



SD-WAN

REST API

Disclaimer

This Lab Guide is designed to assist candidates to facilitate Technology learning. While every effort has been made to ensure that all material is as complete and accurate as possible, the enclosed material is presented on an “as is” basis. Neither the authors nor RSTForum assume any liability or responsibility to any person or entity with respect to loss or damages incurred from the information contained in this Lab guide. This workbook was developed by RSTForum. Any similarities between material presented in this Lab Guide and any other Lab Guide or any other material is completely coincidental.



Cisco SD-WAN REST API.

Cisco SD-WAN vManage provides a REST API interface and its functionality over REST API. The REST API documentation for cisco vManage can be found at the following link:

https://sdwan-docs.cisco.com/Product_Documentation/Command_Reference/Command_Reference/vManage_REST_APIs

A Swagger based documentation is also provided at the following URL:

<https://30.0.0.100:8443/apidocs>

This documentation gives you the option of interacting with the API and to see how the schema of the API is organized.

Internal Server Error

Try it out! Hide Response

Request URL

```
https://30.0.0.100:8443/dataservice/system/device/vedges
```

Response Body

```
{
  "header": {
    "generatedOn": 1592406589239,
    "viewKeys": {
      "uniqueKey": [],
      "preferenceKey": "grid-vEdgeList"
    },
  },
  "columns": [
    {
      "title": "State",
      "property": "vedgeCertificateState",
      "display": "iconAndToolTip",
      "iconProperty": "vedgeCertificateState",
      "toolTipProperty": "vedgeCertificateState",
      "icon": [
        {
          "key": "tokenGenerated",
          "value": "images/certStates/token-generated.png",
          "displayKey": "Token Generated"
        }
      ]
    }
  ]
}
```

Response Code

```
200
```

Response Headers

```
{
  "cache-control": "no-cache, no-store, must-revalidate",
  "connection": "keep-alive",
  "content-encoding": "gzip",
  "content-type": "application/json",
  "date": "Wed, 17 Jun 2020 15:09:49 GMT",
  "pragma": "no-cache",
  "strict-transport-security": "max-age=31536000; includeSubDomains",
  "transfer-encoding": "chunked",
  "vary": "Accept-Encoding",
  "x-content-type-options": "nosniff",
  "x-frame-options": "DENY",
  "x-xss-protection": "1; mode=block"
}
```



Step 1: Exploring Swagger Interface

Explore the API by trying out some REST calls in the swagger interface. The username is admin and password is admin.

Monitoring - Device Details

try out the GET call for the /device endpoint that can be found under Monitoring - Device Details.

Method	Endpoint	Description
GET	/data/device/state/{state_data_type}	Retrieve device state data
GET	/data/device/state/{state_data_type}/query	Retrieve device state data files
GET	/data/device/state/{state_data_type}/fields	Retrieve device state data files
GET	/device/status	Retrieve device status
GET	/device/unreachable	Retrieve list of unreachable devices of type vedge, vsmart, or vbond.
GET	/device/hardwarehealth/summary	Retrieve hardware health summary
GET	/device/config/html	Retrieve device running configuration in HTML
POST	/device/syncall/memorydb	Synchronize memory database for all devices
GET	/device/sync_status	Retrieve list of currently syncing devices
GET	/device/reachable	Retrieve list of reachable devices
GET	/device/hardwarehealth/detail	Retrieve detailed hardware health
GET	/device/vedgeinventory/detail	Retrieve detailed vEdge inventory
GET	/device/vedgeinventory/summary	Retrieve vEdge inventory
GET	/device/keyvalue	Retrieve all device monitoring details of the devices present in the system as key value
GET	/device/tlocutil	Retrieve TLOC list
GET	/device/tloc	Retrieve TLOC status
GET	/device/tlocutil/detail	Retrieve detailed TLOC list
GET	/device/queues	Retrieve synchronized queue information
GET	/device/stats	Retrieve synchronized queue information
DELETE	/device/unreachable/{deviceIP}	Delete unreachable device
POST	/device/enableSDAVC/{deviceIP}/{enable}	Enable/Disable SDAVC on device
GET	/device	Retrieve list of all devices

Click on /devices

the output will be as below



it secure | 30.0.0.100:8443/apidocs/#!/Monitoring_-_Device_Details/listAllDevices

POST /device/enableSDAVC/{deviceIP}/{enable} Enable/Disable SDAVC on device

GET /device Retrieve list of all devices

Implementation Notes
Retrieve list of all devices

Response Class (Status)
Model Model Schema

Response Content Type application/json

Response Messages

HTTP Status Code	Reason	Response Model
200	Success	
400	Bad Request	<pre> { "error": { "message": "", "details": "", "code": "" } } </pre>
403	Forbidden	
500	Internal Server Error	<pre> { "error": { "message": "", "details": "", "code": "" } } </pre>

Try it out!

GET /device/models/{uuid} Retrieve the device model for the device

GET /device/models Retrieve all device models that are supported by the vManage along with its details

Click on Try it out

You will see request URL created for Rest API call to get devices details

<https://30.0.0.100:8443/dataservice/device>

the following output shows the Response Body and Response Header received for this request.

Response code 200 means success.

The output will be in JSON format as seen below:



Try it out! [Hide Response](#)

Request URL

```
https://30.0.0.100:8443/dataservice/device
```

Response Body

```
{
  "header": {
    "generatedOn": 1592409892335,
    "viewKeys": {
      "uniqueKey": [
        "system-ip"
      ],
      "preferenceKey": "grid-Device"
    },
    "columns": [
      {
        "title": "Hostname",
        "property": "host-name",
        "display": "iconAndText",
        "iconProperty": "device-type",
        "hideable": false,
        "icon": [
          {
            "key": "vmanage",
            "value": "images/vmanage_table.png"
          }
        ]
      }
    ]
  }
}
```

Response Code

```
200
```

Response Headers

```
{
  "cache-control": "no-cache, no-store, must-revalidate",
  "connection": "keep-alive",
  "content-encoding": "gzip",
  "content-type": "application/json",
  "date": "Wed, 17 Jun 2020 16:04:52 GMT",
  "pragma": "no-cache",
  "strict-transport-security": "max-age=31536000; includeSubDomains",
  "transfer-encoding": "chunked",
  "vary": "Accept-Encoding",
  "x-content-type-options": "nosniff",
  "x-frame-options": "DENY",
  "x-xss-protection": "1; mode=block"
}
```

We can see that the output of this call is verbose and contains a lot of useful information.



Monitoring - Device Config

try out the GET call for the /device config that can be found under Monitoring – Device Details.

secure | 30.0.0.100:8443/apidocs/#!/Monitoring_-_Device_Details/getDeviceRunningConfig

GET /device/config Retrieve device running configuration

Implementation Notes
Retrieve device running configuration

Parameters

Parameter	Value	Description	Parameter Type	Data Type
deviceId	<input type="text" value="(required)"/>	Device ID	query	array[string]

Response Messages

HTTP Status Code	Reason	Response Model
200	Success	
400	Bad Request	Model Model Schema <pre>{ "error": { "message": "", "details": "", "code": "" } }</pre>
403	Forbidden	
500	Internal Server Error	Model Model Schema <pre>{ "error": { "message": "", "details": "", "code": "" } }</pre>

[Try it out!](#)

For the deviceId parameter give the ip address of device that we want to monitor.

We will use IP address of our VManage device to monitor its configuration.

In this example we will use the web browser to send rest api get request.

Type following request on your browser

<https://30.0.0.100:8443/dataservice/device/config?deviceId=123.123.123.123>



```
system
host-name          vmanage
system-ip          123.123.123.123
site-id            123
admin-tech-on-failure
organization-name  rstforum
vbond 200.0.3.200
aaa
auth-order local radius tacacs
usergroup basic
  task system read write
  task interface read write
!
usergroup netadmin
!
usergroup operator
  task system read
  task interface read
  task policy read
  task routing read
  task security read
!
usergroup tenantadmin
!
user admin
  password $6$lEedTgAXN.dbFv1P$PPkraQ0xwqnNScLfo.oTRVj/V1Hnr771LDLfbmgD5By4X9e/TKVjTY4tI2iFhnyXak8h1SfLeJP8.VYHzBq1E/
!
logging
disk
  enable
!
!
vpn 0
interface eth1
  ip address 200.0.5.123/24
  tunnel-interface
  allow-service dhcp
  allow-service dns
  allow-service icmp
  no allow-service sshd
  no allow-service netconf
  no allow-service ntp
  no allow-service stun
  allow-service https
!
  no shutdown
!
  ip route 0.0.0.0/0 200.0.5.10
!
vpn 512
interface eth0
  ip address 30.0.0.100/24
  no shutdown
!
!
```

section We can see that the output of this call is verbose and contains a lot of useful information.

