

# Lab: Sử dụng Postman tương tác với SD-WAN REST API

## 1. Mô tả

- Sử dụng công cụ Postman tạo Collection để chứa các request xác thực và lấy thông tin từ SD-WAN vManage
- Sử dụng công cụ Postman thực hiện gửi các request đến SD-WAN (vManage) để lấy thông tin thiết bị, template
- Từ Collection trong Postman, tiến hành tạo request để xác thực với vManage
- Sau khi xác thực thành công thì tiếp tục tạo các request để lấy thông tin những thiết bị có trong Controller

## 2. Yêu cầu kỹ thuật

- Cài đặt công cụ Postman (Bài lab sử dụng Postman version 9.15.11)
- Kết nối đến SD-WAN vManage, thông tin Vmanage trong bài lab:
  - ❖ Host: <https://sandbox-sdwan-1.cisco.com>
  - ❖ Username: devnetuser
  - ❖ Password: RG!\_Yw919\_83
- Tạo các request để xác thực, lấy thông tin thiết bị có trong SD-WAN

## 3. Thực hiện

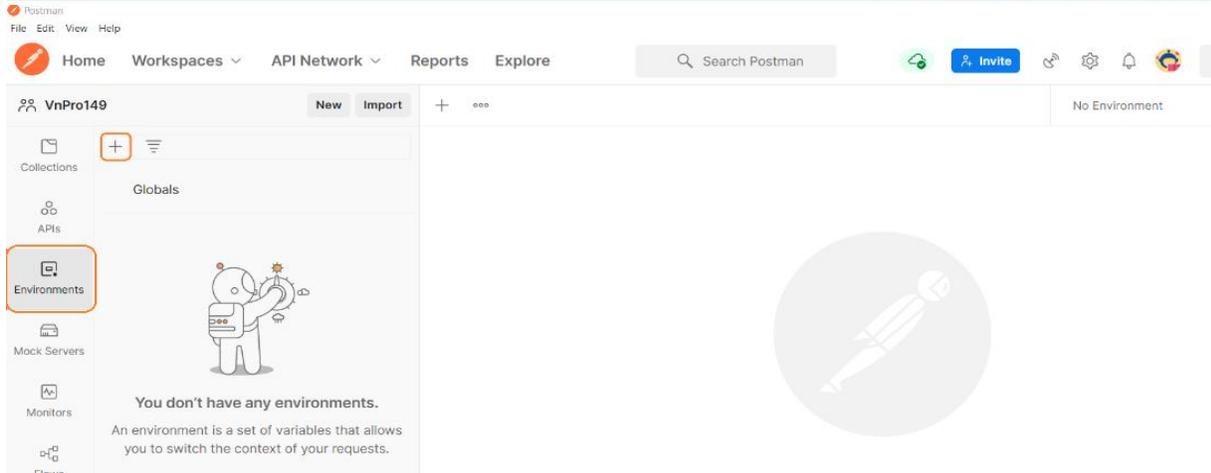
### a. Tạo môi trường

Sử dụng Environment trong Postman như một nơi để lưu trữ “biến” để có thể tái sử dụng ở nhiều nơi.

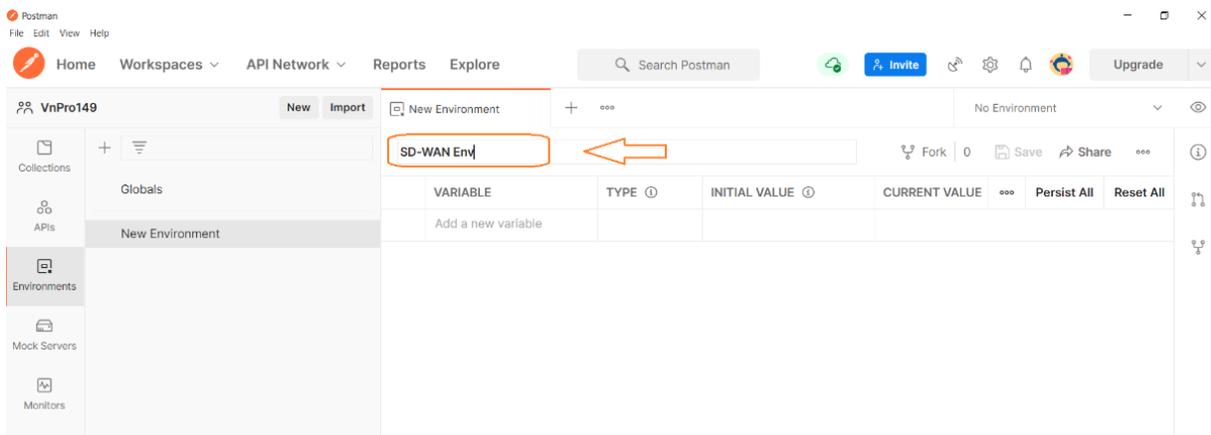
Ở Postman sẽ chia làm 2 loại Environments: Local và Global

- Local: Phạm vi ảnh hưởng chỉ có khi chọn đúng Enviroments.
- Global: Phạm vi ảnh hưởng đến toàn bộ các project có trong Postman, nhưng nếu có 2 biến cùng tên ở Local và Global thì sẽ ưu tiên lấy Local.

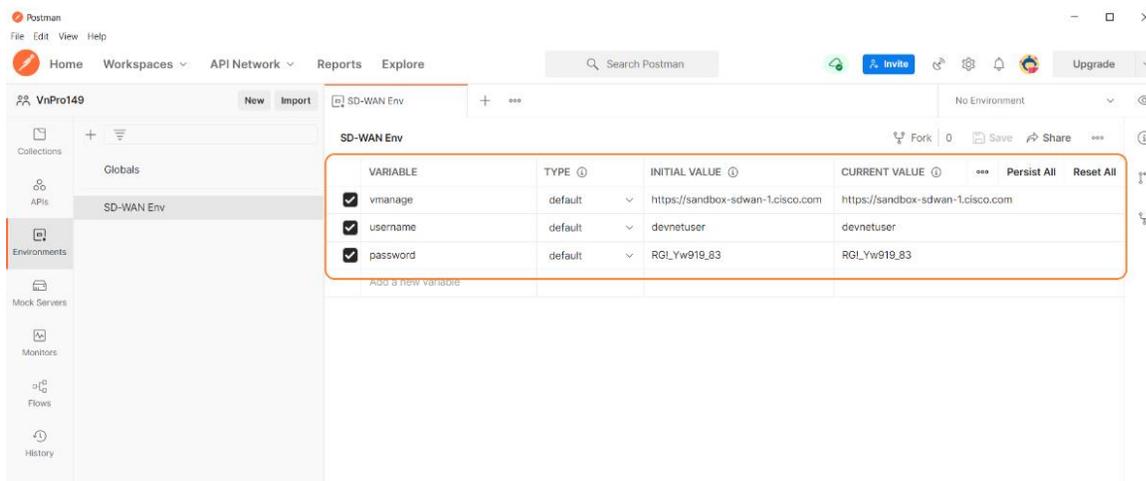
**Bước 1:** Để tạo môi trường, bạn cần vào phần Environment sau đó ấn vào dấu cộng phía góc trái



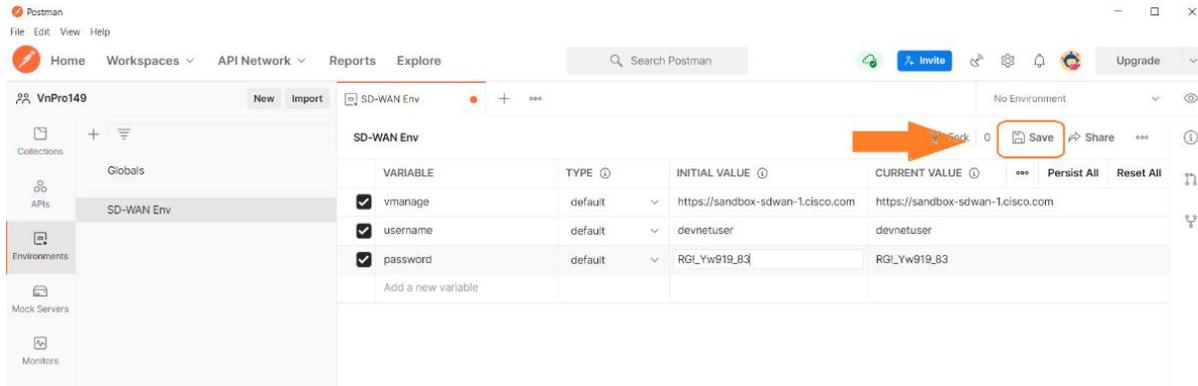
## Bước 2: Nhập tên môi trường



## Bước 3: Điền thông tin truy cập của vManage vào môi trường để có thể sử dụng nhiều nơi



## Bước 4: Nhấn save để lưu

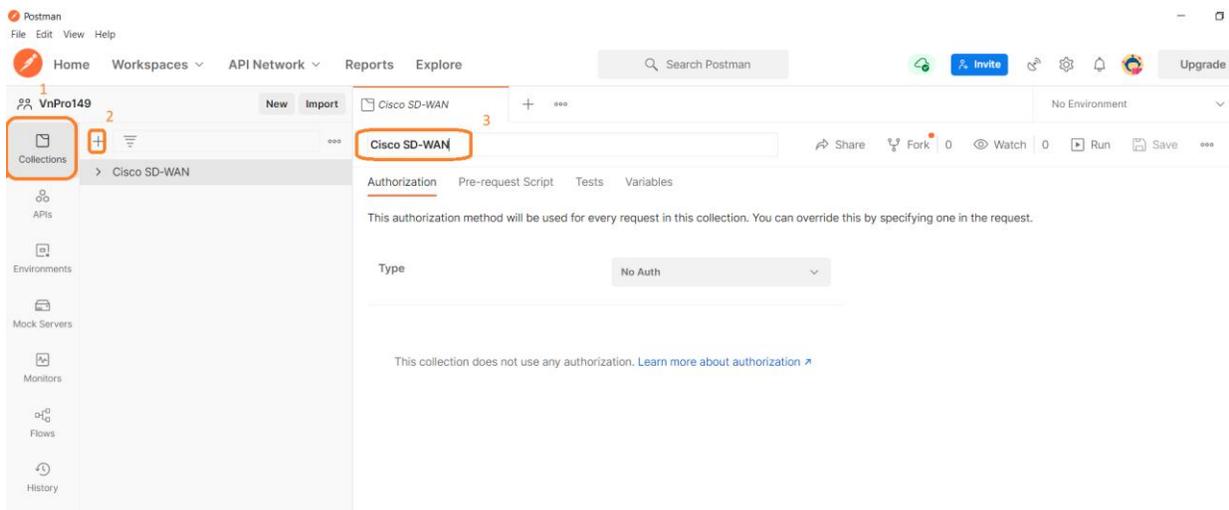


## b. Xác thực

Thông thường việc đầu tiên khi chúng ta tương tác với API đó là việc xác thực. Xác thực để đảm bảo rằng chỉ những người dùng có quyền mới có thể truy cập vào API. Chúng ta có thể sử dụng các tính năng RBAC để có thể giới hạn quyền truy cập tài nguyên.

Để hiểu rõ hơn về các xác thực của **CISCO SD-WAN REST API**, bạn có thể tham khảo thêm tài liệu trên trang chính thức của Cisco: <https://developer.cisco.com/sdwan/>

## Bước 1: Tạo collection

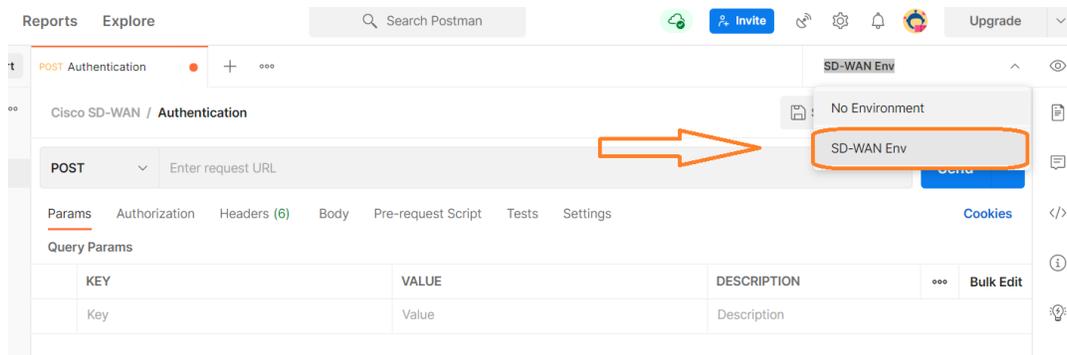


## Bước 2: Tạo một request để xác thực

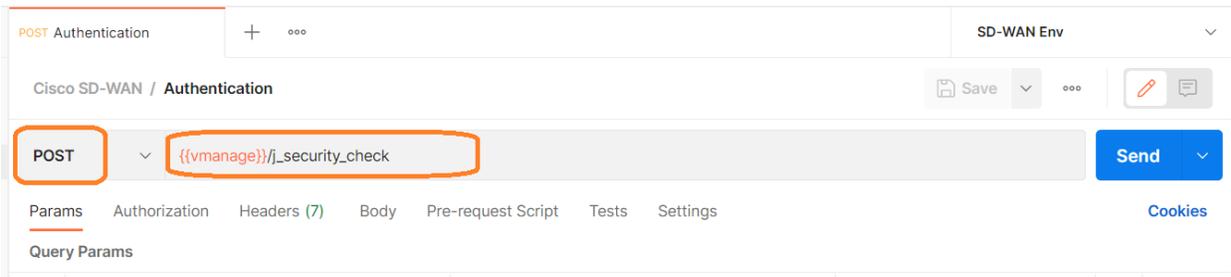
Bên trong Cisco-SD-WAN collection, tạo một API request đặt tên là **Authentication**.

- Authentication call sử dụng phương thức POST
- Để xác định địa chỉ của vManage chúng ta sử dụng biến môi trường `{{vmanage}}` đã tạo ở phần trước. Biến môi trường sẽ được thay thế bằng giá trị đã tạo trong môi trường khi gửi request.
- Địa chỉ mà chúng ta trở đến là `j_check_security`

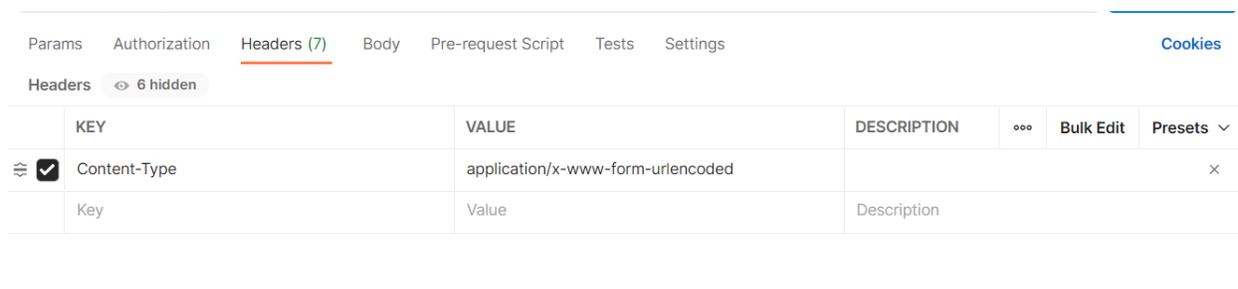
Chọn đúng môi trường đã tạo



Điền request URL



Dưới phần **Headers** tab, nhập thông tin **Content-Type**



Thông tin username và password được điền vào **Body** tab với key là **j\_username** và **j\_password**, phần value gọi lại biến trong môi trường

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> j_username	{{username}}			
<input checked="" type="checkbox"/> j_password	{{password}}			
Key	Value	Description		

Sau khi đã điền tất cả những tham số cần thiết để xác thực với vManage: địa chỉ vManage, phương thức, header và body. Nhấn nút **Send** và trả về kết quả như sau:

POST Authentication + ... SD-WAN Env

Cisco SD-WAN / Authentication Save ...

POST {{vmanage}}/j\_security\_check Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

Query Params

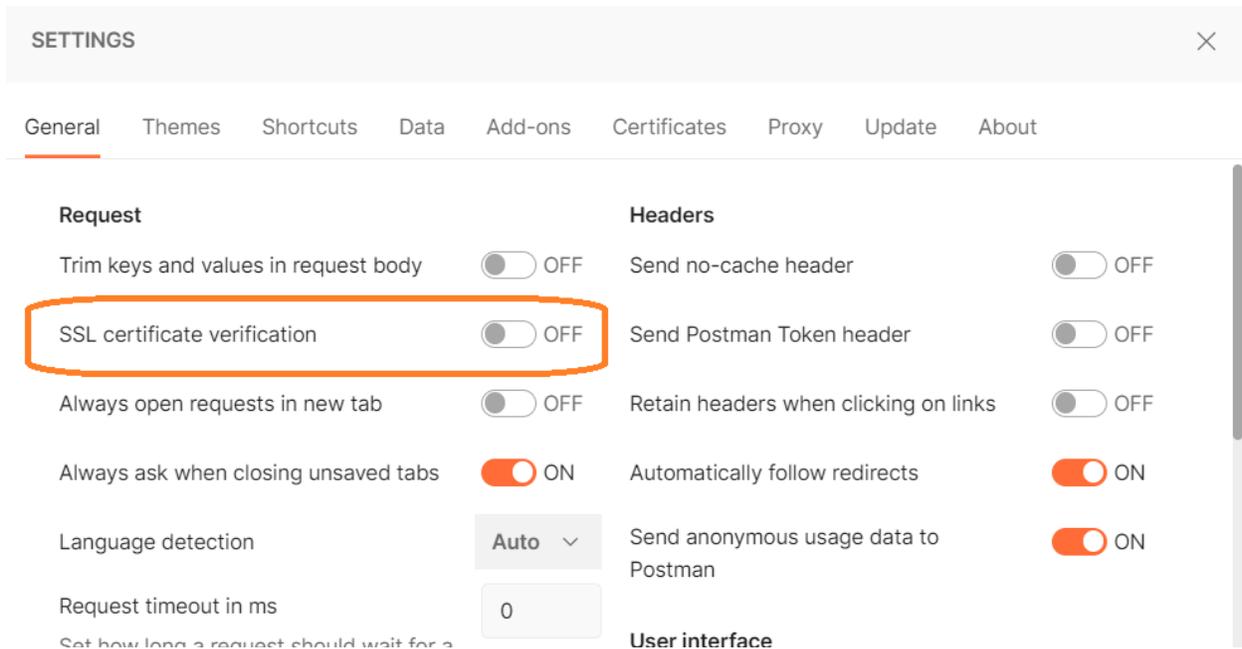
KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies (1) Headers (11) Test Results Status: 200 OK Time: 1289 ms Size: 444 B Save Response

Name	Value	Domain	Path	Expires	HttpOnly	Secure
JSESSIONID	8b2NHPqj3D...	sandbox-sdw...	/	Session	true	true

Phần **Body** sẽ không trả về bất kỳ thông tin nào và status phải là **200 OK**. Chú ý phần Cookie sẽ trả về **JSESSIONID** và giá trị bên trong. Thông tin **Cookie** đó sẽ được sử dụng để xác thực cho lần gọi API tiếp theo, cookie chỉ tồn tại trong một khoảng thời gian nhất định.

**Lưu ý:** Nếu bạn sử dụng SD-WAN vManage của cá nhân và chưa được cấu hình SSL thì bạn phải vào phân File > Setting > tắt chức năng SSL certificate verification



### Bước 3: Phòng Ngừa API CSRF

**CSRF** hay còn gọi là kỹ thuật tấn công “**Cross-site Request Forgery**“, nghĩa là kỹ thuật tấn công giả mạo chính chủ thể của nó. **CSRF** nói đến việc tấn công vào chứng thực request trên web thông qua việc sử dụng Cookies. Đây là nơi mà các hacker có khả năng sử dụng thủ thuật để tạo request mà bạn không hề biết.

Tính năng này được sử dụng để chống kỹ thuật tấn công giả mạo (CRSF) có thể xảy ra khi sử dụng Cisco SD-WAN REST APIs. Ở đây chúng ta sử dụng `csrf_token` để phòng chống CSRF, token này sẽ được đính kèm trong các lần gọi API và sẽ thay đổi liên tục trong phiên làm việc. Nếu token được sinh ra và token được gửi lên ko trùng nhau thì loại bỏ request.

Tạo một request mới để thực hiện lấy Token:

- Sử dụng phương thức **Get**
- Resource trở đến là `/dataservice/client/token`

The screenshot shows the Postman interface with a collection named 'Cisco SD-WAN'. A request is configured with the method 'GET' and the URL '[[vmanage]]/dataservice/client/token'. The response is displayed in the 'Body' tab, showing a JSON object with a 'token' field containing a long alphanumeric string.

KEY	VALUE	DESCRIPTION
Key	Value	Description

```
1 {  "token": "AD08526832B18D668F91CE5EF062BD7A2212D5DB10520B2AB9A29B678A8F434AD9E455B310145EA464C7F594BFBA017D44CB"}
```

### c. Lấy thông tin và trạng thái thiết bị SD-WAN

Sau khi đã xác thực thành công và có được **JSESSIONID** cookie, bây giờ bạn có thể truy xuất dữ liệu từ **Cisco SD-WAN REST API**.

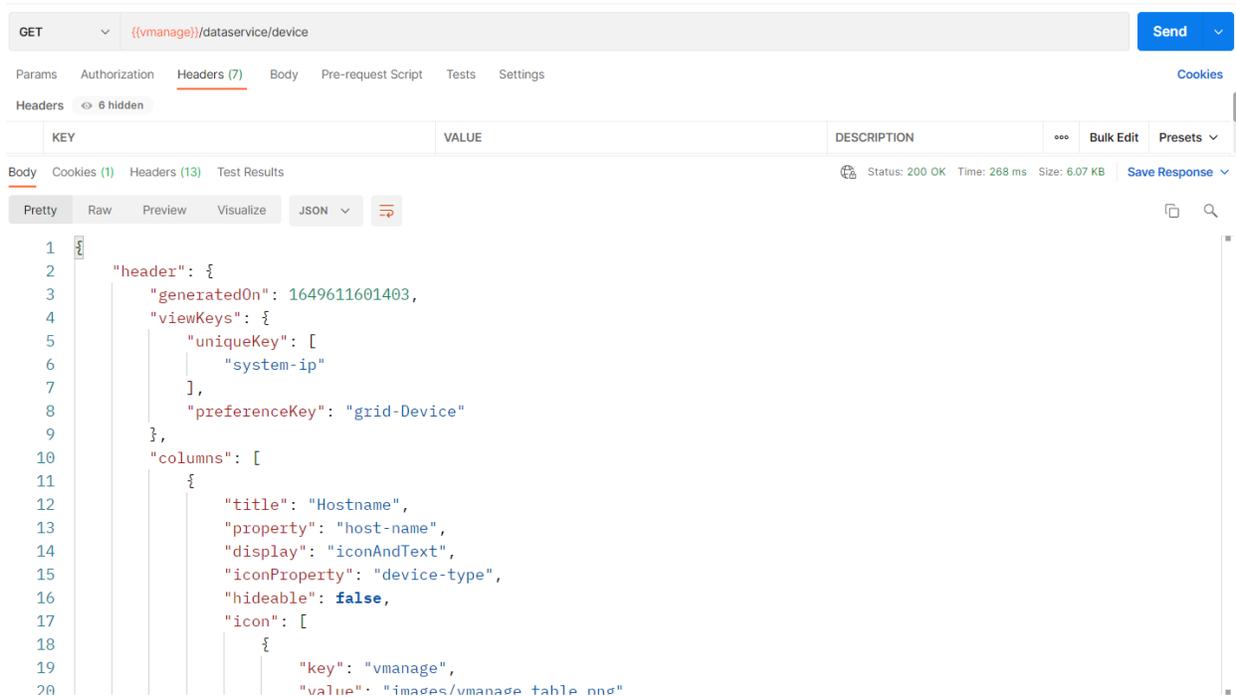
#### Bước 1: Danh sách thiết bị

Để lấy được danh sách thiết bị chúng ta sử dụng phương thức GET cùng với resource đến api endpoint là **/dataservice/device**. Dữ liệu sẽ được trả về với định dạng JSON cùng với danh sách của tất cả thiết bị có trong SD-WAN fabric:

The screenshot shows the Postman interface with a collection named 'Cisco SD-WAN'. A request is configured with the method 'GET' and the URL '[[vmanage]]/dataservice/device'. The 'Headers' tab is selected, showing a 'Content-Type' header set to 'application/x-www-form-urlencoded'.

KEY	VALUE	DESCRIP
<input checked="" type="checkbox"/> Content-Type	application/x-www-form-urlencoded	
Key	Value	Description

Sau khi nhấn **Send** sẽ trả về kết quả tương tự như hình:



The screenshot shows a REST client interface with the following details:

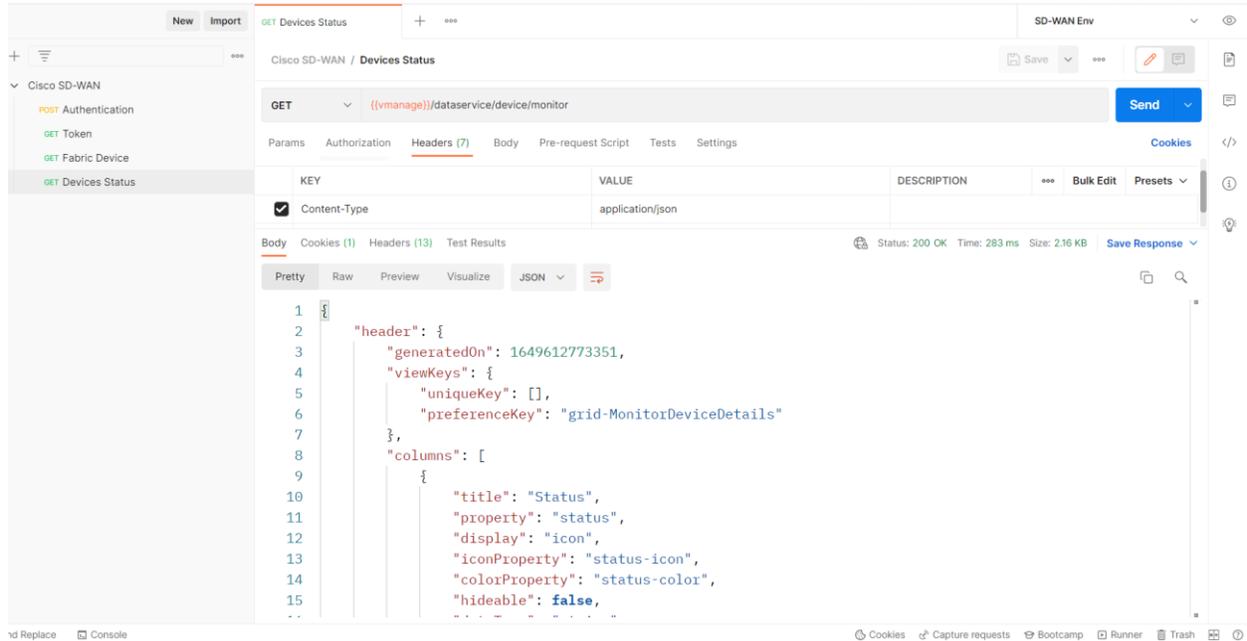
- Method: GET
- URL: {{vmanage}}/dataservice/device
- Status: 200 OK
- Time: 268 ms
- Size: 6.07 KB
- Response body (JSON):

```
1 {
2   "header": {
3     "generatedOn": 1649611601403,
4     "viewKeys": {
5       "uniqueKey": [
6         "system-ip"
7       ],
8     "preferenceKey": "grid-Device"
9   },
10  "columns": [
11    {
12      "title": "Hostname",
13      "property": "host-name",
14      "display": "iconAndText",
15      "iconProperty": "device-type",
16      "hideable": false,
17      "icon": [
18        {
19          "key": "vmanage",
20          "value": "images/vmanage_table.png"
21        }
22      ]
23    }
24  ]
25 }
```

Nếu bạn không nhận được kết quả như hình, kiểm tra lại status code đã trả về **200 OK**. Nguyên nhân có thể do **JSESSIONID** cookie đã hết hạn, nếu vậy bạn cần phải xác thực lại.

## Bước 2: Trạng thái thiết bị

Request tiếp theo được tạo trong collection được gọi để xem thông tin cụ thể trạng thái của tất cả thiết bị trong fabric. Sử dụng phương thức **GET** và trở đến resource **/data/device/monitor**.



The screenshot shows a REST client interface with the following details:

- Request Method: GET
- URL: ((vmanage))/dataservice/device/monitor
- Content-Type: application/json
- Status: 200 OK
- Time: 283 ms
- Size: 2.16 KB

```
1 {
2   "header": {
3     "generatedOn": 1649612773351,
4     "viewKeys": {
5       "uniqueKey": [],
6       "preferenceKey": "grid-MonitorDeviceDetails"
7     },
8     "columns": [
9       {
10        "title": "Status",
11        "property": "status",
12        "display": "icon",
13        "iconProperty": "status-icon",
14        "colorProperty": "status-color",
15        "hideable": false,
16        ...
17      }
18    ]
19  }
20 }
```

#### d. Lấy thông tin device counters và interface statistics

Nếu bạn muốn có thêm thông tin liên quan đến các kết nối của fabric, số lần thiết bị được khởi động lại và nhiều thông tin khác. Nếu bạn tìm kiếm trong **API Documentation**, bạn sẽ tìm thấy resource `/dataservice/device/counters`.

Kết quả trả về như sau:

The screenshot shows a REST client interface for the endpoint `GET {{vmanage}}/dataservice/device/counters`. The response is displayed in JSON format:

```
1 {
2   "header": {
3     "generatedOn": 1649613328983
4   },
5   "data": [
6     {
7       "system-ip": "10.10.1.5",
8       "number-vsmart-control-connections": 0,
9       "expectedControlConnections": 0,
10      "ompPeersUp": 0,
11      "ompPeersDown": 0,
12      "rebootCount": 3,
13      "crashCount": 0
14    },
15    {
16      "system-ip": "10.10.1.1",
```

Tiếp theo, lấy thông tin số liệu thống kê của các interface có trong SD-WAN fabric sử dụng resource `/dataservice/statistics/interface`.

The screenshot shows a REST client interface for the endpoint `GET https://{{vmanage}}:{{port}}/dataservice/statistics/interface`. The response is displayed in JSON format:

```
1 {
2   "header": {
3     "generatedOn": 1584780919293,
4     "viewKeys": {
5       "uniqueKey": [],
6       "preferenceKey": "grid-raw_interfacestatistics"
7     },
8     "columns": [
9       {
10        "title": "Entry_time",
11        "property": "entry_time",
12        "displayFormat": "DD MMM YYYY h:mm:ss A z",
13        "inputFormat": "unix-time",
14        "dataType": "date"
15      },
16      {
17        "title": "StatCycleTime",
18        "property": "statcycletime",
19        "displayFormat": "DD MMM YYYY h:mm:ss A z",
20        "inputFormat": "unix-time",
21        "dataType": "date"
```