

使用帶Python的Catalyst Center API

目錄

[簡介](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[設定](#)

[概觀](#)

[模組](#)

[生成令牌](#)

[測試API](#)

[具有標頭引數的API](#)

[具有查詢引數的API](#)

簡介

本文檔介紹如何使用Python在Cisco Catalyst Center上提供的不同API。

必要條件

需求

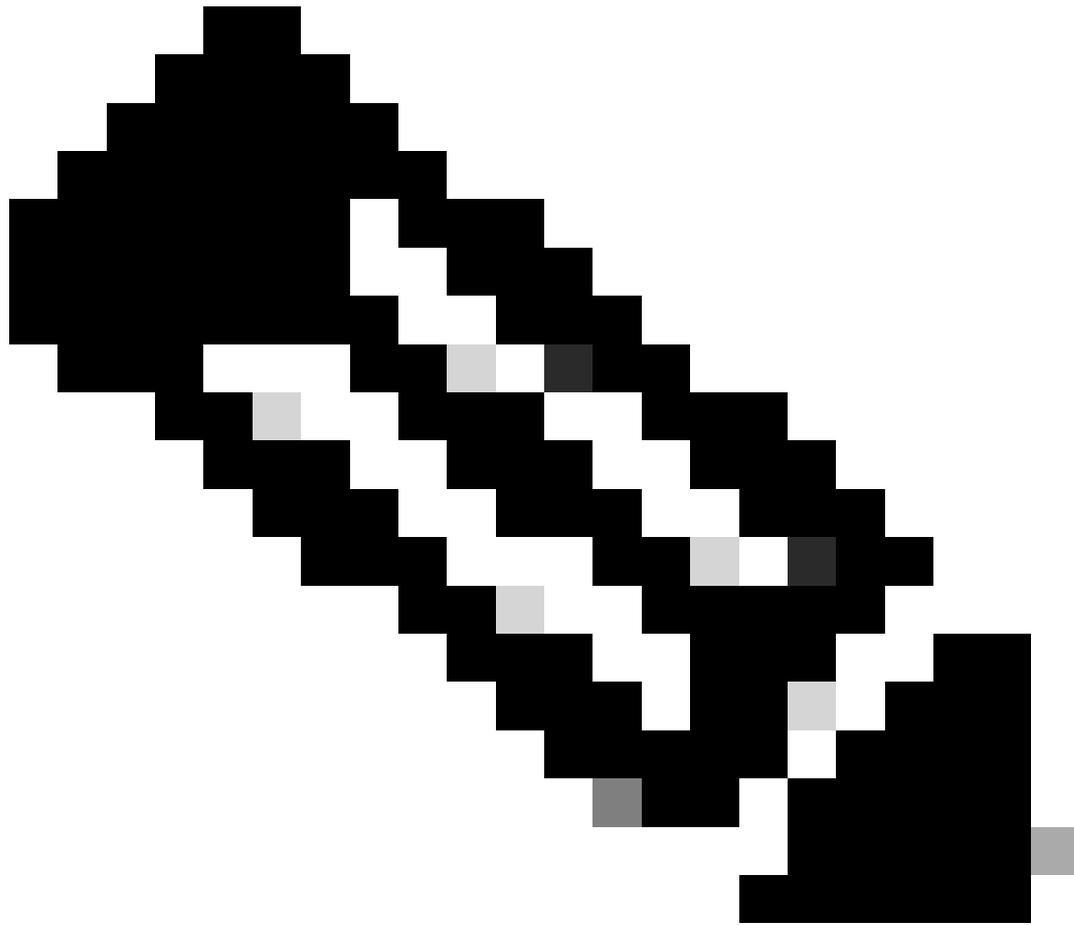
基本知識：

- Cisco Catalyst Center
- API
- Python

採用元件

- Cisco Catalyst中心2.3.5.x
- Python 3.x.x

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。



注意：思科技術支援中心(TAC)不提供對Python的技術支援。如果您遇到Python問題，請與Python支援部門聯絡以獲取技術支援。

設定

概觀

Cisco Catalyst Center提供了許多API。要驗證可以使用哪些API，請在Catalyst Center上導航到平台>開發人員工具包> API。

Check out our API capabilities and try them out for yourself

Explore our developer documentation or test different APIs in your network environment to build, connect, and leverage rich capabilities of Cisco DNA Center.



🔍 Search

Authentication

Cisco DNA Center System

Health and Performance

Licenses

Platform

User and Roles

Connectivity

Fabric Wireless

SDA

Wireless

Ecosystem Integrations

ITSM

Event Management

Integrations

🔍 Search API

Authentication

Authentication APIs provide an authorized token for accessing any REST API.

***Prerequisite*:** Add the request header 'x-auth-token' with the generated authorized token to get a successful API response.

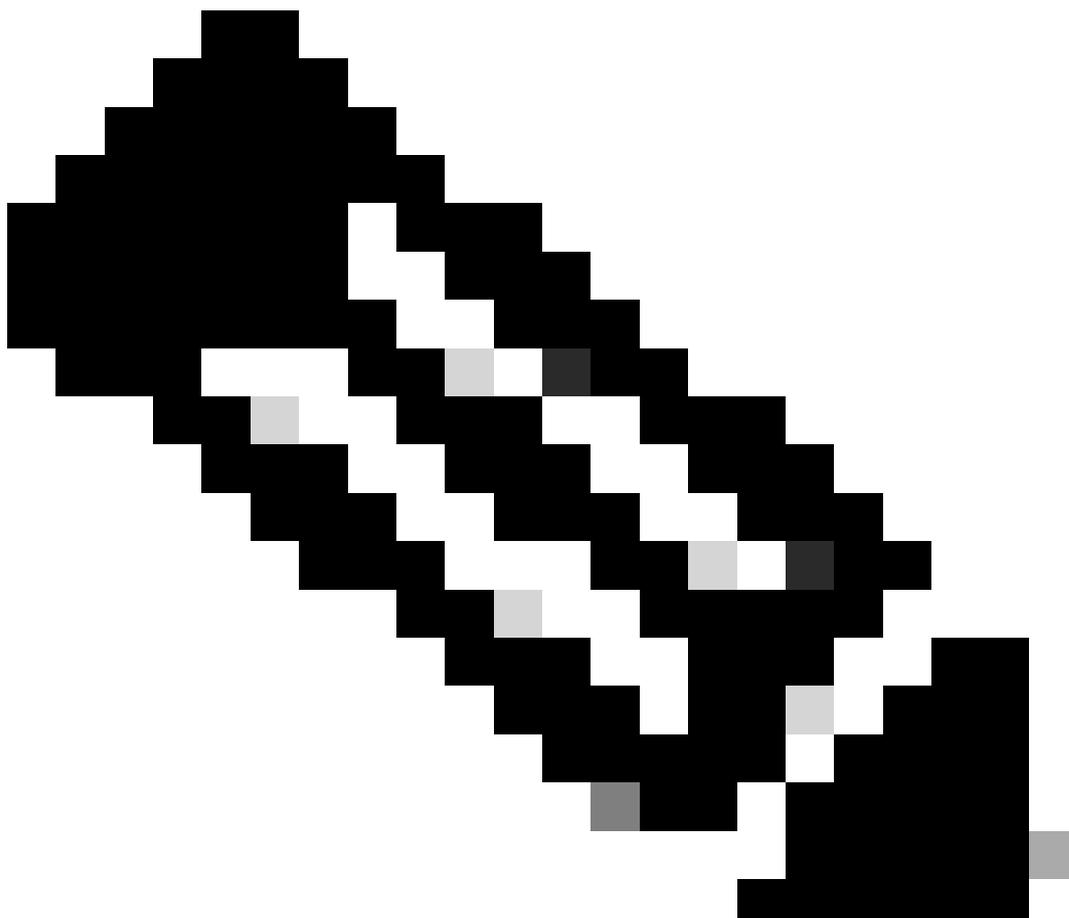
Method	Name	Description	URL	Actions
POST	importCertificate	This method is used to upload a certificate	/certificate	— ▾
POST	importCertificateP12	This method is used to upload a PKCS#12 file	/certificate-p12	— ▾
POST	Authentication API	API to obtain an access token, which remains valid for 1 hour. The token obtained using this API is required to be set as value to the X-Auth-Token HTTP...	/auth/token	— ▾

Catalyst Center API頁面

每個API都有其自己的目的，具體取決於需要在Catalyst Center上執行的資訊或操作。要使API正常工作，先決條件是，必須使用令牌向Catalyst Center正確驗證並獲得成功的API響應。令牌相應地標識REST呼叫方的許可權。

確定構成API的元件也很重要，這些元件如下：

- URL：提供特定資源存取的端點。
- 方法：所有API都必須包含方法。它定義了客戶端要對特定終端執行的操作/操作。範例：POST、GET、PUT、DELETE。
- 標題：以鍵值對格式提供有關請求的其他資訊。例如，授權報頭提供使用憑證的身份驗證方法。
- 引數：使用API向端點提供特定指令的變數。引數可以是端點URL的一部分。
- 負載：在API呼叫期間需要傳送到終結點的資料。



注意：有關Catalyst Center上提供的每個API的詳細資訊，請參閱[API參考指南](#)。

模組

使用的Python模組：

- 請求：此模組允許將HTTP/1.1請求傳送到特定URL。有關模組的詳細資訊，請參閱[請求模組指南](#)。
- base64：提供編碼和解碼功能。有關模組的詳細資訊，請參閱[base64模組指南](#)。
- json：此模組允許從API響應獲取特定資料。有關模組的詳細資訊，請參閱[json模組指南](#)。

註：有關如何安裝Python模組的詳情，請參閱[安裝Python模組](#)文檔。

生成令牌

必須使用身份驗證API呼叫API來生成新令牌。

驗證API：

```
POST https://<CatalystCenterIP>/dna/system/api/v1/auth/token
```

必須說明的是，生成的令牌有效期為1小時。1小時後，必須使用上面提到的同一API生成新令牌。

在新的Python檔案中，導入模組(請求、base64和json)，然後建立四個變數：

```
import requests
import base64
import json

user = 'user'    # User to login to Catalyst Center
password = 'password'    # Password to login to Catalyst Center
token = ''      # Variable to store the token string
authorizationBase64 = ''    # Variable that stores Base64 encoded string of "username:password"
```

身份驗證API支援將基本身份驗證作為報頭中的授權令牌。基本身份驗證是一種可用於對終端進行身份驗證的方法，該方法提供以冒號分隔的使用者名稱和口令(username : password)。這兩個值都採用base64編碼，並且終端對登入憑證進行解碼，並檢查使用者是否可以訪問。

要為我們的使用者名稱和口令建立Base64連線的字串，請使用base64模組。為此，使用b64encode功能。

```
byte_string = (f'{user}:{password}').encode("ascii")
authorizationBase64 = base64.b64encode(byte_string).decode()
```

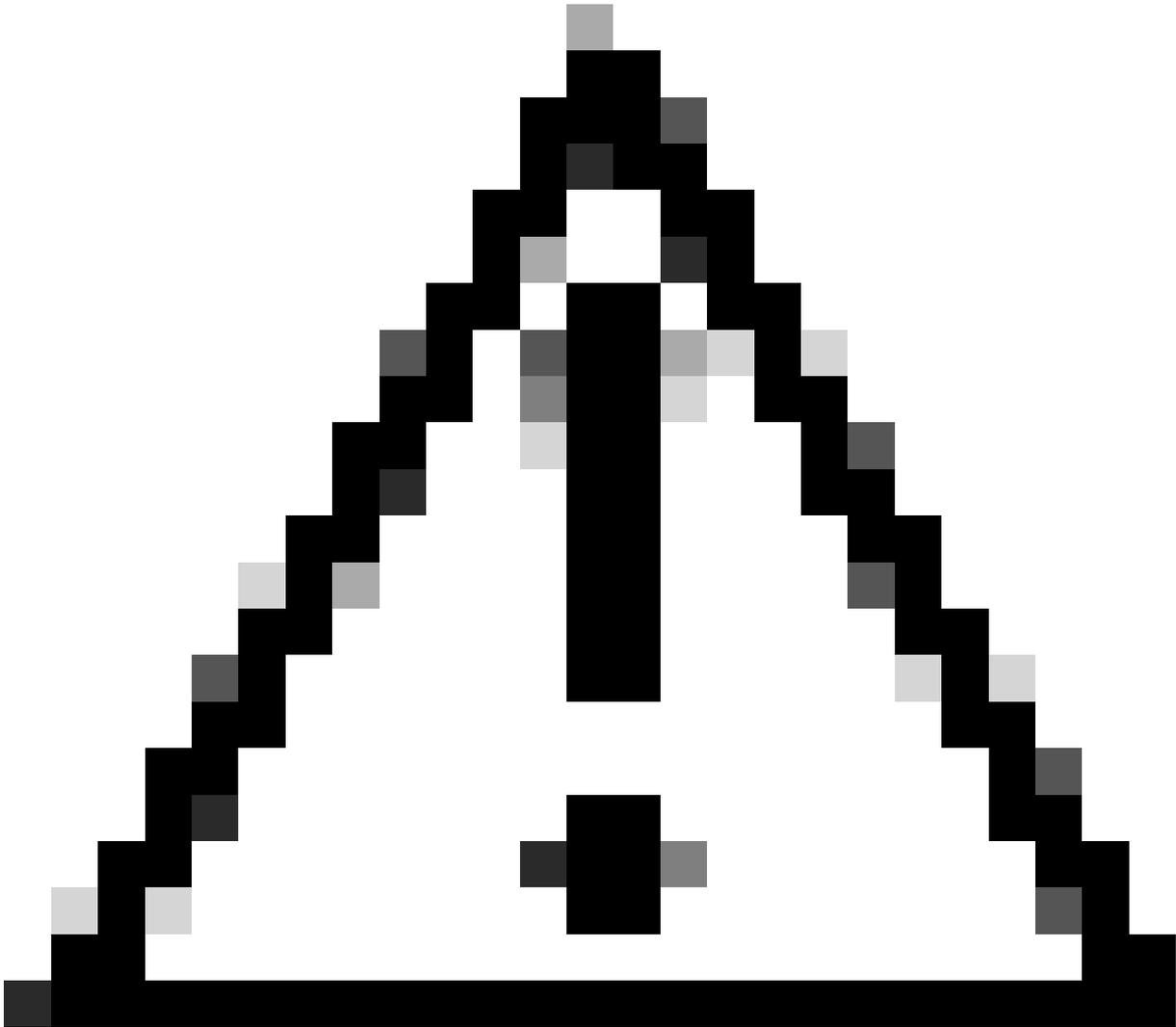
從上面的代碼中，使用「.encode(「ascii」)」函式建立了byte_string變數。這是因為base64.b64encode函式需要類似於位元組的對象。另請注意，user和password變數用於保留字串格式user : password。最後，使用使用者和密碼建立了一個base64編碼的位元組字串。使用'decode()'方法，值會轉換為str物件。

要驗證它，可以列印authorizationBase64變數的值：

```
print(authorizationBase64)
```

輸出範例：

```
am9yZ2QhbDI6Sm9yZ2VhbDxXxXx
```



注意：base64不是加密演算法。不得用於安全目的。身份驗證API還支援將AES金鑰加密作為報頭中的授權令牌，從而提供更高的安全性。

現在，已使用用於向Catalyst Center進行身份驗證的使用者和密碼建立了一個base64編碼的字串，現在應使用模組請求繼續進行API身份驗證 API呼叫。此外，名為request的函式可用於獲取包含請求文本的響應對象。

方法的語法：

```
requests.request("method", "url", **kwargs)
```

**kwargs表示傳遞到請求中的任何引數，例如cookie、使用者代理、負載、報頭等。

驗證API指定方法為POST，URL為「/dna/system/api/v1/auth/token」，且必須在Header中指定基本驗證。

建立這些變數是為了將其用於request()函式。

```
url = https://<CatalystCenterIP>/api/system/v1/auth/token
headers = {
    'content-type': "application/json",
    'Authorization': 'Basic ' + authorizationBase64
}
```

對於headers變數，指定了兩個引數。第一個是content-type，它指定傳送到端點的資源的媒體型別（這有助於端點準確分析和處理資料）。第二個是Authorization，在本例中，變數authorizationBase64（儲存我們的base64隨附字串）被作為引數傳送到Catalyst Center進行身份驗證。

現在，繼續使用request()函式執行API呼叫。下一個程式碼會顯示函式的語法：

```
response = requests.request("POST", url, headers=headers)
```

建立response變數是為了儲存我們發出的API呼叫的資料。

若要列印取得的回應，請將print函式與回應變數中的text()方法一起使用。text()方法使用從Catalyst Center接收的響應生成str對象。

```
print(response.text)
```

輸出範例：

```
{"Token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpzZW50L3N1bWU6IiwiaWF0IjoiYXZ5bWw2vMPubU0JN1q"}
!--- Output is suppressed
```

注意：如果Catalyst Center使用的是自簽名證書，則API請求可能會失敗，並顯示下一個錯誤：

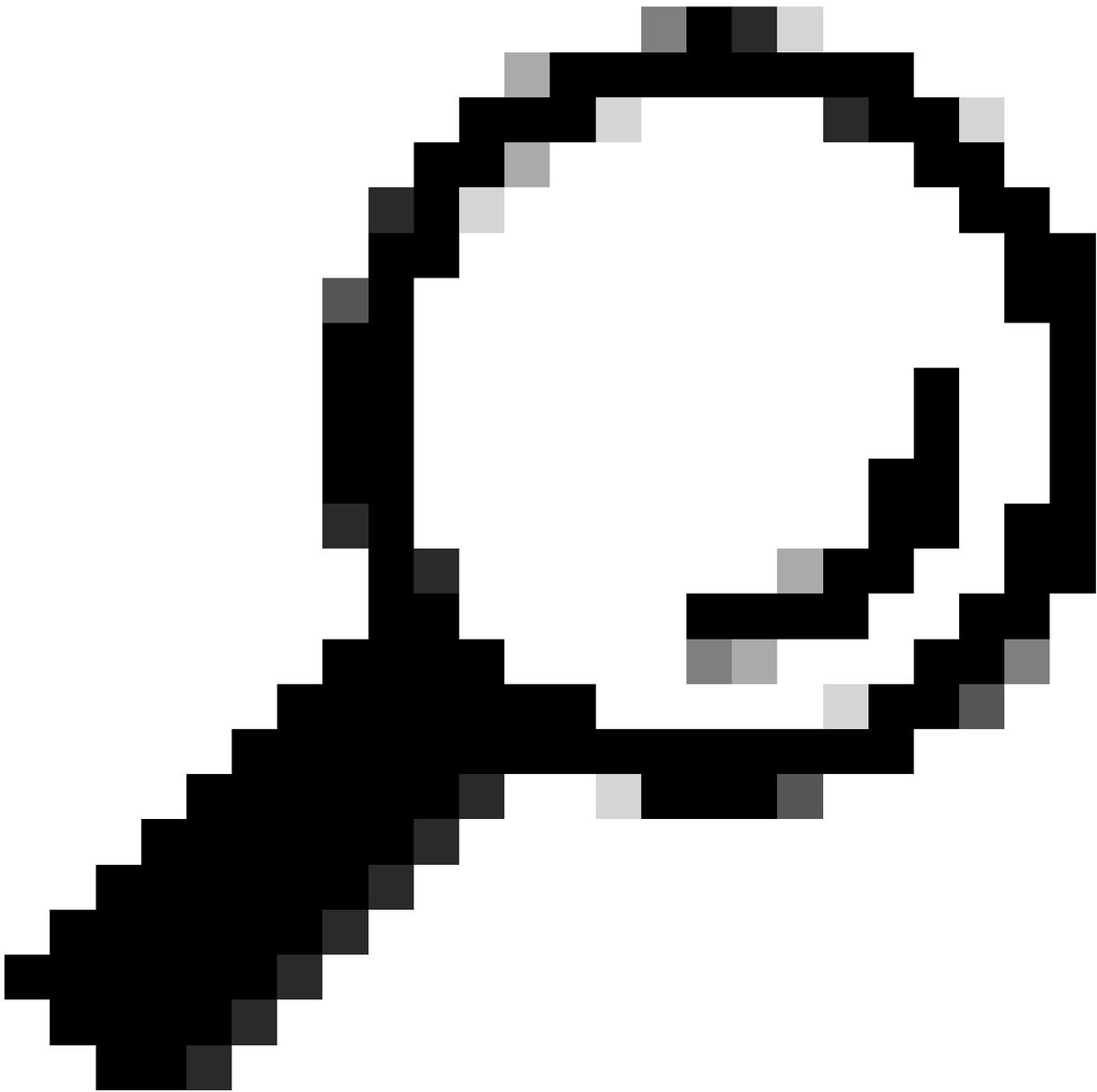
```
requests.exceptions.SSLError: HTTPSConnectionPool(host='X.X.X.X', port=443): Max retries exceeded
```

要解決此問題，需要將verify引數作為False增加到請求函式中。這會忽略從端點(Catalyst Center)驗證SSL憑證。

```
response = requests.request("POST", url, headers=headers, verify=False)
```

從從API身份驗證呼叫收到的響應中可以看到，其結構類似於Python中的詞典，但其實是str對象。

若要驗證物件的型別，請使用type()函式。



提示：由於每個生成的令牌預設在1小時內到期，因此可以建立一個Python方法（包含用於生成令牌的代碼），並在每次令牌到期時呼叫該方法，而無需僅透過呼叫建立的方法運行整個程式。

測試API

現在，令牌已成功分配給令牌變數，您可以使用可用的Catalyst Center API。

在這種情況下，將測試Cisco DNA中心節點配置摘要 API。

Cisco DNA中心節點配置摘要

GET <https://<CatalystCenterIP>/dna/intent/api/v1/nodes-config>

此API提供有關Catalyst Center當前配置的詳細資訊，例如配置的NTP伺服器、節點名稱、集群內鏈路、LACP模式等。

Cisco DNA Center Nodes Configuration Summary API指定，在此例中，使用的方法為GET，URL為「/dna/intent/api/v1/nodes-config」，並且，由於已提取令牌字串並將其分配給令牌變數，因此，這次令牌作為令牌在API呼叫報頭中的變數傳遞為「X-Auth-Token」：，後面跟該令牌。

這會驗證每次執行的API呼叫對Catalyst Center的請求。請記住，每個令牌持續1小時。1小時後，必須生成新令牌才能繼續向Catalyst Center進行API呼叫。

繼續建立變數以測試API：

```
nodeInfo_url = "https://<CatalystCenterIP>/dna/intent/api/v1/nodes-config"
nodeInfo_headers = {
    'X-Auth-Token': token
}

nodeInfoResponse = requests.request("GET", nodeInfo_url, headers=nodeInfo_headers)
```

建立nodeInfo_url變數是為了儲存API的URL。nodeInfo_headers變數會儲存API的標頭。在這種情況下，'X-Auth-Token：'和token變數被作為引數傳遞以成功向Catalyst Center驗證請求。最後，nodeInfoResponse變數儲存API的響應。

若要驗證收到的回應，您可以使用print()函式。

輸出範例：

```
{"response": {"nodes": [{"name": "Catalyst Center", "id": "ea5dbec1-fbb6-4339-9242-7694eb1cXxXx", "netw
!--- Output is suppressed
```

注意：如果Catalyst Center中使用的是自簽名證書，則API請求可能會失敗，並顯示下一個錯誤：

```
requests.exceptions.SSLError: HTTPSConnectionPool(host='X.X.X.X', port=443): Max retries exceeded
```

要解決此問題，需要將verify引數作為False增加到請求中。這會抑制從端點(Catalyst Center)驗證SSL憑證。

```
nodeInfoResponse = requests.request("GET", nodeInfo_url, headers=nodeInfo_headers, verify=False)
```

從API收到的響應可能難以讀取。使用json()模組，可以用更易讀的字串形式列印響應。首先，必須使用json.loads()函式後跟json.dumps()函式將API響應載入到JSON對象：

```
jsonFormat = (json.loads(nodeInfoResponse.text)) # Creating a JSON object from the string received from
print(json.dumps(jsonFormat, indent=1)) # Printing the response in a more readable string using the dump
```

json.dumps：此函式返回JSON對象，該對象作為JSON格式化字串中的引數。

縮排：此引數定義JSON格式字串的縮排級別。

輸出範例：

```
{
  "response": {
    "nodes": [
      {
        "name": "X.X.X.X",
        "id": "ea5dbec1-fbb6-4339-9242-7694eb1xXxX",
        "network": [
          {
            "slave": [
              "enp9s0"
            ],
            "lACP_supported": true,
            "intra_cluster_link": false,
!--- Output is suppressed
```

具有標頭引數的API

有些API需要在報頭中傳送一些引數才能按預期工作。在這種情況下，將測試Get Client Conversation Details API。

```
GET https://<CatalystCenterIP>/dna/intent/api/v1/client-enrichment-details
```

要驗證API需要哪些報頭引數才能按預期工作，請導航到平台>開發人員工具包> API >獲取客戶端增強詳細資訊，然後按一下API的名稱。此時將打開一個新窗口，並在Parameters選項下顯示API正常工作所需的標頭引數。

Get Client Enrichment Details



GET

https://10.88.244.133/dna/intent/api/v1/client-enrichment-details

Enriches a given network End User context (a network user-id or end user's device Mac Address) with details about the user, the devices that the user is connected to and the assurance issues that the user is impacted by

[Cisco DevNet API Guide](#)

TAGS

Client Enrichment

Network Event

Parameters

Responses

Policies

Code Preview

Request Header Parameters

Name	Description	DataType	Required	Default Value
entity_type	Client enrichment details can be fetched based on either User ID or Client MAC address. This parameter value must either be network_user_id/mac_address	string	Yes	
entity_value	Contains the actual value for the entity type that has been defined	string	Yes	
issueCategory	The category of the DNA event based on which the underlying issues need to be fetched	string	No	

在這種情況下，對於entity_type引數，根據說明，值可以是network_user_id或mac_address，並且entity_value引數必須包含已定義的實體型別的值。

要繼續執行該操作，需要定義兩個新變數，entity_type和entity_value及其對應值：

```
entity_type = 'mac_address' #This value could be either 'network_user_id' or 'mac_address'.
entity_value = 'e4:5f:02:ff:xx:xx' #Depending of the 'entity_type' used, need to add the correspondi
```

也會建立新變數來執行API呼叫。API呼叫的URL儲存在userEnfuration_url變數中。標頭儲存在userEnfurationHeaders變數中。收到的響應儲存在userEnfinanceResponse變數中。

```
userEnrichment_url = "https://<CatalystCenterIP>/dna/intent/api/v1/user-enrichment-details"
```

```
userEnrichmentHeaders = {
  'X-Auth-Token': token,
  'entity_type': entity_type,
  'entity_value': entity_value,
}
```

```
userEnrichmentResponse = requests.request("GET", userEnrichment_url, headers=userEnrichmentHeaders)
```

您可以看到，從userEnfurationHeaders中，entity_type和entity_value變數作為API呼叫的頭引數傳遞以及token變數傳遞。

若要驗證收到的回應，請使用print()函式。

```
print(userEnrichmentResponse.text)
```

輸出範例：

```
[ {
  "userDetails" : {
    "id" : "E4:5F:02:FF:xx:xx",
    "connectionStatus" : "CONNECTED",
    "tracked" : "No",
    "hostType" : "WIRELESS",
    "userId" : null,
    "duid" : "",
    "identifier" : "jonberrypi-1",
    "hostName" : "jonberrypi-1",
    "hostOs" : null,
    "hostVersion" : null,
    "subType" : "RaspberryPi-Device",
    "firmwareVersion" : null,
    "deviceVendor" : null,
    "deviceForm" : null,
    "salesCode" : null,
    "countryCode" : null,
    "lastUpdated" : 1721225220000,
    "healthScore" : [ {
      "healthType" : "OVERALL",
      "reason" : "",
      "score" : 10
    }, {
      "healthType" : "ONBOARDED",
      "reason" : "",
      "score" : 4
    }
  ]
} ]
!--- Output is suppressed
```

具有查詢引數的API

查詢引數可用來篩選API傳回的特定結果數目。這些引數會新增到API的URL中。

Get Device List API呼叫已測試。

```
GET https://10.88.244.133/dna/intent/api/v1/network-device
```

Get Device List API返回增加到Catalyst Center中的所有裝置的清單。如果請求特定裝置的詳細資訊，查詢引數有助於過濾特定資訊。

要驗證哪些查詢引數可用於API，請導航到Platform > Developer Toolkit > APIs > Get Device List，然後按一下API的名稱。此時將打開一個新窗口，並在Parameters選項下顯示API可用的查詢引數。

Get Device list

GET https://10.88.244.133/dna/intent/api/v1/network-device

Returns list of network devices based on filter criteria such as management IP address, mac address, hostname, etc. You can use the .* in any value to conduct a wildcard search. For example, to find all hostnames beginning with myhost in the IP address range 192.25.18.n, issue the following request: GET /dna/intent/api/v1/network-device?hostname=myhost.*&managementIpAddress=192.25.18.* If id parameter is provided with comma separated ids, it will return the list of network-devices for the given ids and ignores the other request parameters. You can also specify offset & limit to get the required list.

Cisco DevNet API Guide

Parameters Responses Code Preview

Request Query Parameters

Name	Description	DataType	Required	Default Value
hostname	hostname	array	No	
managementIpAddress	managementIpAddress	array	No	
macAddress	macAddress	array	No	
locationName	locationName	array	No	
serialNumber	serialNumber	array	No	
location	location	array	No	
family	family	array	No	
type	type	array	No	
series	series	array	No	

本例中使用了managementIpAddress和serialNumber查詢引數（請考慮不需要為API呼叫使用所有查詢引數）。繼續建立並指定兩個查詢引數的對應值。

```
managementIpAddress = '10.82.143.250'  
serialNumber = 'FD025160X9L'
```

如上所述，查詢引數在API的URL中增加，特別是在其末尾，使用後跟查詢引數的「？」。

如果要使用多個查詢引數，將「&」符號置於它們之間以形成所謂的查詢字串。

下一個示例顯示如何將查詢引數增加到儲存API呼叫的URL的deviceListUri變數中。

```
deviceListUri = "https://<CatalystCenterIP>/dna/intent/api/v1/network-device?managementIpAddress=" + m
```

請注意，先前建立的變數已附加至URL字串。換句話說，URL的整個字串如下所示：

```
deviceListUri = "https://<CatalystCenterIP>/dna/intent/api/v1/network-device?managementIpAddress=10.82
```

繼續API呼叫，將建立deviceListHeaders變數以儲存API報頭以及作為引數傳遞的token變數，並且deviceListResponse變數儲存API響應。

```
deviceListHeaders = {  
    'X-Auth-Token': token,  
}
```

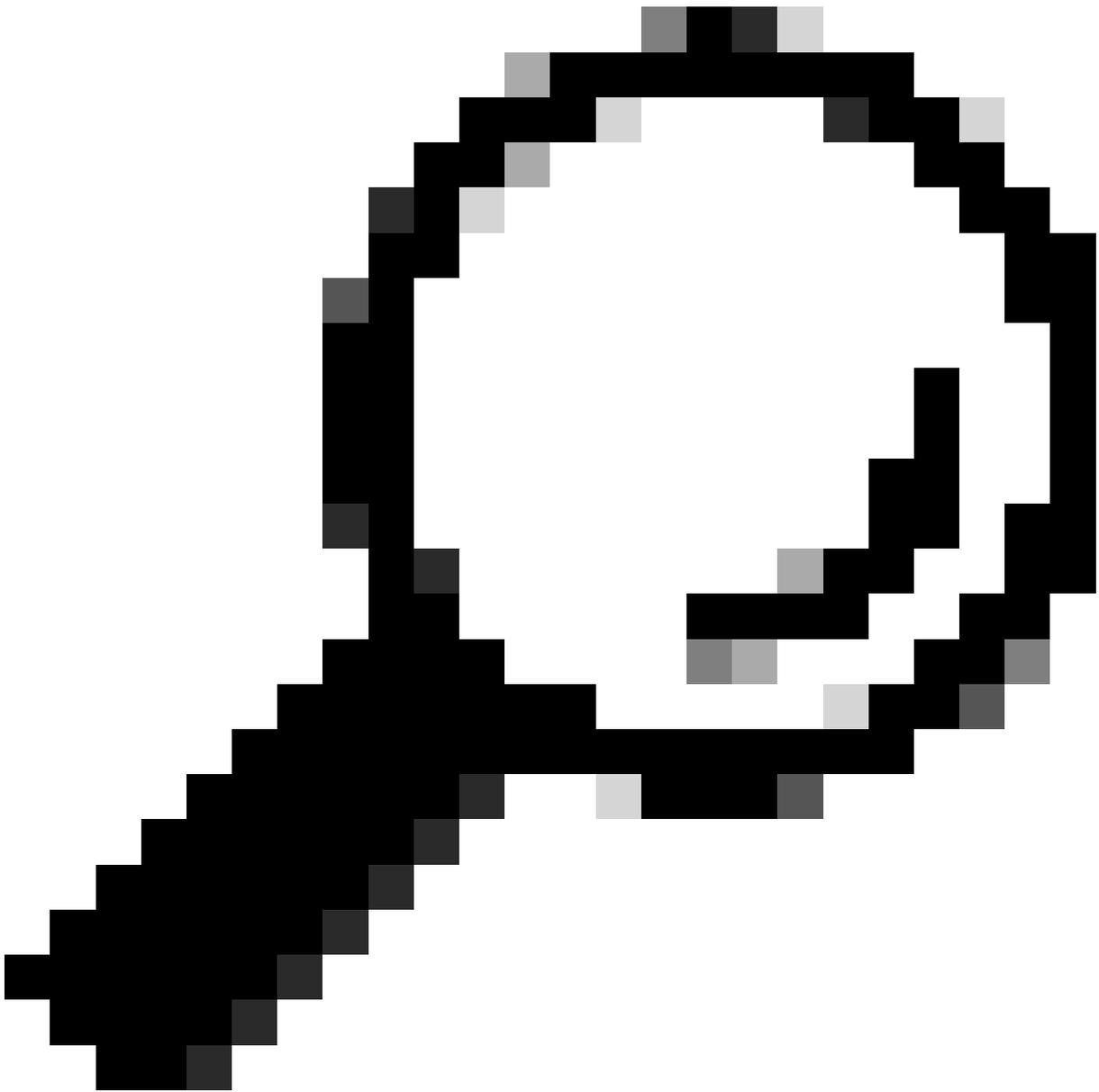
```
deviceListResponse = requests.request("GET", deviceListUri, headers=deviceListHeaders)
```

若要驗證收到的回應，您可以使用print()函式。

```
print(deviceListResponse.text)
```

輸出範例：

```
{"response":[{"family":"Switches and Hubs","description":"Cisco IOS Software [Cupertino], Catalyst L3 S  
!--- Output is suppressed
```



提示：要以更易讀的方式列印響應，您可以使用測試API部分中介紹的`json.loads()`和`json.dumps()`函式。

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）。