



SD-AVC REST API

- [REST API Overview, on page 1](#)
- [Authentication from SD-AVC Network Service, on page 3](#)
- [System, on page 4](#)
- [External Sources, on page 6](#)
- [User-defined Applications, on page 7](#)
- [Generic Applications, on page 16](#)
- [REST API Notes and Limitations, on page 16](#)
- [User-defined Application Attribute Values, on page 16](#)

REST API Overview

The REST API provides numerous system functions, including:

- Displaying information about devices in the SD-AVC network
- Controlling external sources
- Displaying information about generic traffic
- Creating user-defined applications



Note Using the REST API requires authentication. See [Authentication from SD-AVC Network Service, on page 3](#).

Table 1: REST API Functionality

Authentication	
POST https://<SD-AVC-network-service-address>:8443/avc-sd-service/ external-api/login	Acquires an authentication token, enabling use of the REST API. Authentication from SD-AVC Network Service, on page 3
System	

GET /avc-sd-service/external-api/status	Displays the SD-AVC version and system times. Display System Version and System Times, on page 4
GET /avc-sd-service/external-api/devices	Displays devices in the SD-AVC network. Display Devices, on page 4
POST /avc-sd-service/external-api/cleanDevices	Removes a device from the SD-AVC network. Delete Devices from SD-AVC, on page 5
GET /avc-sd-service/external-api/analytics?period= <i>Period</i> - GET /avc-sd-service/external-api/analytics/ <i>SegmentName</i> ?period= <i>Period</i> - GET /avc-sd-service/external-api/analytics/ <i>SegmentName</i> / <i>DeviceName</i> ?period= <i>Period</i>	Display traffic analytics (applications and bandwidth) for the complete SD-AVC network, a specific segment, or a specific device. Display Traffic Analytics, on page 5
External Sources	
GET /avc-sd-service/external-api/configuration/externalSource/ <i>externalSourceName_enabled</i> /start - GET /avc-sd-service/external-api/configuration/externalSource/ <i>externalSourceName_enabled</i> /stop	Enables or disables receiving data from an external source. In this release, the only external source is the MS Office365 Connector (o365). Enable/Disable External Sources, on page 6
GET /avc-sd-service/external-api/configuration	Displays status of external sources. Display Status of External Sources, on page 7
User-defined Applications	
POST /avc-sd-service/external-api/app-rules	Create one or more user-defined applications. Create User-defined Application Rules, on page 8
GET /avc-sd-service /external-api/app-rules - GET /avc-sd-service /external-api/app-rules?sourceId= <i>sourceId</i>	Displays user-defined applications defined by REST API. Display User-defined Application Rules, on page 14

GET /avc-sd-service/external-api/app-rules/status - GET /avc-sd-service/external-api/app-rules/status[?sourceId=sourceId]	Displays activation status of user-defined applications, per device. Display User-defined Application Status, on page 15
DELETE /avc-sd-service /external-api/app-rules - DELETE /avc-sd-service/external-api/app-rules?sourceId=sourceId	Deletes a set of user-defined applications. Delete User-defined Applications, on page 15
Generic Applications	
GET /avc-sd-service/external-api/apps/generics	Displays the list of traffic types that contribute to "generic" traffic. Display Generic Application Traffic Types, on page 16

Authentication from SD-AVC Network Service

Using the REST API requires a token-based authentication from the SD-AVC network service. To acquire an authentication token:

1. Send the following HTTP request to the API:

POST https://<SD-AVC-network-service-address>:8443/avc-sd-service/external-api/login

Example:

POST https://192.168.0.1:8443/avc-sd-service/external-api/login

- In the request header, include the following key:

Content-Type: application/x-www-form-urlencoded

- In the request body, include the following keys, providing login credentials:

username: *username*

password: *password*

2. The API response body provides an authentication token. Use the token to authorize REST API calls to the SD-AVC network service.



Note The token expires after 12 hours.

Example token:

```
{
  "token": "Bearer eyJhbGciOiJIUzUxMiJ9.eyJqdGkiOiJhYjZkGGUxOS0zMmU3LTRlY2ItYWQ5OC1kYmVmZTdja2E5YzYiLCJzZWl0OiJsYWl0LCJleHAiOiJlMzAwMqk1MzJ9.EfP3wd4fZbWrOQ6Sskh-I0bbPffF4NaruB-o_OV0EQ7fwMwfmuUNP00R58fRGKkYWR3tQu8HjovDp37EPtD15Q"
}
```

System

System: Overview

The REST API can display information about the SD-AVC system, and make changes to the configuration.

Display System Version and System Times

API:

GET /avc-sd-service/external-api/status

Description:

Displays:

- Current time: Time in UNIX format.
- System uptime: SD-AVC uptime in milliseconds.
- SD-AVC version

Example Response:

```
{"currentTime":1536757252625,"systemUpTime":"57214777","version":"2.1.1"}
```

Display Devices

API:

GET /avc-sd-service/external-api/devices

Description:

Displays the devices in the SD-AVC network, organized by segment, in JSON format. The response includes errors and warnings, and additional information per device.

Response:

The output shows errors and warnings for:

- total network
- each segment
- each device

Example Response:

The example represents a network with one segment (datacenter-01) and one device within that segment (asr-device-100).

```
{
  "total":{
    "connection":{
      "error":[]
    }
  }
}
```

```

        "warn": []
      },
    },
    "segments": [
      {
        "name": "datacenter-01",
        "connection": {
          "error": [],
          "warn": []
        },
        "devices": [
          {
            "name": "asr-device-100",
            "ip": "192.168.1.0",
            "connection": {
              "error": [],
              "warn": []
            }
          }
        ]
      }
    ]
  }
}

```

Delete Devices from SD-AVC

API:

POST /avc-sd-service/external-api/cleanDevices

Description:

Removes a device from the SD-AVC network. Specify the device and segment in the body.

Body:

```
{"devices":["DeviceName"],"segment":"SegmentName"}
```

Example Body:

```
{"devices":["SDAVC-ASR-200","SDAVC-ASR-201"],"segment":"datacenter01"}
```

Example Response:

```
{"success":true,"message":"2 devices from segment datacenter01 were deleted successfully"}
```

Display Traffic Analytics

API:

GET /avc-sd-service/external-api/analytics?period=*Period*

-

GET /avc-sd-service/external-api/analytics/*SegmentName*?period=*Period*

-

GET /avc-sd-service/external-api/analytics/*SegmentName*/*DeviceName*?period=*Period*

Description:

Displays traffic analytics (applications and bandwidth) for the complete SD-AVC network, a specific segment, or a specific device. Optionally, specify a period for the analytics. The response includes:

- Application name and bandwidth (bytes) used by the application
- Total bandwidth (bytes) used

Table 2: Properties

Property	Description
SegmentName	(Optional) Specifies a segment. Response includes only analytics from this segment.
DeviceName	(Optional) Specifies a device. Response includes only analytics from this device.
Period	Use <code>?period=Period</code> to specify the period to include in the analytics. Possible values for <i>Period</i> : 120, 720, 1440, 2880 minutes (These correspond to 2, 12, 24, and 48 hours.)

Example:

```
GET /avc-sd-service/external-api/analytics/datacenter01/device-300?period=1440
```

Example Response:

```
{ "apps": [ { "name": "vmware-vsphere", "bandwidth": 226331127989634 }, { "name": "telepresence-control", "bandwidth": 146787859067274 }, { "name": "unknown", "bandwidth": 132586088501412 }, { "name": "google-services", "bandwidth": 122981674585168 } ], "totalBandwidth": 628686750143488 }
```

External Sources

External Sources: Overview

External sources provide additional application information that SD-AVC uses for classifying network traffic. An example is the MS-Office365 Connector ([MS-Office365 Connector](#)).

Enable/Disable External Sources

API:

```
GET /avc-sd-service/external-api/configuration/externalSource/externalSourceName_enabled/start
```

```
GET /avc-sd-service/external-api/configuration/externalSource/externalSourceName_enabled/stop
```

Description:

Enables or disables receiving data from an external source.

Table 3: Properties

Property	Description
externalSourceName	(Mandatory) Name of the external source. Note In this release, the only external source that operates with SD-AVC is "o365" (MS-Office365 Connector).

Examples:

```
GET /avc-sd-service/external-api/configuration/externalSource/o365_enabled/start
```

```
GET /avc-sd-service/external-api/configuration/externalSource/o365_enabled/stop
```

Display Status of External Sources

API:

```
GET /avc-sd-service/external-api/configuration
```

Description:

Displays the status (enabled/disabled) of external sources.

Example:

```
GET /avc-sd-service/external-api/configuration
```

Response:

In the response, **0** is disabled, **1** is enabled.

Example Response:

In this example, MS Office 365 Connector is enabled.

```
{"o365_enabled": "1"}
```

User-defined Applications

User-defined Applications: Overview

Network devices operating with SD-AVC use Cisco NBAR2 and other tools to identify network traffic. The composite of information that NBAR2 uses to identify a network applications is called an "application" (or a "protocol" in the Protocol Packs released periodically by Cisco). User-defined applications may be specified on individual devices by CLI, or network-wide using SD-AVC.

Each application includes:

- **Signature:** Details that identify the network application

- **Attributes:** Assigned characteristics of the application, such as business-relevance, used for visibility and QoS policy. See [User-defined Application Attribute Values](#), on page 16.

Table 4: Application Types

Application Type	Description
Protocol Pack applications	Standard applications provided by Cisco in a Protocol Pack.
User-defined applications on individual devices	Defined by CLI on individual devices, sometimes called custom protocols.
Network-wide user-defined applications	Defined by SD-AVC REST API. These appear on the SD-AVC Dashboard > External Sources page.

SD-AVC User-defined Applications

SD-AVC can provision user-defined applications at the network level, available for all participating devices in the network. In effect, this is similar to adding user-defined applications manually on each device.

Create User-defined Application Rules

API:

POST /avc-sd-service/external-api/app-rules

Description:

Defines one or more user-defined applications.

Body:

Body must include the full set of user-defined applications. Executing the API overwrites any currently defined user-defined applications for the specified source (sourceId).

```
{
  "sourceId": string,
  "rules": [{
    "allSegments": Boolean,
    "segment": string,
    "rules": [{
      "appName": string,
      "serverNames": [string],
      "L3L4": [{
        "ipAddresses": [string],
        "ports": [integer(s) or range],
        "l4Protocol": string,
        "vrf": string
      }],
      "attributes": {
        "category": string,
        "sub-category": string,
        "application-group": string,
        "business-relevance": string,
        "traffic-class": string,
        "application-set": string
      }
    }],
  }]
```



```
    }
  }
```

Table 5: Top-level Properties

Property	Description
sourceId	(Mandatory) ID of the external source. Note In the initial release of the REST API, only one source is supported.
rules	(Mandatory) Contains complete list of the user-defined application rules. Note This property contains a sub-property also called rules.

Table 6: Sub-properties of rules

Property	Description
allSegments	(Must include either allSegments or segment.) Set to true to apply the user-defined applications to all segments, not only one segment. Possible values: true, false (default)
segment	(Must include either allSegments or segment.) List of user-defined application rules for a specific SD-AVC segment.
rules	(Mandatory) List of segment rules.

Table 7: Sub-properties of rules > rules

Property	Description
appName	(Mandatory) Name of user-defined application, reflecting name of the network application. Note Do not use a name that conflicts with an existing application, such as one defined in the Protocol Pack.
serverNames	(Must include at least one of serverNames, L3L4, and attributes.) List of all server names (FQDNs, SNIs, ...) for the network application. Note Server names are case-sensitive.

Property	Description
L3L4	(Must include at least one of serverNames, L3L4, and attributes.) List of all IP-based rules. (See sub-properties below.)
attributes	(Must include at least one of serverNames, L3L4, and attributes.) Attributes to assign to the application. (See sub-properties below.)

Table 8: Sub-properties of rules > rules > L3L4

Property	Description
IpAddresses	(Mandatory) List of IPs. Can be both normal IP and subnet (using CIDR notation).
ports	Port(s) or port range. Examples: "ports": [23] "ports": [23,24] "ports": [23, "25-30"]
I4Protocol	Transport layer protocol. Possible values: TCP, UDP, TCP-UDP
vrf	VRF name.

Table 9: Sub-properties of rules > rules > attributes

Property	Description
application-set	(Must include at least one of serverNames, L3L4, and attributes.) Attributes to assign to the application.
application-group	(Defining a partial list of attributes is supported. If attributes is included, must include at least one of these properties.) See User-defined Application Attribute Values , on page 16.
category	
sub-category	
traffic-class	
business-relevance	

Response:

Response code 200 indicates success.

In case of failure, the response body provides information about the reason for failure.

Example 1: Single domain name

This example shows:

- 1 network segment: datacenter01
- 1 user-defined application: myDocs
- 1 server name
- No attributes specified

```
{
  "sourceId": "mySource",
  "rules": [{
    "segment": "datacenter01",
    "rules": [{
      "appName": "myDocs",
      "serverNames": [
        "www.myApp.com"
      ]
    }]
  }]
}
```

Example 2: Three IP addresses and ports

This example shows:

- 1 network segment: datacenter01
- 1 user-defined application: myDocs
- 3 IP addresses and 3 ports
- No attributes specified

```
{
  "sourceId": "mySource",
  "rules": [{
    "segment": "datacenter01",
    "rules": [{
      "appName": "myDocs",
      "L3L4": [{
        "ipAddresses": ["2.2.2.2"],
        "ports": [20],
        "l4Protocol": "TCP"
      },
      {
        "ipAddresses": ["3.3.3.3"],
        "ports": [30],
        "l4Protocol": "TCP"
      },
      {
        "ipAddresses": ["4.4.4.4"],
        "ports": [40],
        "l4Protocol": "TCP"
      }
    ]
  }]
}
```

Example 3: Two user-defined applications in one network segment

```

    }}
  }}
}

```

Example 3: Two user-defined applications in one network segment

This example shows:

- 1 network segment: datacenter01
- 2 user-defined applications: myDocs and myTelepresence
- No attributes specified for the myDocs user-defined application
- business-relevance attribute specified for the myTelepresence user-defined application
- IP address with subnet specified
- Individual ports and a range of ports

```

{
  "sourceId": "mySource",
  "rules": [{
    "segment": "datacenter01",
    "rules": [{
      "appName": "myDocs",
      "serverNames": [
        "www.myApp.com"
      ],
      "l3l4": [{
        "ipAddresses": ["10.1.1.0/24", "2.2.2.2"],
        "ports": [23, 34, "37-42"],
        "l4Protocol": "TCP",
        "vrf": "vrf1"
      }]
    }],
  },
  {
    "appName": "myTelepresence",
    "l3l4": [{
      "ipAddresses": ["2.2.2.2"],
      "ports": [35],
      "l4Protocol": "TCP"
    }],
    "attributes": {
      "business-relevance": "business-relevant"
    }
  }
]
}}
}

```

Example 4: User-defined applications in two network segments

This example shows:

- 2 network segments: datacenter01, datacenter02
- 3 user-defined applications: myDocs, myTelepresence, myEnterpriseIM
- No attributes specified for: myDocs, myEnterpriseIM

- business-relevance attribute specified for myTelepresence
- IP address with subnet specified
- Individual ports and a range of ports

```
{
  "sourceId": "mySource",
  "rules": [{
    "segment": "datacenter01",
    "rules": [{
      "appName": "myDocs",
      "serverNames": [
        "www.myDocs.com"
      ],
      "L3L4": [{
        "ipAddresses": ["10.1.1.0/24", "2.2.2.2"],
        "ports": [23, 34, "37 - 42"],
        "l4Protocol": "TCP",
        "vrf": "vrf1"
      }]
    }],
    {
      "appName": "myTelepresence",
      "L3L4": [{
        "ipAddresses": ["2.2.2.2"],
        "ports": [35],
        "l4Protocol": "TCP"
      }],
      "attributes": {
        "business-relevance": "business-relevant"
      }
    }
  ]
},
{
  "segment": "datacenter02",
  "rules": [{
    "appName": "myEnterpriseIM",
    "serverNames": [
      "www.myEnterpriseIM.com"
    ],
    "L3L4": [{
      "ipAddresses": ["2.2.2.10"],
      "ports": [23],
      "l4Protocol": "TCP"
    }]
  }]
}
]
```

Example 5: Using allSegments and specific network segments

This example shows:

- 2 user-defined applications (myDocs, myTelepresence) for all network segments, using allSegments
- User-defined application (myEnterpriseIM) only for 1 network segment: datacenter02
- 3 user-defined applications: myDocs, myTelepresence, myEnterpriseIM

- No attributes specified for: myDocs, myEnterpriseIM
- business-relevance attribute specified for myTelepresence
- IP address with subnet specified
- Individual ports and a range of ports

```
{
  "sourceId": "mySource",
  "rules": [{
    "allSegments": true,
    "rules": [{
      "appName": "myDocs",
      "serverNames": [
        "www.myApp.com"
      ],
      "L3L4": [{
        "ipAddresses": ["10.1.1.0/24", "2.2.2.2"],
        "ports": [23, 34, "37 - 42"],
        "l4Protocol": "TCP",
        "vrf": "vrf1"
      }]
    }],
    {
      "appName": "myTelepresence",
      "L3L4": [{
        "ipAddresses": ["2.2.2.2"],
        "ports": [35],
        "l4Protocol": "TCP"
      }],
      "attributes": {
        "business-relevance": "business-relevant"
      }
    }
  ]
},
{
  "segment": "datacenter02",
  "rules": [{
    "appName": "myEnterpriseIM",
    "serverNames": [
      "www.myEnterpriseIM.com"
    ],
    "L3L4": [{
      "ipAddresses": ["2.2.2.10"],
      "ports": [23],
      "l4Protocol": "TCP"
    }]
  }]
}
]
```

Display User-defined Application Rules

API:

GET /avc-sd-service /external-api/app-rules

GET /avc-sd-service /external-api/app-rules?sourceId=*sourceId*

Description:

Displays the user-defined applications defined by REST API.

Response:

The response lists the user-defined applications defined for a single source or all sources. The response body uses the same JSON structure as POST.

If no *sourceId* is specified, the response lists the user-defined applications for all sources.

If *sourceId* is specified, the response lists the user-defined applications for the specified source. The *sourceId* is user-defined by POST when defining user-defined applications.



Note In the initial release of the REST API, only one source is supported.

Display User-defined Application Status

API:

GET /avc-sd-service/external-api/app-rules/status

GET /avc-sd-service /external-api/app-rules/status[?sourceId=*sourceId*]

Description:

The SD-AVC network service sends the user-defined applications defined by REST API to the devices in the network. This API displays the activation status of the applications, per device.

If *sourceId* is specified, the output is limited to that source. The *sourceId* is user-defined by POST when defining user-defined applications.



Note In the initial release of the REST API, only one source is supported.

Response:

The response lists each network device, arranged by segment. For each device:

- ID/version of application rules currently loaded on the device
- Status: SUCCESS, FAILED, IN-PROGRESS

Delete User-defined Applications

API:

DELETE /avc-sd-service /external-api/app-rules

DELETE /avc-sd-service /external-api/app-rules?sourceId=*sourceId*

Description:

Deletes a set of user-defined applications.

If no *sourceId* is specified, deletes the full set of user-defined applications.

If *sourceId* is specified, deletes the full set of user-defined applications for the specified source. The *sourceId* is user-defined by POST when defining user-defined applications.



Note In the initial release of the REST API, only one source is supported.

Response:

Response code 200 indicates success.

Generic Applications

Generic Applications: Overview

"Generic" network traffic is not attributed to a specific network application. This portion of network traffic reduces the classification index, which is shown in the SD-AVC Dashboard.

Display Generic Application Traffic Types

API:

GET /avc-sd-service/external-api/apps/generics

Description:

Displays the list of traffic types that contribute to generic traffic. The response is preconfigured - it does not depend on current traffic.

Response:

```
["statistical-conf-audio", "rtp-audio", "spdy", "statistical-p2p", "rtp-video", "http", "statistical-conf-video", "quic", "statistical-download", "ssl", "unknown", "rtp"]
```

REST API Notes and Limitations

See [SD-AVC Notes and Limitations](#).

User-defined Application Attribute Values

When creating new user-defined applications using the SD-AVC REST API, use the following attribute values.



Note This list is provided for convenient reference, but may not be comprehensive.

Attribute	Possible Values
application-set	authentication-services backup-and-storage collaboration-apps consumer-browsing consumer-file-sharing consumer-gaming consumer-media consumer-misc consumer-social-networking database-apps desktop-virtualization-apps email enterprise-ipc file-sharing general-browsing general-media general-misc local-services naming-services network-control network-management remote-access saas-apps signaling software-development-tools software-updates streaming-media tunneling

Attribute	Possible Values
application-group	aol-group apple-group apple-talk-group banyan-group bittorrent-group capwap-group cisco-jabber-group cisco-phone-group corba-group dameware-group edonkey-emule-group espn-group fasttrack-group flash-group fring-group ftp-group gnutella-group google-group gtalk-group hangouts-group icq-group imap-group ipsec-group irc-group kakao-group kerberos-group ldap-group

Attribute	Possible Values
application-group (continued)	ms-cloud-group ms-crm-group ms-lync-group msn-messenger-group netbios-group nntp-group npmp-group other pop3-group prm-group qq-group skype-group smtp-group snmp-group sqlsvr-group stun-group telepresence-group tftp-group vmware-group vnc-group wap-group webex-group xns-xerox-group xunlei-group yahoo-group yahoo-messenger-group

Attribute	Possible Values
category	anonymizers backup-and-storage browsing business-and-productivity-tools consumer-file-sharing consumer-internet consumer-messaging consumer-streaming database email epayment file-sharing gaming industrial-protocols instant-messaging inter-process-rpc internet-security layer3-over-ip location-based-services net-admin newsgroup other social-networking software-updates trojan voice-and-video

Attribute	Possible Values
sub-category	authentication-services backup-systems consumer-audio-streaming consumer-cloud-storage consumer-multimedia-messaging consumer-video-streaming consumer-web-browsing control-and-signaling desktop-virtualization enterprise-cloud-data-storage enterprise-cloud-services enterprise-data-center-storage enterprise-media-conferencing enterprise-realtime-apps enterprise-rich-media-content enterprise-sw-deployment-tools enterprise-transactional-apps enterprise-video-broadcast enterprise-voice-collaboration file-transfer naming-services network-management os-updates other p2p-file-transfer p2p-networking remote-access-terminal routing-protocol tunneling-protocols

Attribute	Possible Values
traffic-class	broadcast-video bulk-data multimedia-conferencing multimedia-streaming network-control ops-admin-mgmt real-time-interactive signaling transactional-data voip-telephony
business-relevance	business-irrelevant business-relevant default