# API Cross-Site Request Forgery Prevention

*Table 1: Feature History*

| Feature Name | Release Information | Description |
|---|---|---|
| API Cross-Site Request Forgery Prevention | Cisco IOS XE Catalyst SD-WAN Release 16.12.1b<br>Cisco Catalyst SD-WAN Release 19.2.1 | This feature adds protection against Cross-Site Request Forgery (CSRF) that occurs when using Cisco Catalyst SD-WAN REST APIs. This protection is provided by including a CSRF token with API requests. You can put requests on an allowed list so that they do not require protection if needed. |


- Cisco Catalyst SD-WAN REST API Token-Based Authentication, on page 1
- Token Use, on page 2
- API Docs, on page 2
- Third Party Application Users, on page 2


# Cisco Catalyst SD-WAN REST API Token-Based Authentication

Cisco Catalyst SD-WAN release 19.2 offers token-based authentication when you use the Cisco Catalyst SD-WAN REST API. This protection is provided by requiring that a token be included with API requests. Each API session uses a unique token that is valid throughout the session. If an API request does not include this token, Cisco SD-WAN Manager rejects the request, unless the endpoint is included on an allowed list. (For assistance with adding endpoints to an allowed list, open a case with the Cisco TAC or escalation support team.)

**Note** However, some of the GET API's and all the POST APIs of Cisco SD-WAN Manager, which are not on an allowed list require Cross-Site Request Forgery (CSRF) token authentication.

# Token Use

The following sections describe how the token is used with the API when you use API docs or third party applications.

# API Docs

Cisco SD-WAN Manager automatically generates a token and appends the token to every request that you send from the Cisco SD-WAN Manager API Docs page. This process requires no action from you, and you will not notice any difference from previous releases in how the API Docs page operates.

If there are API requests that you want to exclude from this token-based authentication, you can request that these API endpoints be included in an allowed list by opening a case with the Cisco TAC or escalation support team.

# Third Party Application Users

If you use scripts or third party applications such as Postman, LiveAction, SolarWinds, or SevOne for Cisco SD-WAN Manager API requests, each request must include the token, unless the API is included in an allowed list. If an API request does not include a token and is not included in the allowed list, Cisco SD-WAN Manager rejects the request and returns the response code 403 (forbidden) with the message, “SessionTokenFilter: Token provided via HTTP Header does not match the token generated by the server.”

To request that certain API endpoints be included in an allowed list, open a case with the Cisco TAC or escalation support team.

To include the token in a third party API request:

### Method 1

In the first method, the session you create is stored in the cookies.txt file and the same session can be used for all subsequent requests, using the jsessionid that the file contains. This is the recommended method.

1. To log in to Cisco SD-WAN Manager, use the following example command and modify the URL according to your IP address:

   ```
   sampleuser$ TOKEN=$(curl "https://209.165.200.254/dataservice/client/token" -X GET -b
   cookies.txt -s  -insecure)
   ```

   To verify the login, see the cookies.txt file.

2. After logging in to Cisco SD-WAN Manager, obtain a token by making a request, where *vManage_IP* is the IP address of your Cisco SD-WAN Manager server. You can obtain a token in string format or in JSON format.

   To obtain a token in string format, use the following URL:

   https://*vManage_IP*/dataservice/client/token

   To obtain a token in JSON format (beginning with Cisco IOS XE SD-WAN Release 16.12 and Cisco SD-WAN Release 19.2), use the following URL:

https://*vManage_IP*/dataservice/client/token?json=true

The token that these calls return is valid for the rest of your current session. The following example shows requests for obtaining a token:

Command for obtaining a token in string format:

```
sampleuser$ TOKEN=$(curl "https://vManage_IP/dataservice/client/token" -X GET -b
cookies.txt -s  -insecure)
```

Output in string format:

```
FC5B19BB3521EE20CFBDCD3CEDCC48F50CB1095C9654407936029E9C0EF99FEAE50440B60E49F7CD4A0BAB5307C2855F2E0C
```

Command for obtaining a token in JSON format:

```
TOKEN=$(curl "https://vManage_IP/dataservice/client/token?json=true" -X GET -b cookies.txt
 -s  -insecure)
```

Output in JSON format:

```
sampleuser$ echo $TOKEN

{"token":"56CF324A8F67993B6FCCF57302068B0756DA8703BE712EEA18D4D9055B11312843F9D30B48A3902320FFAA8659AD01202A63"}
```

**Note** JSON format is not supported for curl commands.

3. In the header of each subsequent API request in the current session, include the X-XSRF-TOKEN key, with a value that consists of the token that you generated.

   The following examples show a GET request and a POST request that include a generated token in the header:

   Command:

   ```
   sampleuser$ curl "https://vManage_IP/dataservice/server/info" -b cookies.txt -silent
   -insecure -H "X-XSRF-TOKEN: $TOKEN"
   ```

   Output:

   ```
   {"Achitecture":"amd64","Available processors":2}
   ```

   Command

   ```
   sampleuser$ curl
   "https://vManage_IP/dataservice/settings/configuration/emailNotificationSettings" -X
   POST -b cookies.txt -silent  -insecure -H "X-XSRF-TOKEN: $TOKEN" -d
   '{"enabled":true,"from_address":"test@mydomain.com","protocol":"smtp","smtp_server":"a.com",

   "smtp_port":25,"reply_to_address":"test@test.com","notification_use_smtp_authentication":false}='
   ```

   Output:

   ```
   {"data":[{"enabled":true,"notification_use_email_setting_authentication":false,"notification_use_smtp_authentication":false}]}
   ```

4. To prevent memory leaks, you must logout after each API call, including the token, starting from Cisco SD-WAN Release 19.2.1.

   The following example shows how you can logout:

   Command:

```
sampleuser$ curl "https://vManage_IP/logout" -b cookies.txt  -insecure -H
"X-XSRF-TOKEN:$TOKEN"
```

Output:

```
Replaced cookie JSESSIONID="DcOke5mqix_15qCpWA1blIJVAMnVg3lDMU4ABRgVinvalid" for domain
 209.165.200.254, path /, expire 0
      < set-cookie: JSESSIONID=DcOke5mqix_15qCpWA1blIJVAMnVg3lDMU4ABRgVinvalid
```

**Note** To verify that you have logged out of the session, check the jsessionid and ensure that it ends with 'invalid'.

### Method 2

In the second method, the session you create is not stored and you must create a new session for each request.

1. After logging in to Cisco SD-WAN Manager, obtain a token by making a request, where *vManage_IP* is the IP address of your Cisco SD-WAN Manager server. You can obtain a token in string format or in JSON format.

   To obtain a token in string format, use the following URL:

   https://*vManage_IP*/dataservice/client/token

   To obtain a token in JSON format (beginning with Cisco IOS XE SD-WAN Release 16.12 and Cisco SD-WAN Release 19.2), use the following URL:

   https://*vManage_IP*/dataservice/client/token?json=true

   The token that these calls return is valid for the rest of your current session. The following example shows requests for obtaining a token:

   Command for obtaining a token in string format:

   ```
   sampleuser$ curl --user admin:admin https://vManage_IP/dataservice/client/token --insecure
   ```

   Output in string format:

   ```
   FC5B19BB3521EE20CFBDCD3CEDCC48F50CB1095C9654407936029E9C0EF99FEAE50440B60E49F7CD4A0BAB5307C2855F2E0C
   ```

   Command for obtaining a token in JSON format:

   ```
   sampleuser$ curl --user admin:admin https://vManage_IP/dataservice/client/token?json=true
    --insecure
   {"token":"F1E047E444DB2CA4237B0246DFE133345584B788C6E8776F04749A371B73F3C0C683043F1CDBB5E01BBBDA7D6C35F58EA37A"}
   ```

   Output in JSON format:

   ```
   FC5B19BB3521EE20CFBDCD3CEDCC48F50CB1095C9654407936029E9C0EF99FEAE50440B60E49F7CD4A0BAB5307C2855F2E0C
   ```

2. In the header of each subsequent API request in the current session, include the X-XSRF-TOKEN key, with a value that consists of the token that you generated.

   The following examples show a GET request and a POST request that include a generated token in the header:

   Command:

   ```
   sampleuser$ curl "https://vManage_IP/dataservice/server/info" -H "Cookie:
   JSESSIONID=pSwrx3AEWokiDO1TkFiOjgSehp-ITNdFn7Xj9PsL.c331d01e-91d7-41cc-ab90-b629c2ae6d97"
   --insecure -H "X-XSRF-TOKEN=
   FC5B19BB3521EE20CFBDCD3CEDCC48F50CB1095C9654407936029E9C0EF99FEAE50440B60E49F7CD4A0BAB5307C2855F2E0C"
   ```

Output:

```
{"Achitecture":"amd64","Available processors":2}
```

Command

```
sampleuser$
"https://vManage_IP/dataservice/settings/configuration/emailNotificationSettings" -H
"Cookie:
JSESSIONID=pSwrx3AEWokiDO1TkFiOjgSehp-ITNdFn7Xj9PsL.c331d01e-91d7-41cc-ab90-b629c2ae6d97"
--insecure -H "X-XSRF-TOKEN=
FC5B19BB3521EE20CFBDCD3CEDCC48F50CB1095C9654407936029E9C0EF99FEAE50440B60E49F7CD4A0BAB5307C2855F2E0C"
-X POST --insecure -d
'{"enabled":true,"from_address":"test@mydomain.com","protocol":"smtp","smtp_server":"a.com",
"smtp_port":25,"reply_to_address":"test@test.com","notification_use_smtp_authentication":false}='
```

Output:

```
{"data":[{"enabled":true,"protocol":"smtp","smtp_server":"a.com","from_address":"test@mydomain.com",
"smtp_port":25,"notification_use_smtp_authentication":false,"reply_to_address":"test@test.com"}]}
```

3. To prevent memory leaks, you must logout after each API call, including the token, starting from Cisco SD-WAN Release 19.2.1.

   The following example shows how you can logout:

   Command:

   ```
   sampleuser$ curl "https://vManage_IP/logout" -b cookies.txt  --insecure -H
   "X-XSRF-TOKEN:$TOKEN"
   ```

   Output:

   ```
   Replaced cookie JSESSIONID="DcOke5mqix_15qCpWA1blIJVAMnVg3lDMU4ABRgVinvalid" for domain
    209.165.200.254, path /, expire 0
   < set-cookie: JSESSIONID=DcOke5mqix_15qCpWA1blIJVAMnVg3lDMU4ABRgVinvalid
   ```

**Note** To verify that you have logged out of the session, check the jsessionid and ensure that it ends with 'invalid'.