

# 對基於API的EPNM通知進行故障排除

## 目錄

---

[簡介](#)

[背景資訊](#)

[EPNM API通知](#)

[基本EPNM配置](#)

[面向連線的通知](#)

[運行WebSockets Python客戶端](#)

[訂購面向連線的客戶端](#)

[驗證消息、調試條目、show log、使用的檔名、SQL輸出](#)

[無連線通知](#)

[運行REST Web服務Python客戶端](#)

[訂用無連線客戶端](#)

[驗證消息、DEBUG條目、show log、使用的檔名、SQL輸出](#)

[結論](#)

[相關資訊](#)

---

## 簡介

本文檔介紹在使用REST API訪問裝置故障資訊時，如何排除EPNM通知故障。

## 背景資訊

您執行的客戶端必須能夠處理和訂閱進化的可程式設計網路管理器(EPNM)用來傳送通知的兩種機制中的任何一種。

## EPNM API通知

通知就有關網路的重要事件或問題向網路管理員和操作員發出警報。這些通知有助於確保快速檢測並解決潛在問題，從而減少停機時間，提高整體網路效能。

EPNM可以處理不同的方法，例如通過電子郵件傳送通知、向指定接收者傳送簡單網路管理協定(SNMP)陷阱或向外部系統日誌伺服器傳送系統日誌消息。除了這些方法外，EPNM還提供一個具代表性的狀態傳輸應用程式設計介面(REST API)，可用於檢索有關庫存、警報、服務啟用、模板執行和高可用性的資訊。

目前支援使用兩種不同機制進行基於API的通知：

- 面向連線的通知：客戶端訂閱預定義的URL，並通過安全的HTTPS通道使用具有基本身份驗證的WebSocket客戶端。
- 無連線通知：使用者應具有能夠作為POST請求接受可擴展標籤語言(XML)和/或JavaScript對象表示法(JSON)負載的REST Web服務。

所有通知都共用相同的架構，可以以JSON或XML格式檢索。

## 基本EPNM配置

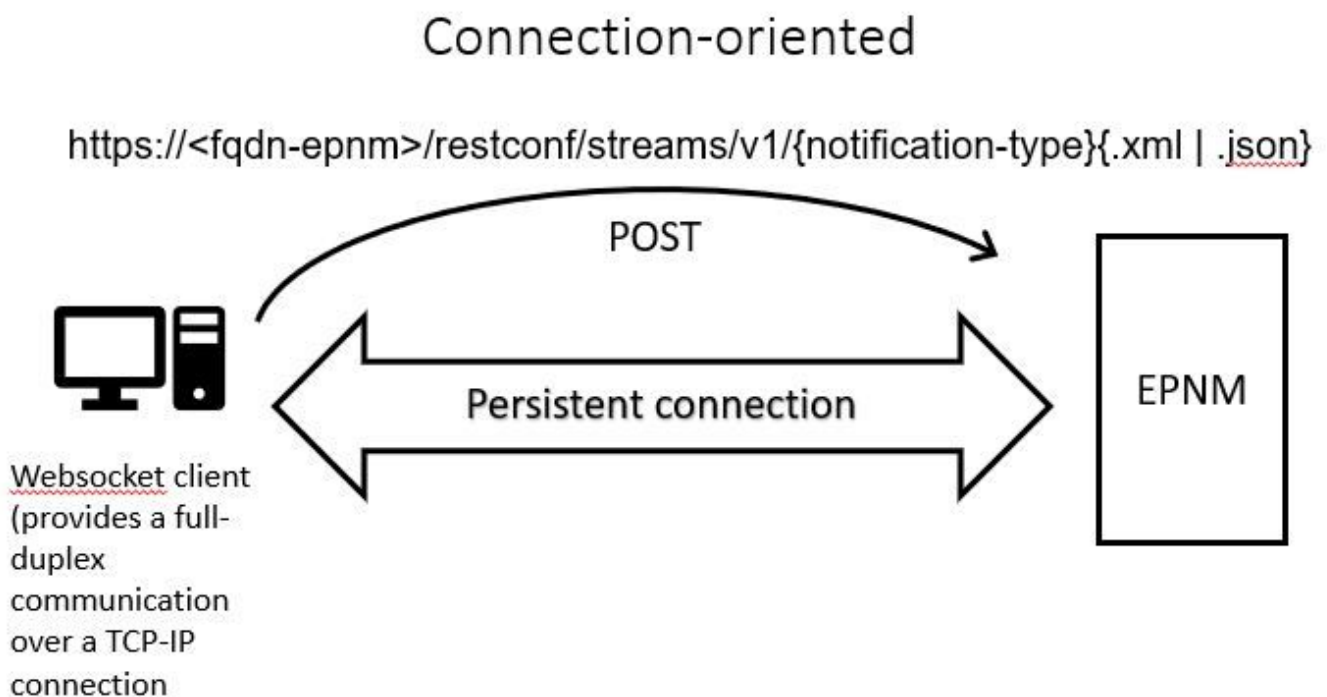
預設情況下，警報和庫存通知處於禁用狀態。若要啟用它們，請更改 `restconf-config.properties` 檔案（無需重新啟動EPNM應用程式）：

```
/opt/CSC01umos/conf/restconf/restconf-config.properties
```

```
epnm.restconf.inventory.notifications.enabled=true  
epnm.restconf.alarm.notifications.enabled=true
```

## 面向連線的通知

在圖片中，客戶端電腦運行WebSocket，並使用預定義的URL、基本身份驗證和通過安全HTTPS通道訂閱EPNM。



## 運行WebSockets Python客戶端

Python中的WebSocket-client庫可用於在客戶端電腦中建立WebSocket。

```
import websocket  
import time  
import ssl  
import base64
```

```

def on_message(ws, message):
    print(message)

def on_error(ws, error):
    print(error)

def on_close(ws, close_status_code, close_msg):
    print("### closed \###")

def on_open(ws):
    ws.send("Hello, Server!")

if __name__ == "__main__":
    username = "username"
    password = "password"
    credentials = base64.b64encode(f"{username}:{password}".encode("utf-8")).decode("utf-8")
    headers = {"Authorization": f"Basic {credentials}"}
    websocket.enableTrace(True)
    ws = websocket.WebSocketApp("wss://10.122.28.3/restconf/streams/v1/inventory.json",
                                on_message=on_message,
                                on_error=on_error,
                                on_close=on_close,
                                header=headers)

    ws.on_open = on_open
    ws.run_forever(sslopt={"cert_reqs": ssl.CERT_NONE})

```

## 訂購面向連線的客戶端

此代碼設定一個WebSocket客戶端，該客戶端可在以下位置訂閱EPNM:

`wss://10.122.28.3/restconf/streams/v1/inventory.json`. 它使用Python `WebSocket`庫，用於建立連線和處理傳入和傳出消息。訂閱還可以是（根據要訂閱的通知型別）：

- `/restconf/streams/v1/alarm{.xml | .json}`
- `/restconf/streams/v1/service-activation{.xml | .json}`
- `/restconf/streams/v1/template-execution{.xml | .json}`
- `/restconf/streams/v1/all{.xml | .json}`

其 `on_message` 中，`on_error` 和 `on_close` 函式是回撥函式，在WebSocket連線分別收到消息、遇到錯誤或關閉時呼叫這些函式。其 `on_open` 函式是當WebSocket連線建立並準備使用時呼叫的回撥。

其 `username` 和 `password` 變數設定為訪問遠端伺服器所需的登入憑據。然後，這些憑證將使用 `base64` 模組並新增到WebSocket請求的標頭中。

其 `run_forever` 對WebSocket對象呼叫方法，以便啟動連線，使其無限期開啟，並偵聽來自伺服器的消息。其 `sslopt` 引數用於配置連線的SSL/TLS選項。其 `CERT_NONE` 標籤禁用證書驗證。

運行代碼以使WebSocket準備好接收通知：

```

(env) devasc@labvm:~/epnm$ python conn-oriented.py
--- request header ---
GET /restconf/streams/v1/inventory.json HTTP/1.1

```

```
Upgrade: websocket
Host: 10.122.28.3
Origin: https://10.122.28.3
Sec-WebSocket-Key: YYYYYYYYYY
Sec-WebSocket-Version: 13
Connection: Upgrade
Authorization: Basic XXXXXXXXXXXX
```

```
-----
--- response header ---
HTTP/1.1 101
Set-Cookie: JSESSIONID=5BFB68B0126226A0A13ABE595DC63AC9; Path=/restconf; Secure; HttpOnly
Strict-Transport-Security: max-age=31536000;includeSubDomains
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Upgrade: websocket
Connection: upgrade
Sec-WebSocket-Accept: Ozns7PGgHjrXj0nAgn1hbyVKPjc=
Date: Thu, 30 Mar 2023 16:18:19 GMT
Server: Prime
-----
```

```
Websocket connected
++Sent raw: b'\x81\x8e\x99ry;\xfc\x1e\x15\x1c\xb5R*\x16\xeb\x04\x1c\x01\xb8'
++Sent decoded: fin=1 opcode=1 data=b'Hello, Server!'
++Rcv raw: b'\x81\x0eHello, Server!'
++Rcv decoded: fin=1 opcode=1 data=b'Hello, Server!'
Hello, Server!
```

您可以使用以下資料庫查詢檢查伺服器的通知訂閱：

```
ade # ./sql_execution.sh "SELECT * from RstcnfNtfctnsSbscrptnMgr WHERE CONNECTIONTYPE = 'connection-or"
```

為了更好地將 `conn-oriented.txt` 檔案（即資料庫查詢的結果），您可以使用類似工具將其轉換為 HTML `aha`（此處在Ubuntu機器中說明了其用途）：

```
devasc@labvm:~/tmp$ sudo apt-get install aha
devasc@labvm:~/tmp$ cat conn-oriented.txt | aha > conn-oriented.html
```

然後開啟 `conn-oriented.html` 瀏覽器中的檔案：

| ID         | INSTANCE_VERSION | CLASSNAME                | CONNECTIONTYPE      | ENDPOINTURL                          | SUBSCRIBEDUSER | SUBSCRIPTIONCREATIONTIME     | SUBSCRIPTIONID      | SUBSCRIPTIONTOPIC | SUBSCRIPTIONUPDATETIME       |
|------------|------------------|--------------------------|---------------------|--------------------------------------|----------------|------------------------------|---------------------|-------------------|------------------------------|
| 2361938571 |                  | @_cnfNtfctnsSbscrptnMgr3 | connection-oriented | d64d9882-c95c-439b-a7e7-65016623Fee1 | root           | Mon Aug 28 16:13:04 BRT 2023 | 3648313822269611499 | Inventory         | Mon Aug 28 16:13:04 BRT 2023 |

根據EPNM線上文檔，一旦建立，相同的連線將在應用程式的整個生命週期中保持活動狀態：

- 直到客戶端與伺服器斷開連線
- 直到伺服器因維護或故障切換而停止運行

如果由於某種原因，您需要刪除特定訂閱，則您可以傳送 HTTP DELETE 請求 SUBSCRIPTIONID 在 URL 中指定 https://

.舉例來說：

```
devasc@labvm:~/tmp$ curl --location --insecure --request DELETE 'https://10.122.28.3/restconf/data/v1/c
```

驗證消息、調試條目、show log，使用的檔名，SQL 輸出

為了排除使用面向連線的機制的客戶端無法正確接收通知的原因，可以運行指定的DB查詢並檢查訂閱是否存在。如果沒有，請要求客戶端所有者確保發出訂閱。

同時，您可以在以下位置啟用調試級別 com.cisco.nms.nbi.epnm.restconf.notifications.handler.NotificationsHandlerAdapter 以便每次傳送訂閱時都捕獲它：

```
ade # sudo /opt/CSC01umos/bin/setLogLevel.sh com.cisco.nms.nbi.epnm.restconf.notifications.handler.Notifi
```

傳送訂閱後，您可以檢查中是否顯示包含WebSocket客戶端IP地址的條目 localhost\_access\_log.txt:

```
ade # zgrep -h '"GET /restconf/streams/. * HTTP/1.1" 101' $(ls -lt /opt/CSC01umos/logs/localhost_access_
```

最後，再次檢查DB（注意時間戳與中的條目匹配） localhost\_access\_log.txt).

| H | CONNECTIONTYPE      | ENDPOINTURL                          | ISENDPOINTREACHABLE | NOTIFICATIONFORMAT | SUBSCRIBEDUSER | SUBSCRIPTIONCREATIONTIME     | SUBSCRIPTIONID      | SUBSCRIPTIONI |
|---|---------------------|--------------------------------------|---------------------|--------------------|----------------|------------------------------|---------------------|---------------|
|   | connection-oriented | 852a674a-e3d0-4ecc-8ea0-787af30f1305 | 0                   | json               | root           | Mon Aug 28 22:17:06 BRT 2023 | 8743327441517764088 | inventory     |

下一個日誌顯示何時傳送訂閱的POST請求：

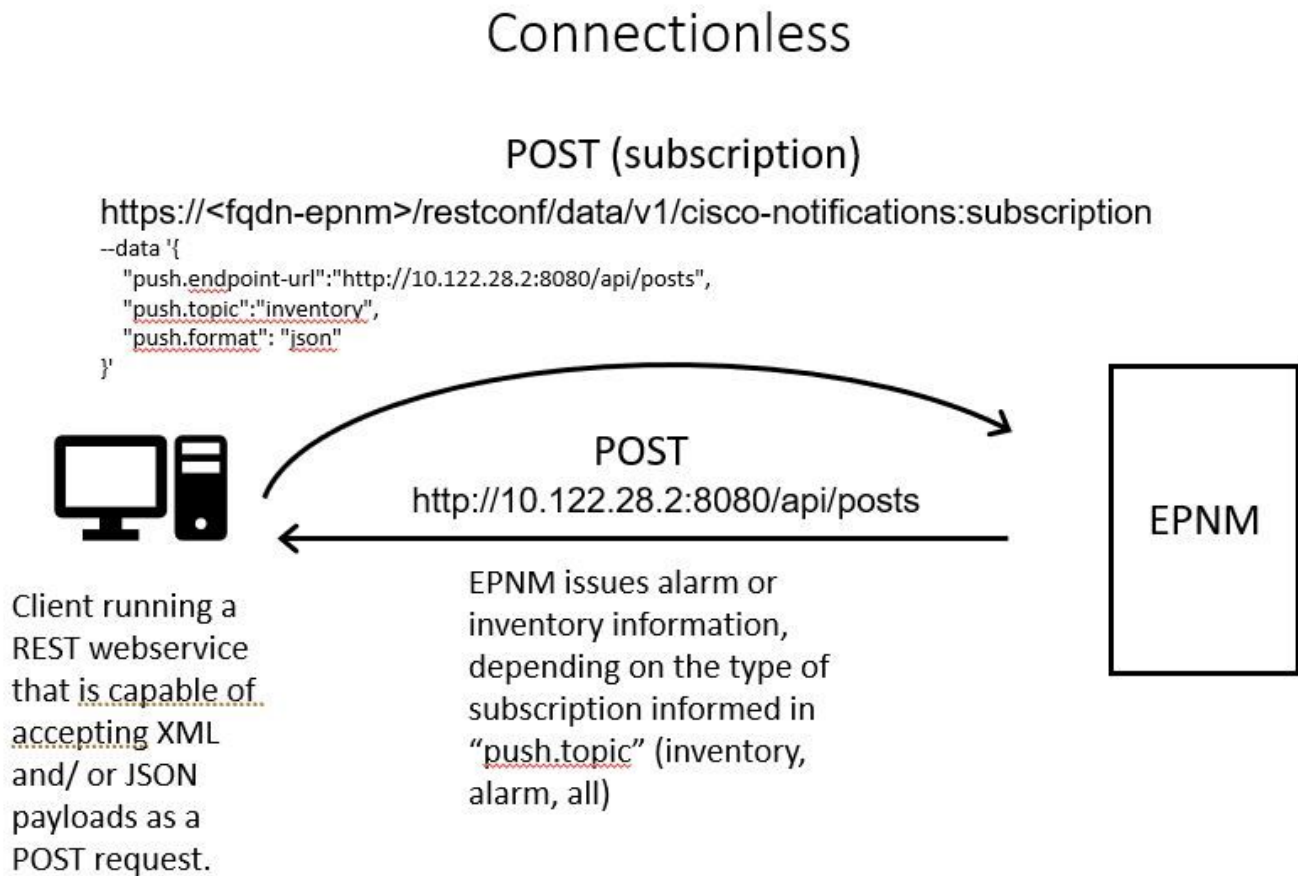
```
ade # grep -Eh 'DEBUG com.cisco.nms.nbi.epnm.restconf.notifications.handler.NotificationsHandlerAdapter
```

只要連線保持活動狀態，就會從EPN-M伺服器向訂閱通知的所有客戶端傳送推送更改更新型別的通知。此示例顯示當NCS2k的主機名更改時EPNM傳送的一個通知：

```
{ "push.push-change-update":{ "push.notification-id":2052931975556780123, "push.topic":"inventory", "pu
```

無連線通知

下一個是工作流程，在 connectionless 通知：



運行REST Web服務Python客戶端

使用者應具有能夠接受XML和/或JSON負載作為POST請求的REST Web服務。此REST服務是 Cisco EPNMrestconf通知框架發佈通知。這是在遠端電腦上安裝的REST Web服務示例：

```
from flask import Flask, request, jsonify app = Flask(__name__) @ app.route('/api/posts', methods=['POST
```

這是一個Python Flask Web應用程式，用於定義單個端點 /api/posts 接受 HTTP POST 請求。其 create\_post() 每當一個 HTTP POST 請求 /api/posts。在 create\_post() 函式，使用檢索來自傳入請求的資料 request.get\_json()，返回JSON負載的字典。然後，負載會列印為 print(post\_data) 用於偵錯目的。然後，使用金鑰建立響應消息 message 和價值 Post created successfully (以詞典格式)。然後此響應消息返回到HTTP狀態代碼為201 (已建立) 的客戶端。

其 if \_\_name\_\_ == '\_\_main\_\_': block是標準Python構造，檢查指令碼是否作為主程式運行，而不是作為模組匯入。如果指令碼作為主程式運行，它將啟動Flask應用程式並在指定的IP地址和埠上運行該應用程式。其 debug=True 引數啟用調試模式，該模式提供詳細的錯誤消息，並在對代碼進行更改時自動重新載入伺服器。

運行程式以啟動 REST web服務：

```
(venv) [apinelli@centos8_cx1abs_spo app]$ python connectionless.py * Serving Flask app 'connectionless'
```

訂用無連線客戶端

使用者訂閱通知：REST服務終結點與主題一起傳送以便訂購。在這種情況下，主題是 all。

```
[apinelli@centos8_cx1abs_spo ~]$ curl --location -X POST --insecure 'https://10.122.28.3/restconf/data/'
```

預期回覆是2011年的回覆，以及回覆正文中訂閱的細節：

```
{ "push.notification-subscription": { "push.subscription-id": 7969974728822328535, "push.subscribed-user": "root" }
```

可以通過GET請求獲取使用者訂閱的通知清單：

```
curl --location --insecure 'https://10.122.28.3/restconf/data/v1/cisco-notifications:subscription' \ --
```

答覆如下：

```
{ "com.response-message": { "com.header": { "com.firstIndex": 0, "com.lastIndex": 1 }, "com.data": { "push.notification-subscription": { "push.subscription-id": 7969974728822328535, "push.subscribed-user": "root" }
```

驗證消息、調試條目、show log, 使用的檔名，SQL輸出

從響應中注意，存在兩個預訂：一個用於 all ("push.topic": "all") 一個用於庫存 ("push.topic": "inventory")。您可以通過查詢資料庫來確認它(請注意，預訂型別為「無連線」，且 SUBSCRIPTIONID 欄位與 GET 命令以黃色突出顯示)：

```
ade # ./sql_execution.sh "SELECT * from RstcnfNtftctnsSbscrptnMngr WHERE CONNECTIONTYPE = 'connection-less'"
```

| ID         | INSTANCE_VERSION | CLASSNAME                | CONNECTIONTYPE  | ENDPOINTURL                       | SUBSCRIBEDUSER | SUBSCRIPTIONCREATIONTIME     | SUBSCRIPTIONID      | SUBSCRIPTIONTOPIC |
|------------|------------------|--------------------------|-----------------|-----------------------------------|----------------|------------------------------|---------------------|-------------------|
| 2361930573 | 0                | cnfNtftctnsSbscrptnMngr3 | connection-less | http://10.122.28.2:8080/api/posts | root           | Tue Aug 29 10:02:05 BRT 2023 | 7969974728822328535 | all               |
| 337897630  | 0                | cnfNtftctnsSbscrptnMngr3 | connection-less | http://10.122.28.2:8080/api/posts | root           | Fri Mar 31 17:45:47 BRT 2023 | 2985507600170167151 | inventory         |

如果需要刪除無連線訂閱，可以傳送 HTTP DELETE 請求，包含您要刪除的訂閱ID。假設您要刪除 **subscription-id 2985507860170167151**:

```
curl --location --insecure --request DELETE 'https://10.122.28.3/restconf/data/v1/cisco-notifications:s
```

現在，如果再次查詢DB，您只會看到使用 SUBSCRIPTIONID 等於 7969974728822328535.

當庫存發生變化時，客戶端將列印通知(其型別與 connection-oriented 一節中顯示的通知關於 connected-oriented 客戶端)，然後是 201響應：

```
(venv) [apinelli@centos8_cx1abs_spo app]$ python connectionless.py * Serving Flask app 'connectionless'
```

## 結論

在本檔案中，有兩種基於API的通知可以在EPNM(connectionless和 connection-oriented文中給出了各個客戶端的示例，這些客戶端可以作為模擬的依據。

## 相關資訊

- [https://www.cisco.com/c/dam/en/us/td/docs/net\\_mgmt/eprn\\_manager/RESTConf/Cisco\\_Evolved\\_Programmable\\_Network\\_Manager\\_5\\_1\\_2\\_F](https://www.cisco.com/c/dam/en/us/td/docs/net_mgmt/eprn_manager/RESTConf/Cisco_Evolved_Programmable_Network_Manager_5_1_2_F)
- [技術支援與文件 - Cisco Systems](#)



## 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）。