HPE Aruba Networking
REST API for AOS-S Switch 16.11

Hewlett Packard
Enterprise

Notices

The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Links to third-party websites take you outside the Hewlett Packard Enterprise website. Hewlett Packard Enterprise has no control over and is not responsible for information outside the Hewlett Packard Enterprise website.

Acknowledgments

Intel®, Itanium®, Pentium®, Xeon®, Intel Inside®, and the Intel Inside logo are trademarks of Intel Corporation in the U.S. and other countries.

Microsoft® and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Adobe® and Acrobat® are trademarks of Adobe Systems Incorporated.

Java® and Oracle® are registered trademarks of Oracle and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

Contents

Chapter 1
About This Guide

About This Guide

This guide provides information on how to use RestAPI and JSON schema.

Applicable Products

This guide applies to these products:

- Aruba 2530 Switch Series (J9772A, J9773A, J9774A, J9775A, J9776A, J9777A, J9778A, J9779A, J9780A, J9781A, J9782A, J9783A, J9853A, J9854A, J9855A, J9856A, JL070A)
- Aruba 2540 Switch Series (JL354A, JL355A, JL356A, JL357A)
- Aruba 2920 Switch Series (J9726A, J9727A, J9728A, J9729A, J9836A)
- Aruba 2930F Switch Series (JL253A, JL254A, JL255A, JL256A, JL258A, JL259A, JL260A, JL261A, JL262A, JL263A, JL264A, JL557A, JL558A, JL559A, JL692A)
- Aruba 2930M Switch Series (JL319A, JL320A, JL321A, JL322A, JL323A, JL324A, R0M67A, R0M68A)
- Aruba 3810 Switch Series (JL071A, JL072A, JL073A, JL074A, JL075A, JL076A)
- Aruba 5400R zl2 Switch Series (J9821A, J9822A, J9850A, J9851A, JL001A, JL002A, JL003A, JL095A)

Chapter 2
Rest API

Rest API

Overview

Representational State Transfer (REST) is a software architecture style consisting of guidelines and best practice for creating scalable web services. RESTful systems typically communicate over HTTP using the same verbs (that is, GET, POST, PUT, DELETE) used by web browsers.

The REST server listens for requests from clients on port 80 (HTTP) and port 443 (HTTPs). The REST service context is "/rest" followed by REST version. For example, http://192.168.1.1/rest/v1.0. REST requests also maybe sent on the HTTPS port (443).

- The REST interface is enabled by default.
- Web management or web management SSL must be enabled in the switch. If web management is disabled, requests will not be processed, irrespective of the REST interface status.
- IPv6 Rest API support is not available.
- REST requests may also be sent to a switch through an OOBM port.
- REST API requests that to stacked switches must use schema version v3 (except 3800 stack) or above.
- Do not use windows extractor to unzip schema files from 16.06 RestAPI.
- Default configuration of stack switches with build 16.04 and above does not display "no rest-interface" [except for 3800 stack].

More information

- JSON Schema for the most recent version of the software:
  - JSON Schema X.16.01.xxxx
  - JSON Schema X.16.02.xxxx
  - JSON Schema X.16.03.xxxx
  - JSON Schema X.16.04.xxxx
  - JSON Schema X.16.05.xxxx
  - JSON Schema X.16.06.xxxx
  - JSON Schema X.16.07.xxxx
  - JSON Schema X.16.08.xxxx
  - JSON Schema X.16.09.xxxx

Commands

rest-interface

Syntax

```
rest-interface

no rest-interface
```

Description

Enables the REST interface.

The `no` form of this command disables the REST interface.

Web management or Web management SSL must be enabled to process the REST requests. If Web management is disabled, the REST requests will not be processed even though the REST interface is enabled.

Command context

Required context: `config`

Examples

Enable the REST interface.

```
switch(config)# rest-interface
switch(config)# show rest-interface

 REST Interface - Server Configuration

  REST Interface            : Enabled
  REST Operational Status   : Up
  REST Session Idle Timeout : 600 seconds
  HTTP Access               : Enabled
  HTTPS Access              : Disabled
```

Disable the REST interface.

```
switch(config)# no rest-interface
switch(config)# show rest-interface

REST Interface - Server Configuration
  REST Interface            : Disabled
  REST Operational Status   : Down
  HTTP Access               : Enabled
  HTTPS Access              : Disabled
```

rest-interface session-idle-timeout

Syntax

`rest-interface session-idle-timeout` <SECONDS>

`no rest-interface session-idle-timeout`

Description

Configure session-idle-timeout for the REST interface sessions. The configurable value range is from 120 to 7200 seconds with the default value at 600 seconds.

The `no` version of the command resets the session-idle-timeout to the default value.

Command context

Required context: `config`

debug rest-interface

Syntax

`debug rest-interface`

Enables debug logs for the rest-interface.

debug rest-interface

```
HP-2920-48G(config)# debug rest-interface
0000:00:23:14.01 rest tHttpd:Received REST POST request
0000:00:23:14.08 rest tHttpd:(http_state:)http_process_post END
0000:00:23:14.15 rest tHttpd:REST request redirected to REST server
0000:00:23:14.22 rest tHttpd:Send REST request message to REST control task
0000:00:23:14.30 rest tHttpd:tHttpd is unblocked with necessary cleanup
0000:00:23:14.38 rest mrest_ctrl:Request for parse header
0000:00:23:14.44 rest mrest_ctrl:Request for parse header after method POST
```

Show Commands

show rest interface

Syntax

`show rest interface`

Description

The REST operational status will be shown as Up only when the REST interface is enabled and either Web management or Web management SSL is enabled. HTTP access is enabled when Web management is enabled. HTTPS access is enabled when Web management SSL is enabled.

Command context

Required context: `operator`

Example

```
switch# show rest-interface
REST Interface - Server Configuration
  REST Interface            : Enabled
  REST Operational Status   : Up
  REST Session Idle Timeout : 600 seconds
  HTTP Access               : Enabled
  HTTPS Access              : Disabled
```

AnyCLI RestAPIs

AnyCLI RestAPIs maintains contest level information.

Example: To create VLAN and enable IP routing follow following:

- Send an API call for Vlan creation.
- Send an API call for exit.
- Send an API call for enabling IP routing.

Restrictions

- Rest interface is not supported in FIPS.
- Rest Interface is not supported on ArubaOS 3800 stack switches.

Rest API Login, Create Vlan and Logout

Use Case Login Sessions

From AOS-S Switch version 16.10.0009 onwards, the factory default switches allow the REST APIs to configure username and password, and block all other APIs. When a REST session is initiated without a username or password, an HTTP error code 401 unauthorized is sent with the error message "`Configure the manager credentials and login session toaccess the switch.`".

If a user is configured on the switch, login request must be posted to the switch for authentication. If user validation is successful, the switch returns a session id as a cookie that will be used for further requests.

Users are able to create a maximum of five REST sessions simultaneously.

```
 WorkStation# curl --noproxy 10.100.167.104 -X POST
http://10.100.167.104:80/rest/v1/login-sessions -d '{"userName":"test",
"password":"test"}'
```

Response from the switch when user validation is successful:

```
{
"uri": "/rest/v1/login-sessions","cookie":
"sessionId=09CG1bRuT5hkCPzI97mmDjpn4uLtsmgkBsAaWUr9h7GxlkbsiASak1PEyj7Ov3n"
}
```

REST API Payload Enhancement

From AOS-S switch 16.10.0009 onwards, the default payload size has been increased as listed in the table below. REST clients are allowed to change the default buffer configuration with a value up to maximum payload size provided in the following table.

AOS-S Switch Model	Default Payload Size	Maximum Payload Size
2530 YA/YB	64K	64K
2540	64K	64K

AOS-S Switch Model	Default Payload Size	Maximum Payload Size
2920	16K	16K
2930 F/M	64K	1024K
3810M	64K	1024K
5400R	64K	1024K

Following operations reset payload buffer size to its default value:

- Disabling REST interface.
- Deletion or timeout of all REST sessions.

Example

The following example shows an authorized user configuring the payload size with a valid value:

```
WorkStation# curl --noproxy 10.100.167.104 -X POST
http://10.100.167.104:80/rest/v1/login-sessions -d
'{"payloadSize":1000000, "userName":"manager", "password":"*****"}'
```

Response from the switch when the payload size configuration is successful:

```
{
"payloadSize":1000000,"uri": "/rest/v1/login-sessions",
":"sessionId=3pZk5JKL2p4DrBrnHQcFNUHslDlv2depNk5cGpoISzFZYaYWT61GSRvsJegaHTJ"
}
```

The following example shows an authorized user configuring the payload size with an invalid value on platform 2930F:

```
WorkStation# curl --noproxy 10.100.167.104 -X POST
http://10.100.167.104:80/rest/v1/login-sessions -d '{"payloadSize":2000000,
"userName":"manager", "password":"*****"}'
```

Response from the switch when the payload size configuration is not successful:

```
{
"message": "payloadSize: Invalid input. Supported maximum value is '1048576'."
}
```

Use case — authentication failure

The switch will return an error if authentication fails:

```
WorkStation# curl --noproxy 10.100.167.104 -X POST
http://10.100.167.104:80/rest/v1/login-sessions -d '{"userName":"test",
"password":"test"}'
```

Switch response upon failure:

```
{
"message": "Authentication failed."
}
```

- An operator may only use the GET method. A manager is supported for all methods.
- Posting a login request is not required and requests may be sent without a cookie if a user name is not configured.

Other failure causes

1. The wrong username/password is sent to create login sessions.
2. The user is configured and REST requests are sent without session id.

Use case — creating a VLAN

Creating VLAN using REST:

```
WorkStation# curl --noproxy 10.100.167.104 --cookie
"sessionId=09CG1bRuT5hkCPzI97mmDjpn4uLtsmgkBsAaWUr9h7GxlkbsiASak1PEyj7Ov3n" -X
POST
http://10.100.167.104:80/rest/v1/vlans -d '{"vlan_id":5, "name":"VLAN5"}`
```

Response from the Switch:

```
{
"uri": "/rest/v1/vlans/5",
"vlan_id": 5,
"name": "VLAN5",
"status": "VS_PORT_BASED",
"type": "VT_STATIC",
"is_voice_enabled": false,
"is_jumbo_enabled": false,
"is_dsnoop_enabled": false
}
```

Use case — fetching the VLAN

Fetching the VLAN details using REST:

```
WorkStation# curl --noproxy 10.100.167.104 --cookie
"sessionId=09CG1bRuT5hkCPzI97mmDjpn4uLtsmgkBsAaWUr9h7GxlkbsiASak1PEyj7Ov3n" -X GET
http://10.100.167.104:80/rest/v1/vlans
```

Response from the Switch:

```
{
    "collection_result": {
       "total_elements_count": 2,
        "filtered_elements_count": 2
    },
    "vlan_element": [
        {
                "uri": "/rest/v1/vlans/1",
                "vlan_id": 1,
                "name": "DEFAULT_VLAN",
                "status": "VS_PORT_BASED",
                "type": "VT_STATIC",
                "is_voice_enabled": false,
                "is_jumbo_enabled": false,
                "is_dsnoop_enabled": false
       },
       {
                "uri": "/rest/v1/vlans/5",
                "vlan_id": 5,
                "name": "VLAN5",
                "status": "VS_PORT_BASED",
                "type": "VT_STATIC",
                "is_voice_enabled": false,
                "is_jumbo_enabled": false,
                "is_dsnoop_enabled": false
       }
    ]
}
```

Use case — logout of session

Logout of REST session.

```
WorkStation# curl --noproxy 10.100.167.104 --cookie
  "sessionId=09CG1bRuT5hkCPzI97mmDjpn4uLtsmgkBsAaWUr9h7GxlkbsiASak1PEyj7Ov3n" -X
DELETE http://10.100.167.104:80/rest/v1/login-sessions
```

The switch will respond with a message simular to: `204 No Content`.

```
[root@UbuntuServer5224 ~]# curl --noproxy 192.168.1.1 -v --cookie
"sessionId=8XU2msUzBJbXYClpFyVmaageB4iiRE6C94QSvYIfd3xiHelPjgUraYOx5JjWIUq" -X
DELETE http://192.168.1.1:80/rest/v1/login-sessions
* About to connect() to 192.168.1.1 port 80 (#0)
* Trying 192.168.1.1... connected
> DELETE /rest/v1/login-sessions HTTP/1.1
> User-Agent: curl/7.22.0 (x86_64-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1 zlib/
1.2.3.4 libidn/1.23 librtmp/2.3
> Host: 192.168.1.1
> Accept: */*
> Cookie:
sessionId=8XU2msUzBJbXYClpFyVmaageB4iiRE6C94QSvYIfd3xiHelPjgUraYOx5JjWIUq
>
< HTTP/1.1 204 No Content
< Server: eHTTP v2.0
< Connection: keep-alive
< RequestId:
```

```
<
* Connection #0 to host 192.168.1.1 left intact
* Closing connection #0
[root@UbuntuServer5224 ~]#
```

Use Case - Clearing Login Sessions

Clearing login sessions

There are two options available to clear login sessions from a switch:

- Enable or Disable the REST interface.
- Reboot the switch.

Clearing a specific session by ID requires the use of the DELETE method of Service Rest Login Sessions.

Use Case Clearing Login Sessions

Clearing login sessions.

```
WorkStation# curl --noproxy 192.168.1.1 -v -k --cookie
 "cookie":"<<Session id>>" -X DELETE  http://192.168.1.1:80/rest/v1/login-sessions
```

```
[root@UbuntuServer54185 ~]# curl --noproxy 192.168.1.1 -v -k --cookie

"cookie":"sessionId=wv6Nahl7D7dCw4egR5Nnu2d76qs93Y98g3CCwECmEAX8XkVNQGfV0sYDY4thzy
J"
                       -X DELETE  http://192.168.1.1:80/rest/v1/login-sessions
* About to connect() to 192.168.1.1 port 80 (#0)
*   Trying 192.168.1.1... connected
> DELETE /rest/v1/login-sessions HTTP/1.1
> User-Agent: curl/7.22.0 (x86_64-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1
zlib/1.2.3.4
                       libidn/1.23 librtmp/2.3
> Host: 192.168.1.1
> Accept: */*
> Cookie:
cookie:sessionId=wv6Nahl7D7dCw4egR5Nnu2d76qs93Y98g3CCwECmEAX8XkVNQGfV0sYDY4thzyJ
>
< HTTP/1.1 204 No Content
< Server: eHTTP v2.0
< Connection: keep-alive
< RequestId:
<
* Connection #0 to host 192.168.1.1 left intact
* Closing connection #0
[root@UbuntuServer54185 ~]#
[root@UbuntuServer54185 ~]#
```

To clear all REST API sessions in switch, use either one of the options.

- Enable/Disable rest-interface.
- Reboot the switch.

To clear a specific session id, use the DELETE method in the Service RestLoginSessions.

Responses and HTTP Status Codes Returned by RestAPI

GET Response and HTTP Status Codes Returned by the RestAPI

GET Response and HTTP status codes

Response Code	Description
200 OK	Request is successful.
400 Bad request	Request is invalid due to incorrect parameters in the request.
404 Not Found	Request has failed due to non-existent resource.
500 Internal Server Error	Request has failed due to an error in the switch.

POST Responses and HTTP Status Codes Returned by RestAPI

POST responses and HTTP status codes

Response Code	Description
200 OK	Command URI is successful.
201 Created	Command URI is created.
202 Accepted	Command URI is accepted.
204 No Content	No content is returned for command URI.
400 Bad Request	Request has failed due to incorrect parameters in the request.
404 Not Found	Request is failed due to non-existent resource.
405 Method Not Allowed	Request not supported on a resource.
500 Internal Server Error	Request is failed due to an error in the switch.

PUT responses and HTTP status codes returned by RestAPI

PUT responses and HTTP status codes

Response Code	Description
200 OK	Resource is successful.
400 Bad Request	Failed due to incorrect parameters in the request.
404 Not Found	Failed due to non-existent resource.
405 Method Not Allowed	Not supported on a resource.
500 Internal Server Error	Failed due to an error in the switch.

DELETE Responses and HTTP Status Codes Returned by RestAPI

DELETE responses and HTTP status codes

Response code	Description
204 No Content	Delete on a resource is successful.
400 Bad Request	Invalid DELETE request [some incorrect parameters in the request].
404 Not Found	Request has failed due to non-existent resource.
405 Method Not Allowed	Not supported on a resource.
500 Internal Server Error	Request is failed due to an error in the switch.

Chapter 3

Authentication, Authorization, and

Authentication, Authorization, and Accounting for REST

Starting with AOS-S switch 16.08, authentication, authorization support is provided for Local, Remote Authentication Dial-In User Service (RADIUS) and Terminal Access Controller Access Control System+ (TACACS+) for the REST interface. Accounting for REST interface is supported on RADIUS and TACACS+. REST users can choose either per-switch passwords or authentication based on RADIUS or TACACS+.

The REST interface behavior has changed, and customers upgrading to 16.08 must be aware of the changes. For details, see Migrating from pre-16.08 code base to 16.08.

From the switch software version 16.10.0015 onwards, Authentication, Authorization, and Accounting (AAA) for REST is enabled for 2530 switches.

Definition of Terms

URI	A URI (Uniform Resource Identifier) is a sequence of characters that identifies a logical or physical resource.
JSON	JSON, or JavaScript Object Notation, is a minimal, readable format for structuring data. It is used primarily to transmit data between a server and web application, as an alternative to XML.

Setting up AAA for REST

Enable the following command to get manager privilege for RADIUS and TACACS authorization.

```
aaa authentication login privilege-mode
```

Syntax

```
aaa authentication login privilege-mode
```

Usage

```
[no] aaa authentication login privilege-mode
```

Description

Specify that switch respects the authentication server's privilege level.

The following sections list the procedure to setup AAA for REST on Local, RADIUS, and TACACS+. For command details, see CLI Commands on page 27

RADIUS

The RADIUS server must be configured and the configuration file must be available on the RADIUS server.

- Authentication - Configure Operator and Manager with primary authentication method as RADIUS and backup method as Local. The commands are:

```
(config)# aaa authentication rest login radius
(config)# aaa authentication rest enable radius
```

- Authorization - Configure the HP-URI-Exception, HP-URI-Json-String, HP-URI-Access in the RADIUS configuration file. Use the following command to configure URI authorization on the switch:

```
(config)# aaa authorization rest-uri radius
```

- Accounting - Enable the URI, exec and system accounting on RADIUS for REST interface using the `aaa accounting` command. For example,

```
(config)# aaa accounting exec start-stop radius
(config)# aaa accounting system start-stop radius
(config)# aaa accounting rest-uri stop-only radius
```

TACACS+

- **Authentication**
  Configure Operator and Manager with primary authentication method as TACACS and backup method as Local. The commands are:

```
(config)# aaa authentication rest login tacacs
(config)# aaa authentication rest enable tacacs
```

- **Authorization**
  - Configure the rules for authorization in the TACACS configuration file.
  - Enable TACACS authorization using the following command:

```
(config)# aaa authorization rest-uri tacacs
```

- **Accounting**
  Enable the URI, exec and system accounting on TACACS+ server for REST interface using the `aaa accounting` command with appropriate options. For example,

```
(config)# aaa accounting exec start-stop tacacs
(config)# aaa accounting system start-stop tacacs
(config)# aaa accounting rest-uri stop-only tacacs
```

Local

- **Authentication**
  Configure Operator and Manager with primary authentication method as Local. The commands are:

```
(config)# aaa authentication rest login local
(config)# aaa authentication rest enable local
```

- **Authorization**
  The following is an example illustrates the configuration to authorize admin1 to execute authentication GET URIs:
  - Enable Local authorization:

    ```
    (config)# aaa authorization rest-uri local
    ```

  - Create a group, group1 with the following parameters and command:

    ```
    URI:            v6/aaa
    Json attribute: authentication
    URI Access:     GET
    Exception:      permit

    (config)# aaa authorization group group1 uri-seq 23 match-uri v6/aaa
    match-json authentication uri-access get permit
    ```

  - Create Local user and associate the user with the group.

    ```
    (config)# aaa authentication local-user admin1 group group1 password
    plaintext
    New password for user1: *********
    Please retype new password for user1: *********
    ```

  With the execution of the above two commands, admin1 is part of group1, which has authorization set to execute authentication GET URIs.

Authentication

The REST interface includes configuration commands for login (operator) and enable (manager) along with primary and secondary authentication methods. For configuration details, see CLI Commands on page 27.

Authorization

URI authorization is supported through Local, RADIUS and TACACS+. URI authorization is performed based on the following attributes:

- URI access method (PUT, POST, DELETE, GET).
- URI (URI string along with the JSON attributes).

Considerations

- If the URI authorization is not configured for local, RADIUS and TACACS users but URI authorization is enabled, then access to any URI execution is denied. This is the same behavior as command authorization.

- The URI-Access, URI-String and URI-JSON attributes are mandatory. The default value for the URI Exception is permit. Only one attribute is allowed for URI-Access and URI-String .The URI-JSON can have single or multiple attributes with comma as separator.
- If the authentication and authorization methods are different for a user, the authorization URI does not have any impact. This is the same behavior as command authorization.
- The `cli_batch` URI has encoded attributes that are denied or allowed in the authorized list of URIs. The encoded data is not authorized. Aruba recommends to permit this only on need basis.
- URI authorization is disabled by default like command authorization and must be enabled through `aaa authorization` command.
- With permit option, all the JSON attributes must match for successful URI execution.
- With deny option, at least one JSON attribute match will result in the denying of URI execution.
- The match is performed only on the JSON attributes and not on the JSON values.

Local authorization

A group for URI authorization must have the following parameters configured:

- URI
- JSON attribute
- URI access method
- Permit or deny

A group can be created to have the above configuration for URI authorization for REST interface, or the existing local groups can be extended to configure the REST URIs, JSON strings, and URI access methods. A group can have both command and URI authorizations configured. For the user associated with such a group, URI authorization is applicable for REST interface and command authorization for other supported management interfaces.

The following table illustrates an example for local authorization:

```
Example URI Executed:
curl -X PUT http://10.100.106.244/rest/v2/vlans
{"collection_result":{"total_elements_count":1,"filtered_elements_count":1},
"vlan_element":[{"uri":"/vlans/1","vlan_id":1,"name":"DEFAULT_VLAN","status":
"VS_PORT_BASED","type":"VT_STATIC","is_voice_enabled":false,"is_jumbo_
enabled":false,
"is_dsnoop_enabled":false,"is_dhcp_server_enabled":false}]}
```

The match-uri, match-json and uri-access in the following table are the parameters of `aaa authorization group` command:

match-uri	match-json	uri-access	Exception	Behavior
.*	.*	.*	permit	All URIs will be allowed to execute.The above URI will be successful.

match-uri	match-json	uri-access	Exception	Behavior
v2	.*	.*	permit	All URI options with v2 will be allowed. The URIs with other versions will not be allowed. The above URI execution will be successful.
v2/vlans	is_jumbo_enabled	PUT	permit	The URI with the Json parameter is_jumbo_ enabled will be executed for PUT method. The above URI execution will not be successful since all JSON parameters in the URI being executed do not match.
.*	is_jumbo_enabled	PUT	permit	Error is returned during configuration.
v2/vlans	.*	.*	permit	The execution of all URIs that belongs to v2/vlans will be allowed for all access methods. The execution of the above URI will be successful.
v2	is_jumbo_enabled	.*	permit	Any URIs that belong to v2 having the JSON attribute as is_ jumbo_enabled will be allowed for execution. If there are any other JSON attributes, they will have the response as FAIL.

match-uri	match-json	uri-access	Exception	Behavior
.*	is_jumbo_enabled	.*	permit	Error is returned during configuration.
.*	.*	GET	permit	All URIs with GET access method will be allowed for execution. The execution of the above URI will fail.
v2/vlans	.*	PUT	permit	All URI options with v2/vlans will be allowed for PUT. The execution of the above URI will be successful.
.*	.*	.*	deny	All URIs will be rejected.
v2	.*	.*	deny	All URI options with v2 (version v2) will not be allowed for execution. The URIs with other versions will be allowed.
v2/vlans	is_jumbo_enabled	PUT	deny	The URI with v2/vlans and JSON attribute is_jumbo_ enabled and access_method as PUT will be denied and any other URI execution will be permitted.
.*	is_jumbo_enabled	PUT	deny	Error is returned during configuration.
v2/vlans	.*	.*	deny	The execution of URIs that belong to v2/vlans will not be allowed for all access methods.

match-uri	match-json	uri-access	Exception	Behavior
v2	is_jumbo_enabled	.*	deny	Any URIs that belong to v2 having the JSON parameter is_jumbo_enabled will not be allowed.
.*	is_jumbo_enabled	.*	deny	Error is returned during configuration.
.*	.*	GET	deny	All GET URIs will not be allowed. Any URIs with other URI access methods will be successful.
v2/vlans	.*	PUT	deny	All URI options with v2/vlans will not be allowed for PUT. Any other URIs or with any other URI access method will be allowed.

RADIUS

Some RADIUS servers maintain the order of attributes in the RADIUS packet as in the configuration, and some don't. The constraints for configuring the authorization Vendor-Specific Attribute (VSA) are different for each type of RADIUS server.

RADIUS servers are categorized into two types based on their ability to maintain the order of attributes in the RADIUS packets and the different configuration constraints for each type of RADIUS server.

RADIUS Servers that Maintain the Order of Attributes in the RADIUS packets

These RADIUS servers maintain the order of attributes in the RADIUS packets as it is in the configuration.

The following RADIUS VSAs are provided to configure the URI, JSON, and Access methods for URI authorization.

Attribute	Value	String/Integer
HP-URI-String (mandatory)	80	String
HP-URI-Json-String (mandatory)	81	String

Attribute	Value	String/Integer
HP-URI-Access (mandatory)	82	String
HP-URI-Exception	83	Integer

- The value for HP-URI-Exception is 0 (permit) and 1 (deny). By default, HP-URI-Exception is permit and is optional.
- The value for HP-URI-Access must be "GET", "POST", "PUT", "DELETE" and ".*"only (".*" refers to all access methods).
- If HP-URI-String is configured with ".*", HP-URI-Json-String must be configured with ".*" only.
- The attributes HP-URI-String, HP-URI-Access, and HP-URI-Exception must be configured with single value and the HP-URI-Json-String can be configured with single or multiple values with comma separator.
- If any of the mandatory VSAs are not configured or if the HP-URI-Json-String is not the last attribute, authentication will fail.
- Both command and REST URI authorization parameters can be configured for a user in RADIUS configuration file.

A sample RADIUS user file:

```
user Cleartext-Password:="user123"
Service-Type = Administrative-User, reply-Message = "Hello",
HP-URI-String = ".*",
HP-URI-Access = "PUT",
HP-URI-Exception = 1,
HP-URI-Json-String = ".*",
HP-URI-String += ".*",
HP-URI-Access += "GET",
HP-URI-Json-String += ".*"
```

RADIUS Servers that Do Not Maintain Order of Attributes in the RADIUS packets

These RADIUS servers do not maintain the order of attributes in the RADIUS packets as it is in the configuration.

The following RADIUS VSAs are provided to configure the URI, JSON, and Access methods for URI authorization.

Attribute	Value	String/Integer
HP-URI-String (mandatory)	80	String
HP-URI-Json-String (mandatory)	81	String
HP-URI-Access (mandatory)	82	String

Attribute	Value	String/Integer
HP-URI-Exception (mandatory)	83	Integer

- All the attributes, HP-URI-String, HP-URI-Access, HP-URI-Exception, and HP-URI-Json-String are mandatory and should be configured exactly once for each URI to be authorized.
- The value for HP-URI-Access must be "GET", "POST", "PUT", "DELETE" and ".*"only (".*" refers to all access methods).
- If HP-URI-String is configured with ".*", HP-URI-Json-String must be configured with ".*" only.
- The attributes HP-URI-String, HP-URI-Access, and HP-URI-Exception must be configured with single value and the HP-URI-Json-String can be configured with single or multiple values with comma separator.
- If any of the mandatory VSAs are not configured, authentication will fail.

A sample RADIUS user file:

```
user Cleartext-Password:="user123"
Service-Type = Administrative-User, reply-Message = "Hello",
HP-URI-String = ".*",
HP-URI-Access = "PUT",
HP-URI-Exception = 1,
HP-URI-Json-String = ".*",
HP-URI-String += ".*",
HP-URI-Access += "GET",
HP-URI-Json-String += ".*"
HP-URI-Exception += 0,
```

TACACS+

The TACACS+ authorization is performed by configuring the rules for the URI authorization in the configuration file on the TACACS+ server. A sample is shown below:

```
group = admin {
default service = permit service = shell {
priv-lvl = 15
}
deny "/rest/v3/vlans vlan_id name POST" permit .*

}
user = user1 { member = admin
login = cleartext testing service = exec {
priv-lvl = 15
}
```

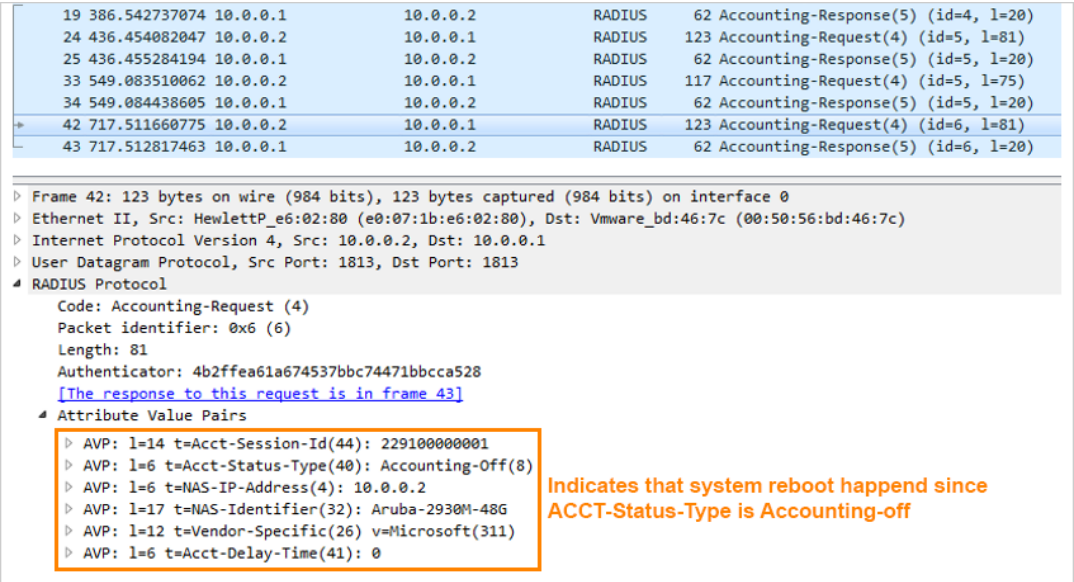
All the attributes must match to permit or deny.

Accounting

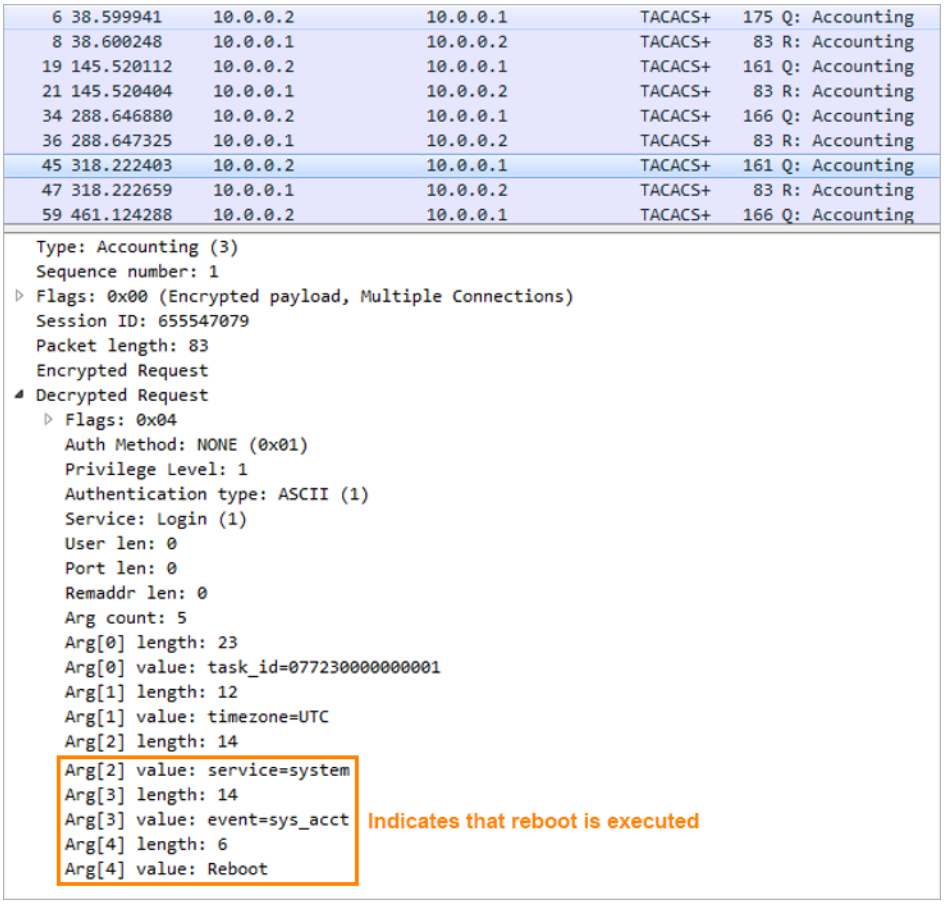
Accounting on the switch collects configured data and forwards it to the RADIUS or TACACS+ server. The accounting through REST interface supports the following:

- SYSTEM accounting (boot, reboot, reload) provides records containing the information about the system events that occur on the switch, which includes system reset/system boot through the rest session. The following are example packets of system accounting:

  RADIUS

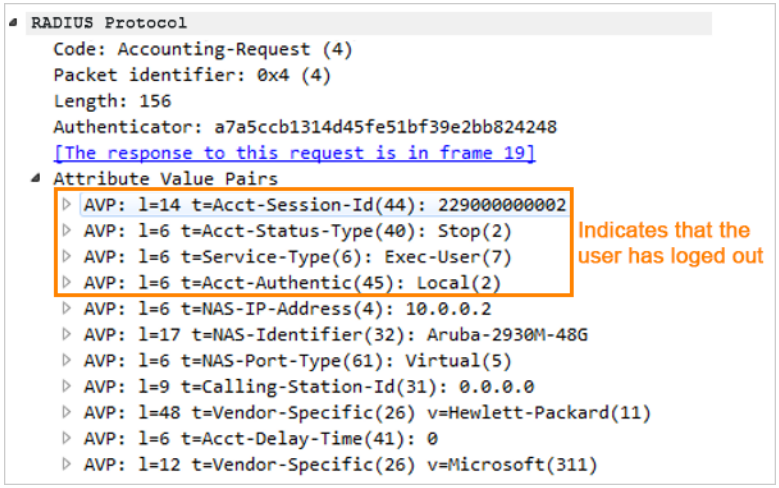


  TACACS+



- EXEC accounting (login, logout) provides records holding the information about REST login session. A maximum of 5 REST sessions are supported on the switch. Executive accounting will collect any attempt at session login and will account the event.

  RADIUS



  TACACS+

- URI accounting (URI execution) provides records containing information on uri request and the Json payload. Uri request information has the information about the rest-uri, Json payload and the http method details (POST/PUT/GET/DELETE). URI accounting happens for successfully executed URIs. The size of the Json payload in the accounting packet depends on the size limitation of the packet size permitted on the accounting server.

  RADIUS

```
    MS-RAS-Vendor = 11
(2) Received Accounting-Request packet from host 10.0.0.2 port 1813, id=6, length=247
(2)   Acct-Session-Id = '000500000003'
(2)   Acct-Status-Type = Stop
(2)   Service-Type = NAS-Prompt-User
(2)   NAS-IP-Address = 10.100.1.250
(2)   NAS-Identifier = 'Aruba-2930M-48G-PoEP'
(2)   NAS-Port-Type = Virtual
(2)   Calling-Station-Id = '10.0.10.193'
(2)   HP-Uri-String = '/rest/v5/accounting'
(2)   HP-Uri-Access = 'POST'
(2)   HP-Uri-Json-String =
'{\"accounting_rest\":{\"accounting_method\":\"AME_TACACS\",\"accounting_mode\":\"AMO_STOP_O
NLY\"}}'
(2)   Acct-Delay-Time = 0
(2)   MS-RAS-Vendor = 11
```

TACACS+

```
Flags: 0x04
Auth Method: NONE (0x01)
Privilege Level: 1
Authentication type: ASCII (1)
Service: Login (1)
User len: 0
Port len: 0
Remaddr len: 11
Remote Address: 10.0.10.193
Arg count: 6
Arg[0] length: 23
Arg[0] value: task_id=082980000000008
Arg[1] length: 12
Arg[1] value: service=REST
Arg[2] length: 11
Arg[2] value: stop_time=0
Arg[3] length: 11
Arg[3] value: method_post
Arg[4] length: 37
Arg[4] value: rest_uri=/rest/v5/authorization_group
Arg[5] length: 122
Arg[5] value: jason_payload={  "group_name": "gt","seq_num": 5,"cmd_permission":"AZP_PERMIT","match_cmd": "conf","is_log_enabled"
```

Limitations:

- The maximum accounting packet size is 1.5K for TACACS+ and 4K for RADIUS.
- REST accounting using external log server like Syslog for REST is not supported.

CLI Commands

Authentication command

`aaa authentication rest`

Syntax

`aaa authentication rest login {local | radius | tacacs} [local| tacacs | none]`

Usage:

`aaa authentication rest {enable | login} <primary-method> [<backup-method>]`

Description

Configure authentication mechanism used to control REST access to a switch.

Authorization commands

`aaa authorization rest-uri`

Syntax

```
aaa authorization rest-uri [local | radius | none | tacacs]
```

Usage:

```
[no] aaa authorization rest-uri {radius | local | tacacs | auto | none}
 [no] aaa authorization rest-uri access-level {manager | all}
```

Description

Configure rest-uri authorization. For each rest-uri issued by the user, an authorization request is sent to the server. rest-uri authorization can be applied to all rest-uris or only manager-level rest-uris.

```
 aaa authorization group
```

Syntax

```
aaa authorization group <ASCII-STR> uri-seq <SEQ-RANGE> match-uri <URI-STR>
match-json <JSON-STR> uri-access
{[put | post | get | delete | all] [permit | deny] log}
```

Usage:

```
[no] aaa authorization group <GROUPNAME> <SEQ-NUM>
              match-command <COMMAND> {deny | permit} [log]
[no] aaa authorization group <GROUPNAME> uri-seq <SEQ-NUM>
              match-uri <URI-STR> match-json <JSON-STR> uri-access <ACCESS>
              {deny | permit} [log]
```

Description

Create or remove an authorization rule.

```
 aaa authentication local-user
```

Syntax

```
aaa authentication local-user <USERNAME> {{ group <GROUPNAME>
 password {plaintext|sha1|sha256 <PASSWORD>}}
 | {aging-period <aging-time>} |
 {min-pwd-length <length>} |
 {clear-password-history}
```

Parameters

local-user <USERNAME>

The local user being added to the authorization group. The username can be up to 16 characters. The username must not contain spaces and is case-sensitive.

group <GROUPNAME>

Name of the authorization group to which the local user belongs. The group must be an existing group.

password {plaintext|sha1|sha256 <PASSWORD>}

The password can have a maximum of 16 characters. It must not contain spaces and is case-sensitive. The default is plaintext.

aging-period <aging-time>

The password aging time.

min-pwd-length <length>

The password minimum length.

clear-password-history

Clear the password history for a user.

Usage:

```
[no] aaa authentication local-user <USERNAME> {{ group <GROUPNAME>
 password {plaintext|sha1|sha256 <PASSWORD>}}
 | {aging-period <aging-time>} |
 {min-pwd-length <length>} |
 {clear-password-history}}
```

Description

Create or remove a local user account.

Accounting commands

```
 aaa accounting rest-uri
```

Syntax

```
aaa accounting rest-uri {[stop-only | interim-update] [radius | tacacs]} server-group
<ASCII-STR>
```

Usage

```
[no] aaa accounting {exec | network | system | commands | rest-uri}
         {start-stop | stop-only | intermim-update}
         {radius | syslog | tacacs}
[no] aaa accounting update periodic <Minutes>
[no] aaa accounting suppress null-username
 aaa accounting session-id {unique | common}
```

Description

Configure the accounting service on the device. Accounting can be configured for EXEC sessions, network connection, commands, rest-uri and system. The accounting data is collected by a RADIUS, SYSLOG, or TACACS+ server. Network accounting is not supported through TACACS+ and SYSLOG. session-id accounting is not supported for TACACS+.

Upgrading from 16.08 Onwards

The following are the behavioral changes to REST interface from 16.08 onwards:

- If the credentials are configured, the execution of `GET Version` URI requires authentication. The earlier releases allow this execution without authentication.
- The Operator cannot execute all the GET URIs. The following table lists the GET URIs restricted for Operator along with the corresponding show commands:

GET URI	Command
GET /ta_profiles	show crypto pki ta-profile

GET URI	Command
GET/ta_profiles/{TaProfile.ta_name}	show crypto pki ta-profile <ta-profile-name>
GET /vlans-ports, GET /vlans	show running-config vlan
GET /vlans-ports/{VlanPort.vlan_id}-{VlanPort.port_id}, GET /vlans/{Vlan.vlan_id}	show running-config vlan <vlan>
GET /intrusion_logs	show port-security intrusion-log
GET /portsec_policies	show port-security
GET /portsec_policies/ {PortSecurityPolicy.port_id}	show port-security <port-list>
GET /radius_servers, GET /radius_profile	show radius
GET /snmp-server/traps	show snmp-server traps
GET /snmp-server/communities/{SnmpServerCommunity.community}	show snmp-server <community>

Chapter 4
Change Log

Change Log

New APIs

Service ArpProtect

HTTP Method	Resource	Request	Response	Description
GET	.../arp-protect	N/A	ArpProtect	Returns the attributes of ARP protection.
PUT	.../arp-protect	ArpProtect	ArpProtect	Updates attributes of ARP protection.
GET	.../arp-protect/ports	N/A	WiredElementList	Returns ARP protect configuration for all ports.
GET	.../arp-protect/ ports/ {ArpProtectPort.por t_id}	N/A	ArpProtectPort	Returns ARP protect per port details.
PUT	.../arp-protect/ ports/ {ArpProtectPort.por t_id}	ArpProtectPort	ArpProtectPort	Updates attributes of ARP Protect per port.
GET	.../arp-protect/vlans	N/A	WiredElementList	Retrieves ARP protect VLANs.
GET	.../arp-protect/vlans/ {ArpProtectVlan.vlan_id}	N/A	ArpProtectVlan	Retrieves ARP protect VLANs by vlan_id.
PUT	.../arp-protect/vlans/ {ArpProtectVlan.vlan_id}	ArpProtectVlan	ArpProtectVlan	Updates ARP protect VLANs by vlan_id.

Service Jumbo

HTTP Method	Resource	Request	Response	Description
GET	.../jumbo	N/A	Jumbo	Returns Jumbo configuration.
PUT	.../jumbo	Jumbo	Jumbo	Updates attributes of Jumbo.

Service MacLockout

HTTP Method	Resource	Request	Response	Description
GET	.../lockout_mac	N/A	WiredElementList	Returns all the MAC.
POST	.../lockout_mac	LockoutMac	LockoutMac	Adds the given mac address to lockout list for blacklisting.
DELETE	.../lockout_mac/{LockoutMac.mac_address}	N/A	N/A	Deletes the mac address from the lockout list.

Service PrivateVlan

HTTP Method	Resource	Request	Response	Description
GET	.../private-vlans	N/A	CommonElementList	Returns all the primary or isolated or community details.
GET	.../private-vlans/isl-ports	N/A	CommonElementList	Return list of isl ports associated with the primary VID's.
GET	.../private-vlans/promiscuous-ports	N/A	CommonElementList	Return list of promiscuous ports associated with the primary VID's.

Updated APIs

Service DhcpRelay

HTTP Method	Resource	Request	Response	Description
GET	.../dhcp-relay/option_82	N/A	DhcpRelayOption82	Returns the DHCP Relay Option 82.
PUT	.../dhcp-relay/option_82	DhcpRelayOption82	DhcpRelayOption82	Updates the given DHCP Relay Option 82.

Attribute 'is_option82_enabled' in DhcpRelayOption82.json is updated with the default_value, false.

Service Dsnoop

HTTP Method	Resource	Request	Response	Description
GET	.../dsnoop	N/A	Dsnoop	Returns the DHCP-Snooping global attributes.
PUT	.../dsnoop	Dsnoop	Dsnoop	Update the DHCP-Snooping global attributes.

Attribute 'is_dhcp_snooping_enabled' in Dsnoop.json is updated with default_value as 'false'.

Service Igmp

HTTP Method	Resource	Request	Response	Description
GET	.../vlans/{VlanIgmp.vlan_id}/igmp	N/A	VlanIgmp	Returns the VLAN IGMP matching the given ID.
PUT	.../vlans/{VlanIgmp.vlan_id}/igmp	VlanIgmp	VlanIgmp	Updates the given VLAN IGMP in the model.
GET	.../igmp	N/A	Igmp	Returns IGMP global configuration.
PUT	.../igmp	Igmp	Igmp	Updates the IGMP global configuration.

- Attribute 'is_igmp_enabled' in VlanIgmp.json is updated with default_value as 'false'.
- Attribute 'filter_unknown_mcast' in Igmp.json is updated with default_value as 'false'.

Service IpRoute

HTTP Method	Resource	Request	Response	Description
GET	.../ip-route	N/A	CommonElementList	Returns the list of IP Route elements matching the given query.
PUT	.../ip-route	IpRoute	IpRoute	Adds the given IP Route in the model.
GET	.../ip-route/{IpRoute.id}	N/A	IpRoute	Returns the IP Route matching the given ID.
PUT	.../ip-route/{IpRoute.id}	IpRoute	IpRoute	Updates the given IP Route matching the given ID.
DELETE	.../ip-route/{IpRoute.id}	N/A	N/A	Updates the given IP Route matching the given ID.

Attribute 'ip_route_mode' is updated with a new value 'IRM_LOOPBACK'.

Service Port

HTTP Method	Resource	Request	Response	Description
GET	.../port-statistics	N/A	CommonElementList	Returns All Ports statistics.
GET	.../port-statistics/{PortStatistics.id}	N/A	PortStatistics	Returns the Port statistics matching the given ID.
GET	.../ports	N/A	CommonElementList	Returns the list of Port elements matching the given query.
GET	.../ports/{Port.id}	N/A	Port	Returns the Port matching the given ID.
PUT	.../ports/{Port.id}	Port	Port	Updates the given Port in the model.

- Attribute '`string name`' in `Port.json` is updated with `min_length` as '0' along with description.
- Attribute '`is_flow_control_enabled`' in `Port.json` is updated with `default_value` as 'false'.
- Attribute '`is_dsnoop_port_trusted`' in `Port.json` is updated with `default_value` as 'false'.

Service Rpvst

HTTP Method	Resource	Request	Response	Description
`GET`	.../rpvst/vlans	N/A	`WiredElementList`	Returns RPVST VLAN configuration for all ports.
`GET`	.../rpvst/vlans/{RapidPvstVlan.vlan_id}	N/A	`RapidPvstVlan`	Returns RPVST per VLAN configuration.
`PUT`	.../rpvst/vlans/{RapidPvstVlan.vlan_id}	`RapidPvstVlan`	`RapidPvstVlan`	Updates attributes of RPVST per VLAN.

Service System

HTTP Method	Resource	Request	Response	Description
`GET`	.../system	N/A	`SystemAttributes`	Returns the host system attributes.
`PUT`	.../system	`SystemAttributes`	`SystemAttributes`	Updates the host system attributes.

A new attribute `rollback_to_good_config` is added in the `SystemAttributes.json`.

Service TACACSProfile

HTTP Method	Resource	Request	Response	Description
`GET`	.../tacacs_profile	N/A	`TacacsProfile`	Returns TACACS Profile.
`PUT`	.../tacacs_profile	`TacacsProfile`	`TacacsProfile`	Updates attributes of TACACS Profile.

Attribute '`ordering_sequence`' in newly added in `TacacsProfile.json`.

Service Vlan

HTTP Method	Resource	Request	Response	Description
GET	.../vlans	N/A	CommonElementList	Returns the list of VLAN elements matching the given query.
POST	.../vlans	Vlan	Vlan]	Adds the given VLAN to the model.
GET	.../vlans/{Vlan.vlan_id}	N/A	Vlan]	Returns the VLAN matching the given ID.
PUT	.../vlans/{Vlan.vlan_id}	Vlan	Vlan]	Updates the given VLAN in the model.
DELETE	.../vlans/{Vlan.vlan_id}	N/A	N/A	Deletes the VLAN with the given ID from the model.

- Attribute 'is_jumbo_enabled' in Vlan.json is updated with default_value as 'false'.
- Attribute 'is_voice_enabled' in Vlan.json is updated with default_value as 'false'.
- Attribute 'is_dsnoop_enabled' in Vlan.json is updated with default_value as 'false'.
- Attribute 'is_dhcp_server_enabled' in Vlan.json is updated with default_value as 'false'.
- Attribute 'is_management_vlan' in Vlan.json is updated with default_value as 'false'.

Accessing Aruba Support

Aruba Support Services	https://www.arubanetworks.com/support-services/
Aruba Support Portal	https://asp.arubanetworks.com/
North America telephone	1-800-943-4526 (US & Canada Toll-Free Number) +1-408-754-1200 (Primary - Toll Number) +1-650-385-6582 (Backup - Toll Number - Use only when all other numbers are not working)
International telephone	https://www.arubanetworks.com/support-services/contactsupport/

Be sure to collect the following information before contacting Support:

- Technical support registration number (if applicable)
- Product name, model or version, and serial number
- Operating system name and version
- Firmware version
- Error messages
- Product-specific reports and logs
- Add-on products or components
- Third-party products or components

Other useful Sites

Other websites that can be used to find information:

Airheads social forums and Knowledge Base	https://community.arubanetworks.com/
Software licensing	https://lms.arubanetworks.com/
End-of-Life information	https://www.arubanetworks.com/support-services/end-of-life/
Aruba software and documentation	https://asp.arubanetworks.com/downloads

Accessing Updates

To download product updates:

Aruba Support Portal

https://asp.arubanetworks.com/downloads

If you are unable to find your product in the Aruba Support Portal, you may need to search My Networking, where older networking products can be found:

My Networking

https://www.hpe.com/networking/support

To view and update your entitlements, and to link your contracts and warranties with your profile, go to the Hewlett Packard Enterprise Support Center More Information on Access to Support Materials page:

https://support.hpe.com/portal/site/hpsc/aae/home/

Access to some updates might require product entitlement when accessed through the Hewlett Packard Enterprise Support Center. You must have an HPE Passport set up with relevant entitlements.

Some software products provide a mechanism for accessing software updates through the product interface. Review your product documentation to identify the recommended software update method.

To subscribe to eNewsletters and alerts:

https://www.hpe.com/support/e-updates

Warranty Information

To view warranty information for your product, go to https://www.arubanetworks.com/supportservices/product-warranties/.

Regulatory Information

To view the regulatory information for your product, view the *Safety and Compliance Information for Server, Storage, Power, Networking, and Rack Products*, available at www.hpe.com/support/Safety-Compliance-EnterpriseProducts

Additional regulatory information

Aruba is committed to providing our customers with information about the chemical substances in our products as needed to comply with legal requirements, environmental data (company programs, product recycling, energy efficiency), and safety information and compliance data, (RoHS and WEEE). For more information, see https://www.arubanetworks.com/company/about-us/environmental-citizenship/.

Documentation Feedback

Aruba is committed to providing documentation that meets your needs. To help us improve the documentation, send any errors, suggestions, or comments to Documentation Feedback (hpe-aruba-techpub-india@hpe.com). When submitting your feedback, include the document title, part number,

edition, and publication date located on the front cover of the document. For online help content, include the product name, product version, help edition, and publication date located on the legal notices page.