

# 配置Postman以在vManage上執行API

## 目錄

[簡介](#)

[系統要求](#)

[背景資訊](#)

[配置Postman以執行API](#)

[步驟1.開啟Postman並建立新的HTTP請求。](#)

[步驟2.使用您的使用者名稱和密碼憑證進行身份驗證，以進行vManage。](#)

[步驟3.請求令牌](#)

[步驟4.繼續執行另一個API以進行vManage。](#)

[步驟5.關閉您的會話](#)

[在自動化環境中運行API呼叫](#)

[如何在變數中儲存令牌？](#)

[如何為新會話清除SESSIONID cookie？](#)

[如何使用收集運行器](#)

## 簡介

本文檔介紹如何使用Postman執行應用程式程式設計介面(API)。

## 系統要求

- 已安裝Postman
- 訪問vManage以及使用者名稱和密碼憑證

**注意：**如果您沒有Postman，請從<https://www.postman.com/downloads/>下載[它](#)

## 背景資訊

主要或最常用的HTTP動詞（或方法，它們被正確稱為）是POST、GET、PUT、PATCH和DELETE。

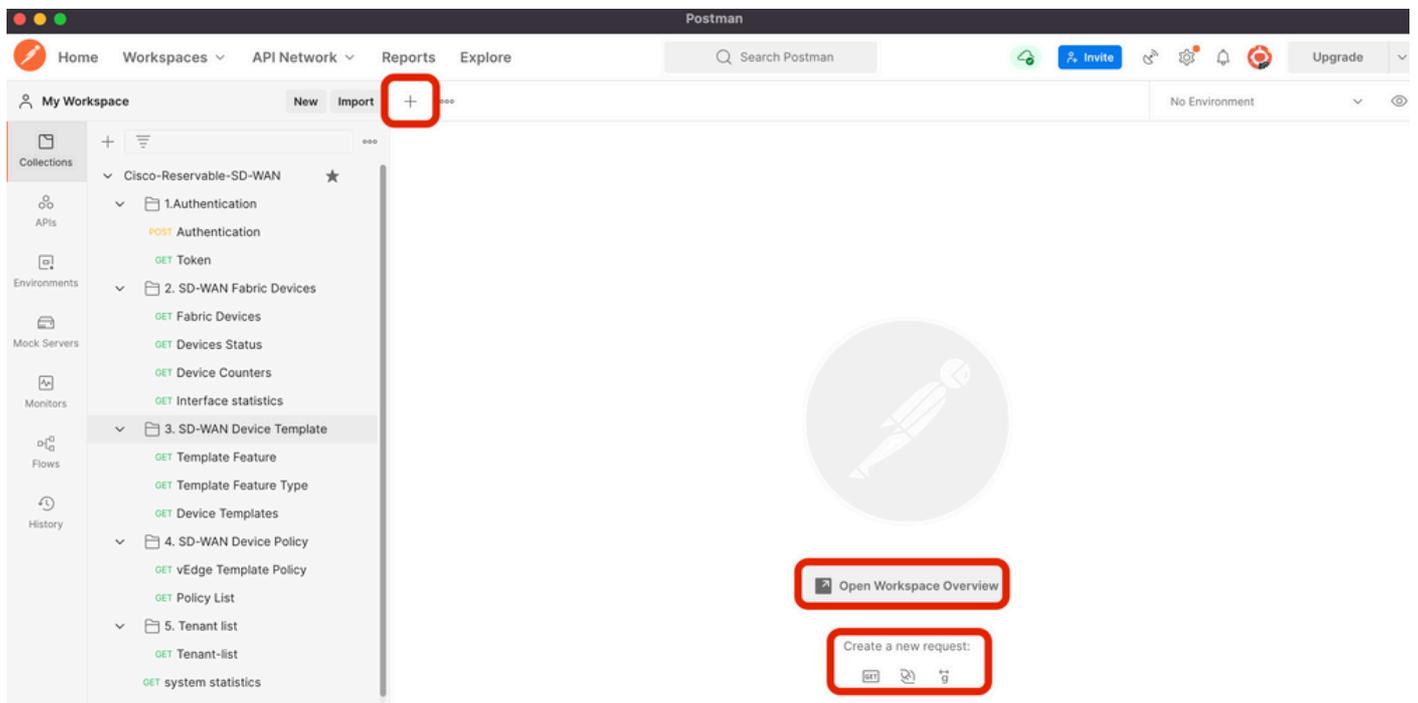
它們分別對應於建立、讀取、更新和刪除（或CRUD）操作。

也有一些其他的動詞，但使用頻率較低。在這些頻率較低的方法中，OPTIONS和HEAD的使用率高於其他方法。

## 配置Postman以執行API

### 步驟1.開啟Postman並建立新的HTTP請求。

如果按一下任何突出顯示的選項，則可以建立新的HTTP請求。



建立新的HTTP請求。

## 步驟2.使用您的使用者名稱和密碼憑證進行身份驗證，以進行vManage。

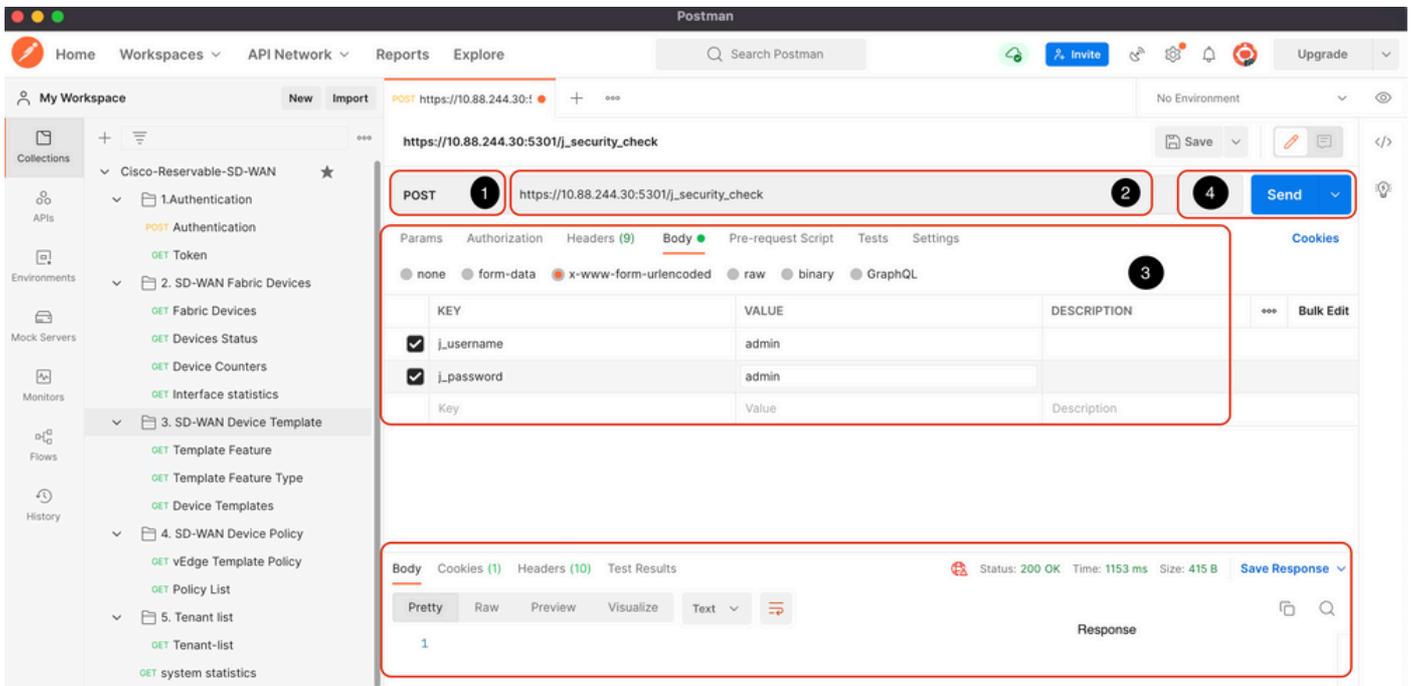
建立另一個HTTP請求。

1. 選擇POST作為HTTP謂詞。
2. 在POST旁邊新增[https://<vmanage-ip>/j\\_security\\_checknext](https://<vmanage-ip>/j_security_checknext)。
3. 按一下「Body」，然後分別將j\_username和j\_password引數及其值新增為KEY引數。
4. 按一下「Send」。

**注意：**在此示例中，vManage ip address ( vManage ip地址 ) 為10.88.244.30，埠為5301

**注意：**用於使用者名稱和密碼值，我們使用admin。

在Postman中填寫引數。



vManage authentication.

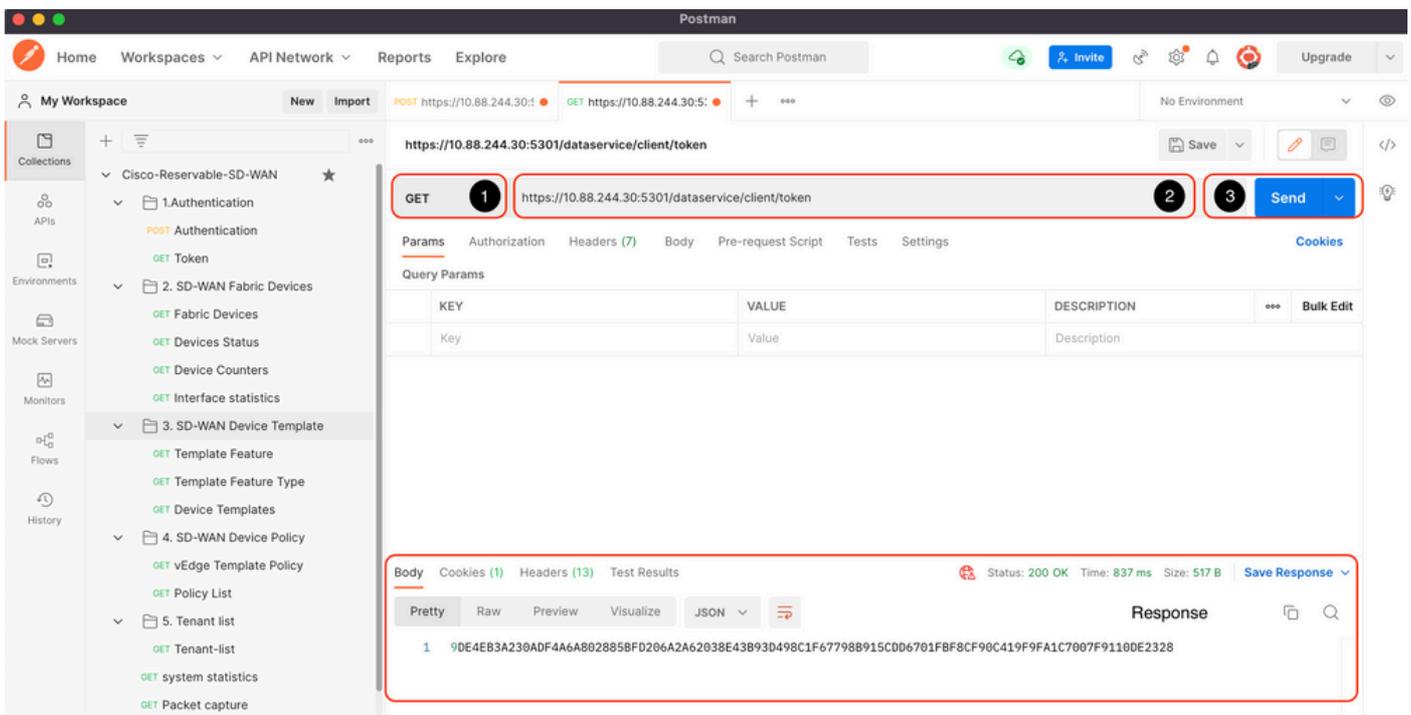
注意：此API呼叫的響應必須為空

### 步驟3.請求權杖

1. 選擇GET作為HTTP謂詞。
2. 在GET `https://<vmanage-ip>/dataservice/client/token`旁邊新增API呼叫詳細資訊
3. 按一下「Send」

註：自vManage 19.2.1版起，成功登入的使用者必須通過API呼叫為每個POST/PUT/DELETE操作傳送X-XSRG-TOKEN或CSRF令牌。

執行API呼叫後，將在正文中獲取響應字串。儲存該字串。圖中所示為Postman輸出示例。



**警告：**如果您沒有獲得如圖所示的令牌，請重複此步驟。

## 步驟4.繼續執行另一個API以進行vManage。

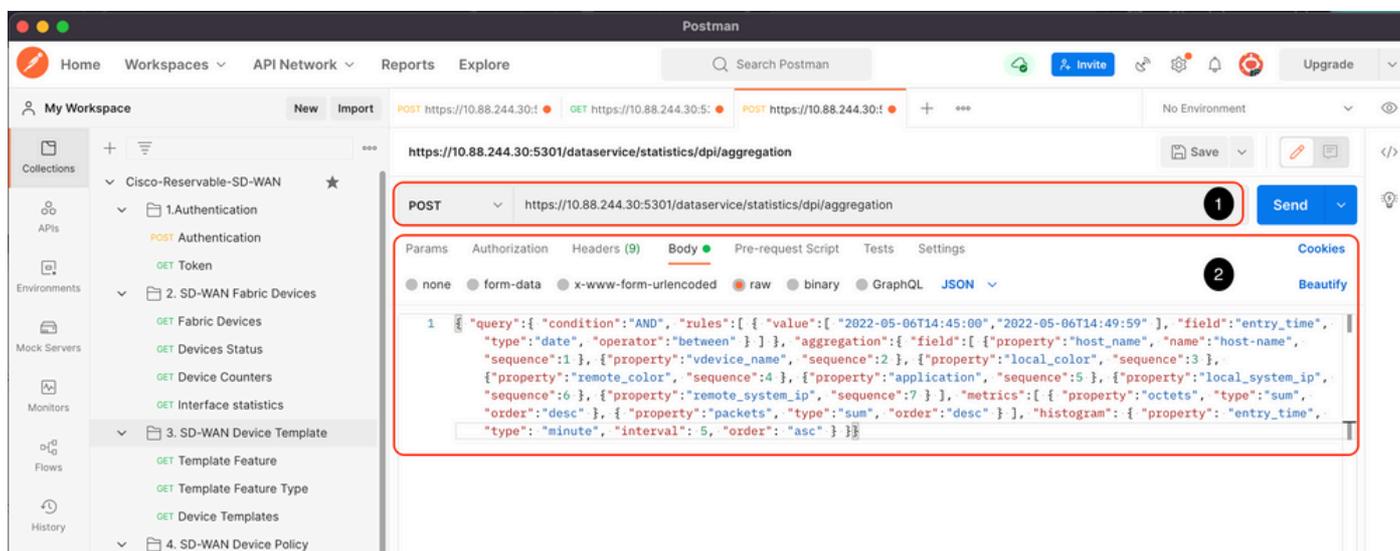
此範例包含一個POST請求

1. 選擇要執行的API呼叫，在我們的情況下為<https://dataservice/statistics/dpi/aggregation>

提示：如果您希望探索其他API呼叫，請轉到vManage url <https://vmanage-ip:port/apidocs>

2. 收集您的API呼叫正文。

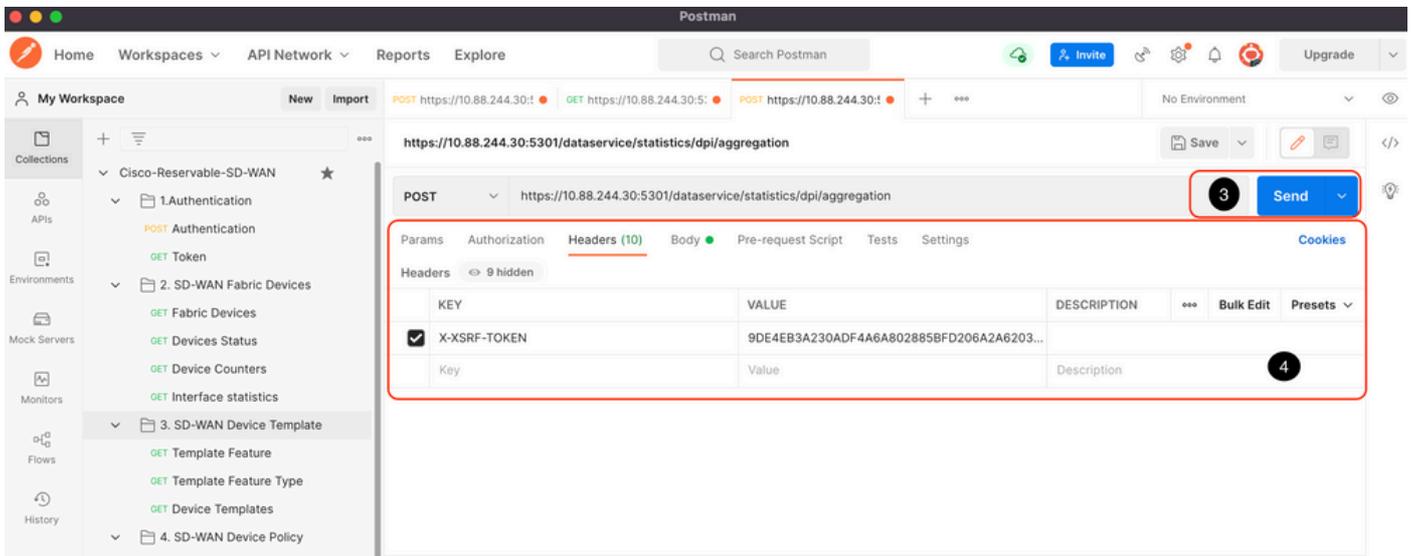
**注意：**此API呼叫包含JSON格式的正文



3. 按一下Header，然後將字串X-XSRF-TOKEN新增為Key作為值。

4. 按一下「傳送」。

顯示的影象顯示了必須如何顯示API呼叫。



DPI聚合API呼叫。

## 步驟5.關閉您的會話

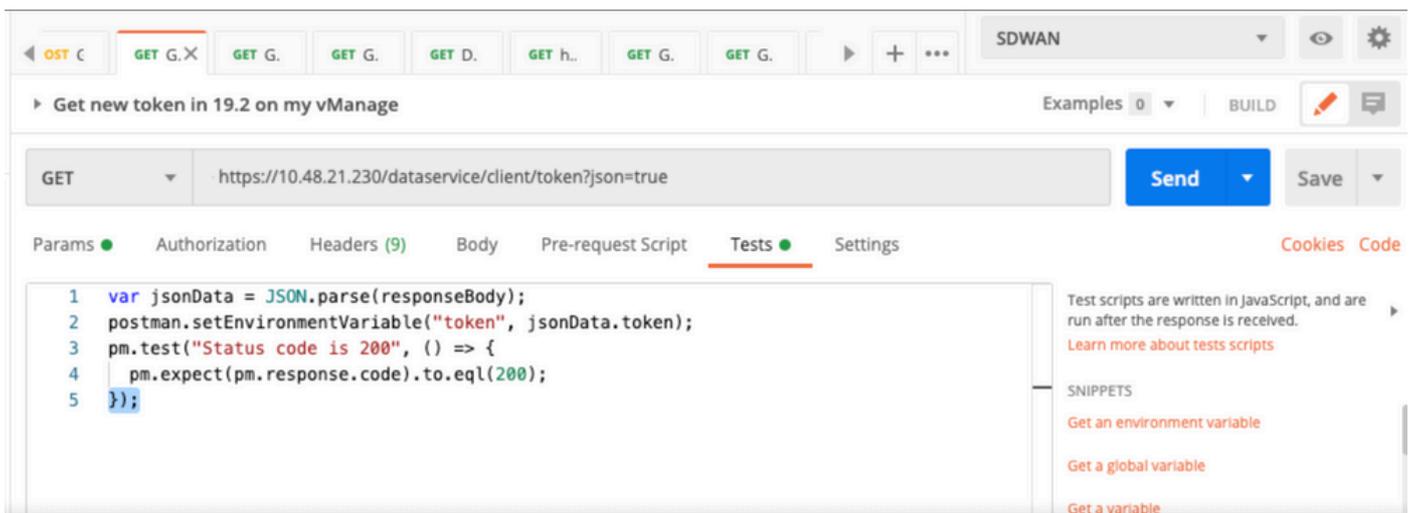
從vManage和/或裝置檢索到所需的所有資訊後，即可釋放vManage的資源，消除惡意使用者使用您的會話的可能性。

## 在自動化環境中運行API呼叫

儲存Cookie和要在後續API呼叫中使用的變數

### 如何在變數中儲存令牌？

將令牌儲存在變數中，以便以後重複使用。



將令牌儲存在變數中

當我們以JSON格式請求令牌時，請將其儲存。使用**測試**頁籤並貼上顯示的行。

```
var jsonData = JSON.parse(responseBody);
postman.setEnvironmentVariable("token", jsonData.token);
```

之後，任何API呼叫都可以使用令牌變數。

Key	Value	Description
Host	<calculated when request is sent>	
User-Agent	PostmanRuntime/7.26.3	
Accept	*/*	
Accept-Encoding	gzip, deflate, br	
Connection	keep-alive	
X-XSRF-TOKEN	{{token}}	
Content-Type	application/json	

使用令牌變數

## 如何為新會話清除SESSIONID cookie?

當您執行API呼叫以退出時，請使用JSESSIONID。

我們不能像在早期版本中那樣使用任何基本身份驗證。相反，我們只提供憑證並將ID儲存在Cookie中。在此之前，我們可以使用預測試來清除所有或特定的cookie。

```

1  const jar = pm.cookies.jar();
2
3  jar.clear(pm.request.url, function (error) {
4    // error - <Error>
5  });

```

清除Cookie

這是通過Pre-request指令碼中的代碼實現的。

## 如何使用收集運行器

現在我們有一些環境，可以在其中運行會話並儲存特定於每個會話的資料，因此您可以使用Collection Runner運行一系列呼叫。

選擇要重複的事件的順序，選擇重複計數，以便Postman可以執行API呼叫，即所選取的每次運行具

有結果的次數。

The screenshot shows the Postman interface. On the left, there is a 'Choose a collection or folder' section with a search bar and a list of API requests. Below this, there are settings for 'Environment' (SDWAN), 'Iterations' (5), 'Delay' (0 ms), and 'Data' (Select File). There are also checkboxes for 'Save responses', 'Keep variable values' (checked), 'Run collection without using stored cookies', and 'Save cookies after collection run'. A 'Run Viptela' button is at the bottom. On the right, the 'RUN ORDER' section shows a list of requests with checkboxes and their methods (POST, GET, PUT). The first four requests are checked and have their methods highlighted in red.

收集運行器

從呼叫的「庫」中，將它們按一定的順序排列，以獲得要執行的特定流/命令。

輸入結果檢查您得到200 OK或其他值作為響應，並將其視為通過或失敗。

The screenshot shows the Postman interface for a specific request. The request is a GET request to 'https://10.48.21.230/dataservice/client/token?json=true'. The 'Tests' tab is selected, and the test script is as follows:

```
1 var jsonData = JSON.parse(responseBody);
2 postman.setEnvironmentVariable("token", jsonData.token);
3 pm.test("Status code is 200", () => {
4   pm.expect(pm.response.code).to.eql(200);
5 });
```

The 'Test Results' section shows the status as '200 OK', 'Time: 67 ms', and 'Size: 550 B'. The 'Body' tab is selected, and the response is shown in JSON format:

```
1 {
2   "token": "23AE920117579F0EF9D470C2DE837A74C292D6A5929E098E06AB6358D399A61BD99B23D17D836D36EE0BAF764E1B10D52059"
3 }
```

檢查響應代碼

```
pm.test("Status code is 200", () => {  
  pm.expect(pm.response.code).to.eql(200);  
});
```

這樣我們就能看到跑步的通過或失敗。

Collection Runner Run Results My Workspace Run In Command Line Docs

20 PASSED 0 FAILED Viptela SDWAN just now Run Summary Export Results Retry New

Iteration 1

- POST Get JSESSIONID in newer release(s) https://10.48.21.230/j\_se... Viptela / Get JSESSIONID in newer ... Status code is 200
- GET Get new token in 19.2 on my vManage https://10.48.21.230/dat... Viptela / Get new token in 19.2 on... 200 OK 53 ms 550 B Status code is 200
- GET Get server info with in-correct token https://10.48.21.230/dat... Viptela / Get server info with in-co... 403 Forbidden 56 ms 583 B Status code is 403
- GET Get server information 19.2 lab vManage with correct token https://10.48.21.230/dat... Viptela / Get server information 1... 200 OK 49 ms 486 B Status code is 200

Iteration 2

- POST Get JSESSIONID in newer release(s) https://10.48.21.230/j\_se... Viptela / Get JSESSIONID in newer ... Status code is 200
- GET Get new token in 19.2 on my vManage https://10.48.21.230/dat... Viptela / Get new token in 19.2 on... 200 OK 48 ms 550 B Status code is 200
- GET Get server info with in-correct token https://10.48.21.230/dat... Viptela / Get server info with in-co... 403 Forbidden 49 ms 583 B Status code is 403

Console

自動運行

## 關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。