
```

Response Example

Returns an appropriate status message.

```
{
  "status":
    {
      "response_code": 400,
      "msg": "Bad Request",
      "more_info":
        "Invalid parameter",
      "error_code": -22
    }
}
```

System Misc UPDATE

```
PUT /open/v1/systemconfig
```

Use this call to update various system configurations.

Request Details

Table 4-126. System Misc UPDATE Request Details

Parameter	Type	Description
sysAccessLine	string	console/telnet/ssh. (required for sysLineTerminalLen, sysLineTerminalDefaultLen)
sysLineTerminalDefaultLen	—	Set the terminal length of an access line to default. Do not use with sysLineTerminalLen parameter. (optional)
sysLineTerminalLen	integer	The terminal length of an access line. Do not use with sysLineTerminalDefaultLen parameter. (optional)

Table 4-126. System Misc UPDATE Request Details (Continued)

Parameter	Type	Description
sysSerialTimeOutDefault	—	Set the serial timeout to default value. Do not use with sysSerialTimeOut parameter. (optional)
sysSerialTimeOut	integer	The serial timeout. Do not use with sysSerialTimeOutDefault parameter. (optional)
sysTelnetServerAdminMode	string	The telnet server admin mode. (optional)
sysTransferBytesCompleted	integer	Bytes transferred for the file. (optional)
NOTE: At least one of the optional arguments must be specified, otherwise a 400 response and an OPEN_E_PARAM error code will be returned.		

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-127. System Misc UPDATE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Update successful.
-22	OPEN_E_PARAM	Error in parameters passed, or none of the optional parameters was specified.
-26	OPEN_E_FAIL	Update failed.

Request Example

```
curl -GET -XPUT -d "sysAccessLine=console&sysLineTerminalLen=
25" 10.10.10.1:8080/open/v1/systemconfig
browser
http://10.10.10.1:8080/open/v1/systemconfig?method=
put&sysAccessLine=console&sysLineTerminalLen=25
```

Response Example

```
{
  "status":
    {
      "response_code": 200,
      "msg": "OK",
      "more_info": "",
      "error_code": 0
    }
}
```

Error Example

```
curl -GET -XPUT 10.10.10.1:8080/open/v1/systemconfig
browser
http://10.10.10.1:8080/open/v1/systemconfig?method=put
```

Response Example

```
{
  "status":
    {
      "response_code": 400,
      "msg": "Bad Request",
      "more_info": "Invalid parameter"
      "error_code": -22
    }
}
```

System Misc DELETE

This resource does not support delete semantics.

systemcontrol

systemcontrol implements a RESTful API for the OpEN API.

System Control CREATE

```
POST /open/v1/systemcontrol
```

Use POST requests to support the system reload and configuration save semantics of this resource.

Request Details

Table 4-128. System Control CREATE Request Details

Parameter	Type	Description
reload	integer	1 = Reload (optional)
saveconfig	integer	1 = Save configuration

One or both of reload and saveconfig must be specified. If both are specified, then saveconfig will be applied prior to the reload. If saveconfig fails, then reload will not be performed.

The results of a reload are undefined. It is likely that the HTTP connection will be closed by the switch and timeout on the client side.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-129. System Control CREATE Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Requested operations performed.
-22	OPEN_E_PARAM	Error in parameters passed.

Table 4-129. System Control CREATE Internal Error Codes (Continued)

Error Code	Code Name	Description
-26	OPEN_E_FAIL	Requested operations failed or could not be performed.

Request Example

```
curl -POST -d "reload=1" 10.10.10.1:8080/open/v1/systemcontrol
browser:
http://10.10.10.1:8080/open/v1/systemcontrol?method=
post&reload=1
```

Response Example

Returns an appropriate status message.

```
{
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Error Example

```
curl -POST -d "reload=0" 10.10.10.1:8080/open/v1/systemcontrol
browser:
http://10.10.10.1:8080/open/v1/systemcontrol?method=
post&reload=0
```

Response Example

Returns an appropriate status message.

```
{
  "status": {
    "response_code": 400,
```

```
        "msg": "Bad Request",  
        "more_info": "Invalid parameters"  
        "error_code": -22  
    }  
}
```

System Control READ

Read semantics are not supported by this resource.

System Control UPDATE

Update semantics are not supported by this resource.

System Control DELETE

Delete semantics are not supported by this resource.

systemmisc

systemmisc implements a RESTful API for the OpEN API.

System Misc CREATE

This resource does not support create semantics.

System Misc READ

`GET /open/v1/systemmisc`

Retrieves the various attributes supported by this resource.

Request Details

Table 4-130. System Misc READ Request Details

Parameter	Type	Description
systemPrompt	string	System prompt (optional).
hostname	string	Hostname (optional).

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-131. System Misc READ Internal Error Codes

Error Code	Code Name	Description
0	OPEN_E_NONE	Success.
-22	OPEN_E_PARAM	Invalid parameter.
-26	OPEN_E_FAIL	Request failed.
-28	OPEN_E_UNAVAIL	Parameter not supported.

Request Example

```
curl -GET 10.10.10.1:8080/open/v1/systemmisc
```

browser:

```
http://10.10.10.1:8080/open/v1/systemmisc
```

Response Example

```
{
  "hostname" : "name",
  "hostnameSize" : 64,
  "systemPromptMaxSize" : 32,
  "systemUpTime" : 34563,
  "systemUptimeMsec" : 430,
  "systemBurnedInMacAddr" : "0a:34:12:6d:fe:12"
  "status": {
    "response_code": 200,
    "msg": "OK",
    "more_info": "",
    "error_code": 0
  }
}
```

Error Example

```
curl -GET -d "tenant=-99" 10.10.10.1:8080/open/v1/systemmisc
```

browser:

```
http://10.10.10.1:8080/open/v1/systemmisc?method=get&tenant=-99
```

Response Example

Returns an appropriate status message.

```
{
  "status": {
    "response_code": 400,
    "msg": "Bad Request",
    "more_info": "Invalid parameter",
  }
}
```

```
        "error_code": -22
    }
}
```

System Misc UPDATE

```
PUT /open/v1/systemmisc
```

Use this call to update various miscellaneous system settings. Use GET (see preceding example) to determine allowed lengths via the `hostnameSize` and `systemPromptMaxSize` attributes.

Request Details

Table 4-132. System Misc UPDATE Request Details

Parameter	Type	Description
systemPrompt	string	System prompt (optional).
hostname	string	Hostname (optional).

NOTE: At least one of the optional arguments must be specified, otherwise a 400 response and an `OPEN_E_PARAM` error code will be returned.

Response Details

HTTP Response Codes

The possible HTTP Response codes are 200, 400, and 500.

Internal Error Codes

The possible error codes for OpEN API are shown in the following table.

Table 4-133. System Misc UPDATE Internal Error Codes

Error Code	Code Name	Description
0	<code>OPEN_E_NONE</code>	Update successful.
-22	<code>OPEN_E_PARAM</code>	Error in parameters passed, or none of the optional parameters was specified.
-26	<code>OPEN_E_FAIL</code>	Update failed.

Request Example

```
curl -GET -XPUT -d "hostname=myswitch"  
10.10.10.1:8080/open/v1/systemmisc
```

browser:

```
http://10.10.10.1:8080/open/v1/systemmisc?method=put&hostname=  
myswitch
```

Response Example

```
{  
  "status": {  
    "response_code": 200,  
    "msg": "OK",  
    "more_info": "",  
    "error_code": 0  
  }  
}
```

Error Example

```
curl -GET -XPUT -d "vlanId=2"  
10.10.10.1:8080/open/v1/systemmisc
```

browser:

```
http://10.10.10.1:8080/open/v1/systemmisc?method=put&vlanId=2
```

Response Example

```
{  
  "status": {  
    "response_code": 400,  
    "msg": "Invalid parameter(s)",  
    "more_info": ""  
    "error_code": -22  
  }  
}
```

System Misc DELETE

This resource does not support delete semantics.



Printed in the U.S.A.

www.dell.com | support.dell.com