

tags: postman api

使用 Postman 探索 VMware API - Part 1



原本只是想透過 **HOL-2225-03-NET NSX Migration Coordinator** 線上實驗有關 NSX V2T 的流程操作。當然移轉場景還是以 **DFW-Only** 為主，為了要實現“告訴 NSX-T Manager 手動移轉已經完成”，這個動作現在還是需要 API 完成。

```
POST https://{nsxt-mgr-ip}/api/v1/migration?
action=finalize_infra
```

發現在 HOL 線上實驗室的环境裡有佈署一套 **Postman**，玩著玩著就直接惡搞了，也順便學習一下之前沒碰的 Postman。之前在本機都是透過 Python 和 cURL 測試的。

- **強烈推薦 HOL-2225 整個系列的線上實驗室**，可以增進 NSX-T 有關的知識和技術。
- 線上實驗室環境中的 Postman 就是用來佈署實驗環境的，裡面的請求範例，剛好可以作為學習範本。
- 嘗試以 HOL-2225-03-NET 環境中的 **Region B**，佈署一組移轉 DFW-Only 的環境。不過這個環境僅有一組 vCSA，但同時有 NSX-V 和 NSX-T，後續可能要傷腦筋將 NSX-V 移除 vCSA 的步驟？

• 使用 Postman 探索 VMware API - Part 1

- 什麼是 Postman
- 安裝 Postman
- 註冊帳號
- 登入 Postman
- 建立並儲存第一個請求
 - 點選 Create New
 - 點選 HTTP Request。
 - 建立 GET 請求
- vSphere API 探索啟程

- 建立 VMware vSphere API Test Collection
- 建立 vCSA 連線授權資訊
- 建立 vcsa_url 變數
- 建立並執行會話 ID 請求
- 持續使用會話 ID
- 增加請求狀態確認
- Console 中顯示訊息
- 待續...
- 附錄
 - 安裝 Linux 桌面版 Postman
 - 建立工作區
 - 刪除工作區
 - 移動集合至不同工作區
 - 工作區其他資訊
 - Desktop Agent 問題排除
- 參考

什麼是 Postman

Postman 是一個可以讓我們測試 HTTP 請求的工具。藉由 Postman 工具對於 API 進行不同的請求方式，以便驗證資料結果是否有如預期。

安裝 Postman

目前 Postman 有兩個版本。

- 桌面版。

The Postman app

The ever-improving Postman app (a new release every two weeks) gives you a full-featured Postman experience.

 Linux 64-bit

By downloading and using Postman, I agree to the [Privacy Policy](#) and [Terms](#).

Version 9.1.5 · [Release Notes](#) · [Product Roadmap](#)

Not your OS? Download for Windows ([x32](#) / [x64](#)) or Mac ([Intel Chip](#) / [Apple Chip](#))

- 可至官網下載使用。[連結](https://www.postman.com/downloads/)
- 安裝參考 Postman App for Linux (https://hackmd.io/uY4B-xIURh2ISCzs0vK_3g#%E5%AE%89%E8%A3%9D-Linux-%E6%A1%8C%E9%9D%A2%E7%89%88-Postman).

- 網路版。[開啟 Postman](https://go.postman.co/home).

Postman on the web

You can now access Postman through your web browser.
Simply create a free Postman account, and you're in.

Launch Postman

網路版限制

Postman 目前以下功能只能在桌面應用中使用，網路版本並不提供，若有以下需求，請直接安裝桌面版本。

- 即時預覽（Live preview）：無法像 Postman 桌面應用，在輸入請求配置時看到所有請求標頭即時更新，只會在編輯時看到授權標頭（Authorization headers）更新。
- 將請求回應儲存至文件（Saving responses to file）
- 證書和代理（Certificates and Proxy）：將採用瀏覽器的設定值並且不能被 Postman 覆蓋。
- Postman 攔截器（Postman Interceptor）

簡單說，使用 Postman interceptor 可以攔截到瀏覽器的網頁請求，並同步傳至 postman 桌面版客戶端軟體，只需要在 postman 中進行適度修改，就可作為傳送請求（send）使用。

注意

- 如果使用的是 Postman Web Client，則還需要下載 Postman 桌面代理（Desktop Agent）。Postman 代理克服了瀏覽器的跨對象資源共享 (CORS) 限制，並有助於從 Postman 的瀏覽器版本發送 API 請求。
- 如果真要使用 Postman 就用桌面版本吧！

Plans and Pricing

Free	Team	Business	Enterprise
The Postman Free plan is the perfect place to start designing, developing, and testing APIs.	The Postman Team plan helps you collaborate across your team to design, develop, and test APIs faster.	The Postman Business plan helps you rapidly scale your organization and manage the entire API workflow, all in one place.	The Postman Enterprise plan is our enterprise-grade offering, with powerful tools to securely manage, organize, and accelerate large-scale API development.
\$0	\$12	\$24	\$69
Per user/month, billed annually \$15 per user/month, billed monthly	Per user/month, billed annually \$30 per user/month, billed monthly	Per user/month, billed annually \$30 per user/month, billed monthly	Per user/month, billed annually
Sign Up	Buy Now	Buy Now	Contact Sales
All the basics for up to 3 team members	Everything in Free, plus: <ul style="list-style-type: none">✓ 10x Postman API usage limit✓ 10 integrations	Everything in Team, plus: <ul style="list-style-type: none">✓ 100x Postman API usage limits✓ 50 integrations✓ Basic roles and permissions✓ Single Sign On (SAML)	Everything in Business, plus: <ul style="list-style-type: none">✓ 1000x Postman API usage limits✓ 100 integrations✓ Reporting and Analytics✓ User groups✓ Automated Provisioning and Deprovisioning (SCIM)✓ Domain Capture*

註冊帳號

如果有以下的需求，建議可以註冊一組 Postman 帳號使用。

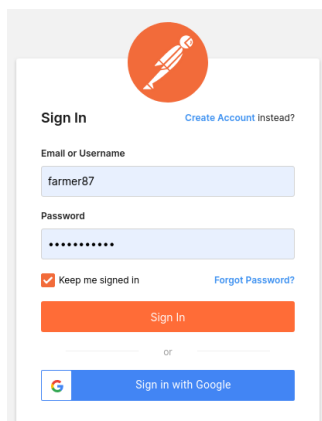
[註冊帳號] (<https://identity.getpostman.com/signup>)

- 在 Postman Workspaces 中組織您的所有 API 開發
- 跨設備同步您的 Postman 數據
- 將您的數據備份到 Postman 雲
- 免費！

登入 Postman

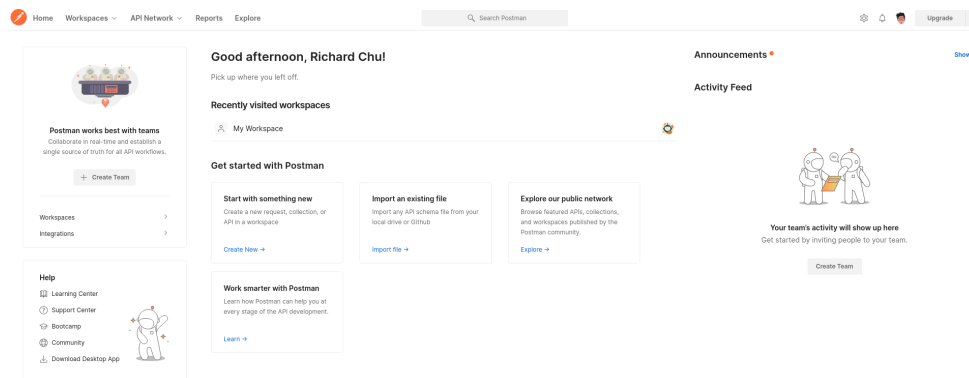
請根據選擇使用的 Postman 版本進行登入。以下將以網路版本作為預設界面。使用註冊的帳號登入 **[網路版 Postman]**

(<https://go.postman.co/home>)



The image shows the Postman Sign In interface. At the top, there is a red circular icon with a white Postman logo. Below it, the text "Sign In" is displayed, followed by a link "Create Account Instead?". The form includes fields for "Email or Username" (containing "farmer87") and "Password" (masked with dots). There is a checkbox for "Keep me signed in" and a link "Forgot Password?". A large orange "Sign In" button is present. Below the button, the word "or" is shown, followed by a Google logo and a blue button labeled "Sign in with Google".

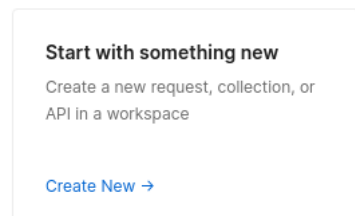
登入畫面



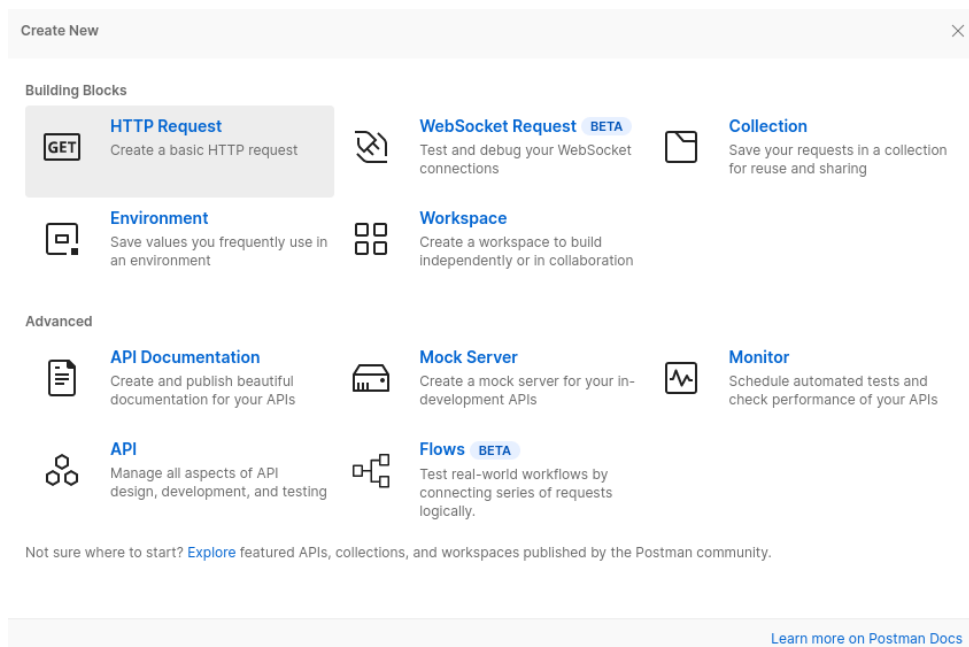
建立並儲存第一個請求

Get started with Postman > Start with something new.

點選 Create New

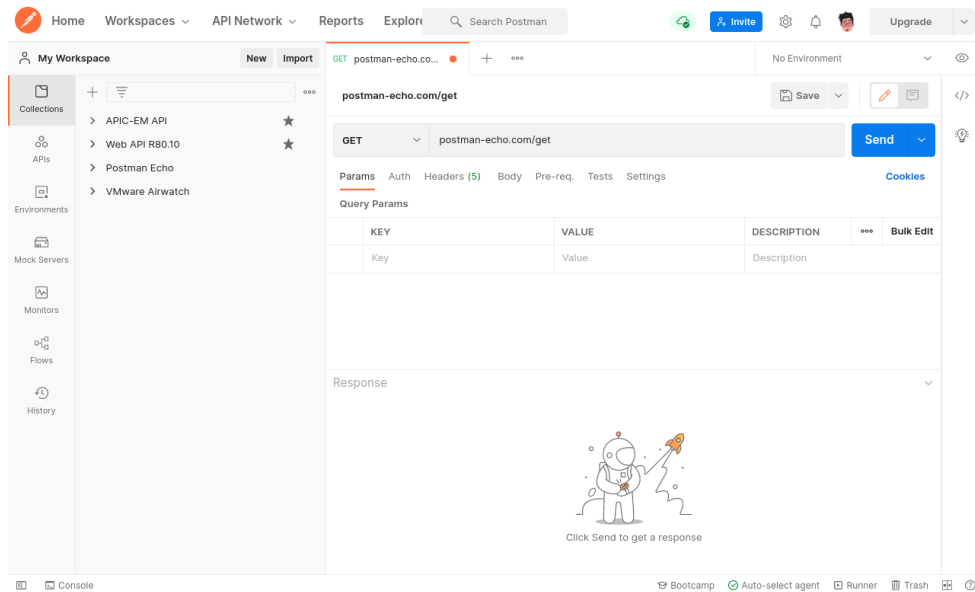


點選 HTTP Request。

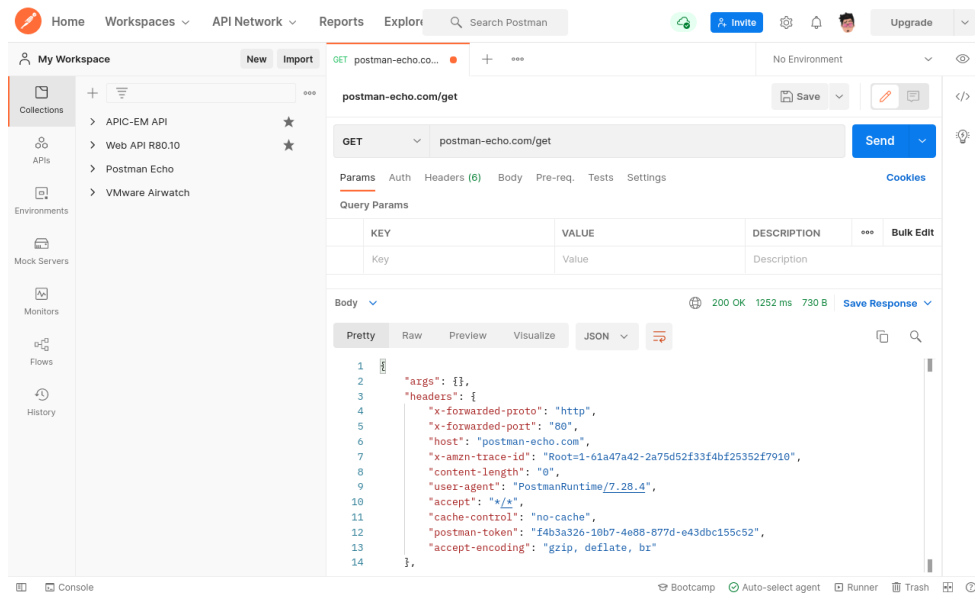


建立 GET 請求

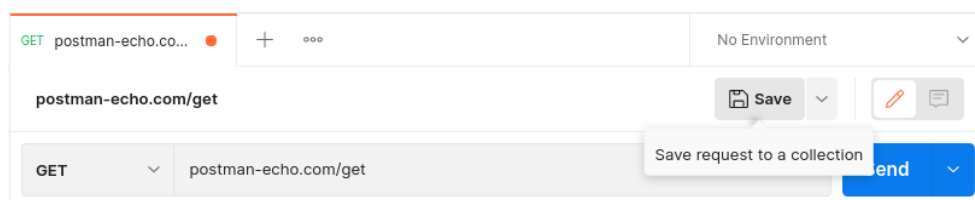
URL 區域輸入 `postman-echo.com/get`，點選 **Send** 執行請求。



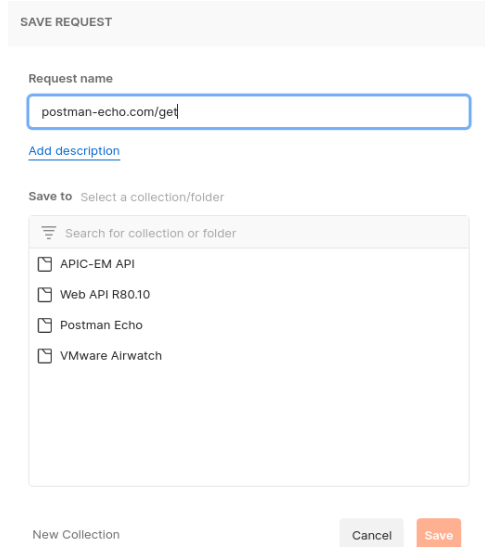
在下方區域會顯示請求回應的相關資訊。



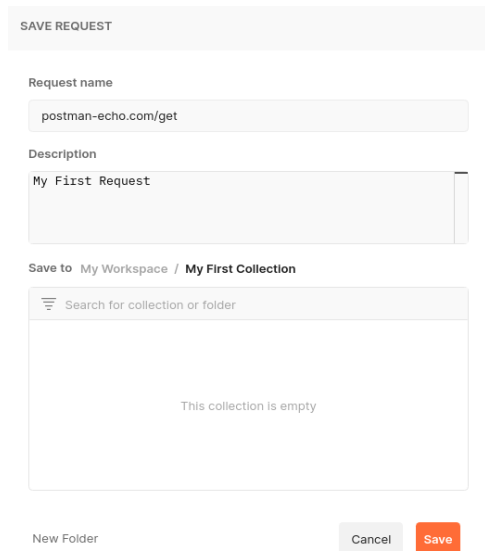
點選 **save** 儲存第一個請求。



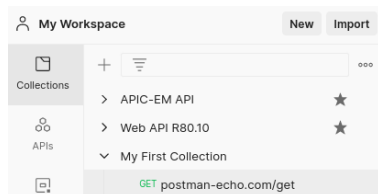
點選 **Add description** 可增加請求說明描述。點選 **New Collection** 可建立並儲存此請求至新的 Collection



點選 **Save** 完成。



從 **Workspaces > Collections** 進行檢視。



vSphere API 探索啟程

有關 vSphere API 詳細資訊，請參考以下 VMware 官方說明：

- **VMware Developer Documentation** (<https://developer.vmware.com/apis>)
- **vSphere Automation API Reference** (<https://developer.vmware.com/apis/vsphere-automation/latest/>)

接著我們嘗試先建立與 vCSA 連線的測試請求。

建立 VMware vSphere API Test Collection

New Collection × + ...

VMware vSphere API Test Share Run Save Copy 0 Eye 0 ...

Authorization Pre-request Script Tests Variables

This authorization method will be used for every request in this collection. You can override this by specifying one in the request.

Type No Auth ▾

建立 vCSA 連線授權資訊

在集合的授權（Authorization）選項，根據要連線的 vCSA(API Endpoint)，選擇連線授權類型為 **Basic Auth**，並輸入正確的 vCSA 登入帳戶資訊，用來建立預設的連線認證方式。

Authorization ● Pre-request Script Tests Variables

This authorization method will be used for every request in this collection. You can override this by specifying one in the request.

Type Basic Auth ▾

The authorization header will be automatically generated when you send the request.
[Learn more about authorization](#)

Username administrator@vsphere.local

Password

☐ Show Password

建立 vcsa_url 變數

接著在集合的變數（Variables）選項，先建立 vCSA 變數（Variable or Key） **vcsa_ur1** 及設定值（Value）。

Authorization ● Pre-request Script Tests Variables ●

These variables are specific to this collection and its requests. [Learn more about collection variables](#)

	VARIABLE	INITIAL VALUE ⓘ	CURRENT VALUE ⓘ	...	Persist All	Reset All
<input checked="" type="checkbox"/>	vcsa_url	https://vcsa67.sysagelab.com	https://vcsa67.sysagelab.com			
	Add a new variable					

⋮

完成變數建立後，點選 **Save** 保存。

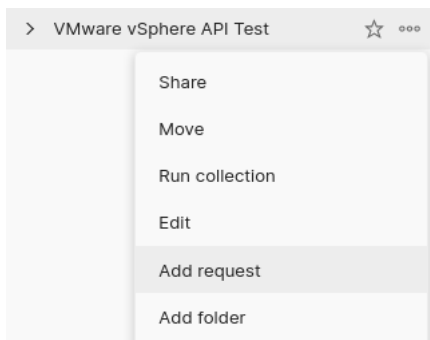
Share Run Save Copy

Save collection

注意 若有進行任何設定調整後，務必點擊 **Save** 進行儲存。

建立並執行會話 ID 請求

接著開始建立與 vCSA 的連線請求。在集合中新增一個新的請求。



完成以下請求的建立。

- 名稱: **Create vCSA Session**
- 方式: **POST**
- URL: **{{vcsa_url}}/api/session**

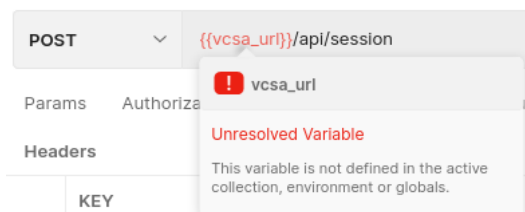
VMware vSphere API Test / Create vCSA Session

POST {{vcsa_url}}/api/session

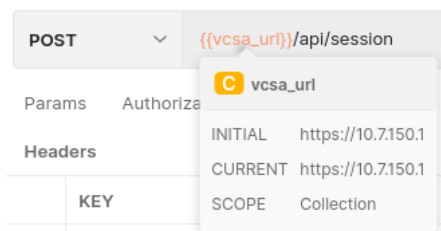
vSphere Automation API > CIS REST APIs > CIS Session > **Create Session** (<https://developer.vmware.com/apis/vsphere-automation/latest/cis/api/session/post/>)

注意

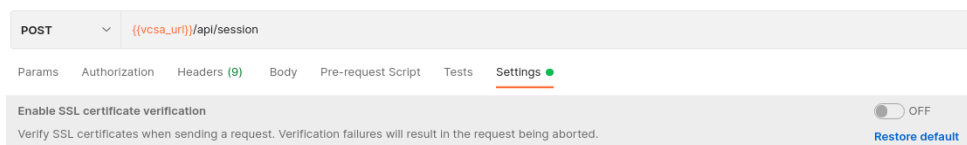
- 若變數不存在，將滑鼠移至該變數，會顯示 **Unresolved Variable** 訊息。



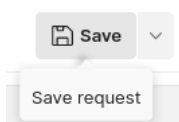
- 若變數設定正確，則會顯示以下正確訊息。



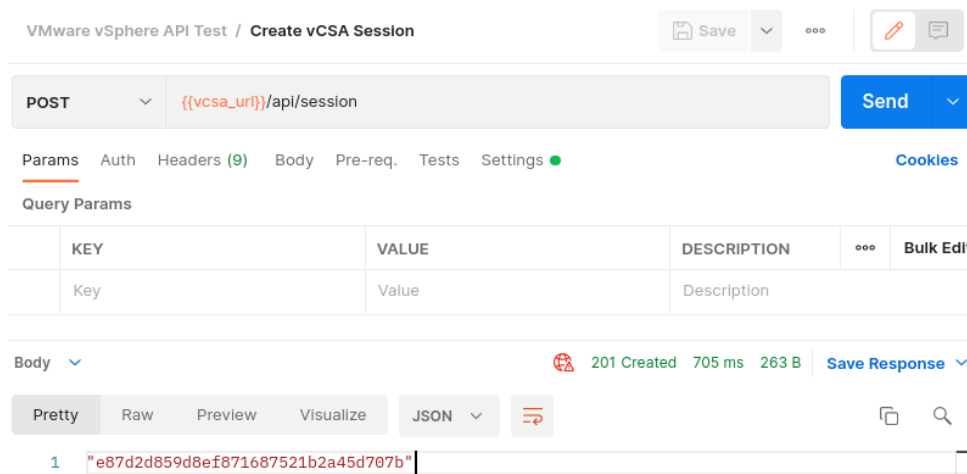
選擇 **Settings** 選項，將 **Enable SSL certificate verification** 取消（OFF），並避免憑證驗證產生的錯誤。



最後別忘了儲存設定。



執行該請求並確認回應狀態。回應狀態碼為 201，且回應主體（Body）回傳一個字串（string），該字串可作為後續請求驗證使用。



注意 如果使用網路版 Postman，如果發生以下情形。請參考附錄 **Desktop Agent 問題排除**。

Could not send request

Cloud Agent Error: Can not send requests to reserved address. Make sure address is publicly accessible or select a different agent.

Use Postman's Desktop Agent

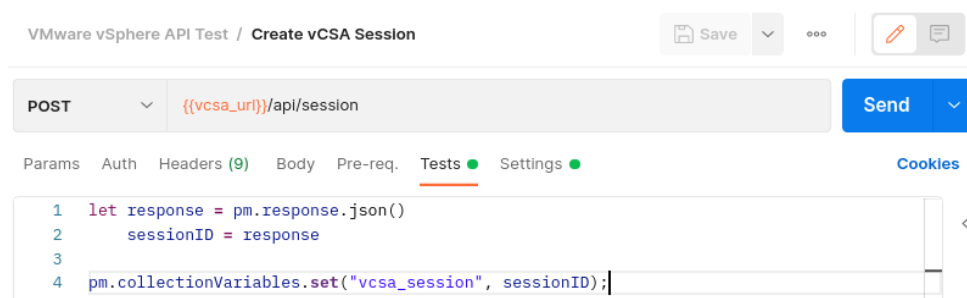
持續使用會話 ID

為了可以使用上述回應字串作為後續請求驗證使用，必須在 **Tests** 選項中增加腳本檔，將回應字串值儲存並建立一個集合中使用的變數 **vcsa_session**。

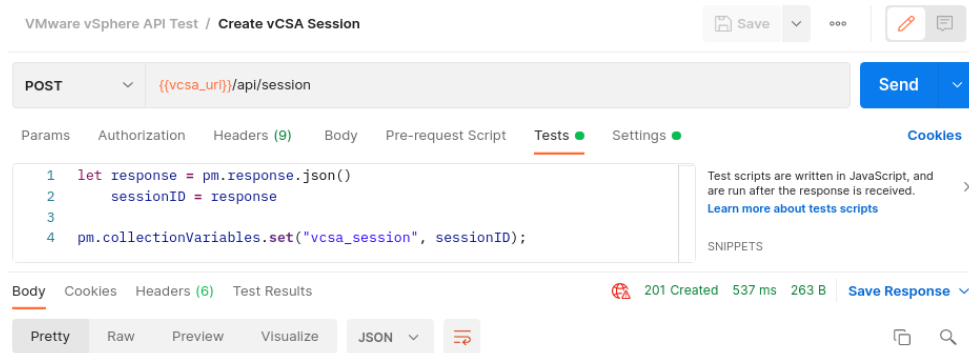
Postman 中 **Pre-request Scripts** 和 **Test Scripts** 採用 **JavaScript** 語法。

請加入以下範例腳本檔內容，並儲存（Save）後送出（Send）請求測試。

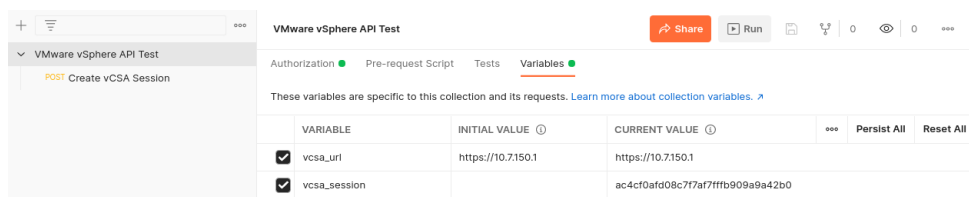
```
1 let response = pm.response.json()
2   sessionID = response
3
4 pm.collectionVariables.set("vcsa_session", sessionID);
```



回應狀態碼 201 為正確狀態，且回傳一組字串“ac4c...42b0”。

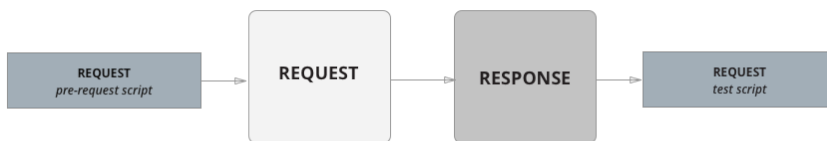


讓我們回到集合 VMware vSphere API Test 的變數選項，變數 vcsa_session 已自行建立，並引入相同的字串“ac4c...42b0”至 CURRENT VALUE。

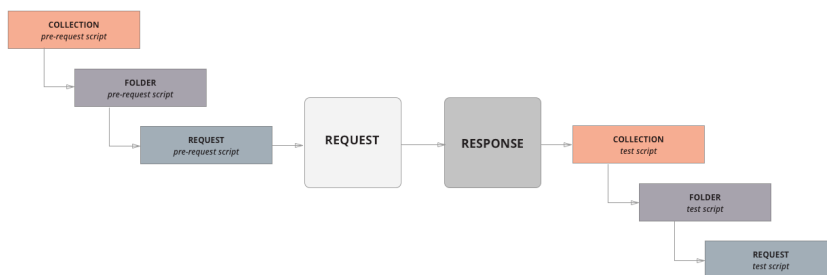


Postman 腳本檔執行順序

- 單一的請求



- 集合中的請求



增加請求狀態確認

從 vCSA API Explorer - /api/session 的回應碼得知相關的回應狀態碼。

☰vSphere Client

🔍

開發人員中心

概觀API Explorer程式碼擷取

/api/session

▼POST

Response types

Code/Reason	Model
201 Newly created session token to be used for authenticating further requests.	None
401 if the session creation fails due to request specific issues. Due to the security nature of the API the details of the error are not disclosed. Please check the following preconditions if using a SAML token to authenticate: - the supplied token is delegate-able. - the time of client and server system are synchronized. - the token supplied is valid. - if bearer tokens are used check that system configuration allows the API endpoint to accept such tokens.	VapiStdErrorsUnauthenticated { ... }
503 if session creation fails due to server specific issues, for example connection to a back end component is failing. Due to the security nature of this API further details will not be disclosed in the error. Please refer to component health information, administrative logs and product specific documentation for possible causes.	VapiStdErrorsServiceUnavailable { ... }

我們可以透過 **Test** 腳本檔，撰寫回應狀態值的確認動作。Postman 內建提供程式範例片段。只要透過點擊的方式，就會自動增加至腳本檔編輯區。

Test scripts are written in JavaScript, and are run after the response is received. >
[Learn more about tests scripts](#)

SNIPPETS

Send a request

Status code: Code is 200

Response body: Contains string

Response body: JSON value check

Response body: Is equal to a string

Response headers: Content-Type header check

Response time is less than 200ms

Status code: Successful POST request

Status code: Code name has string

Response body: Convert XML body to a JSON Object

Use Tiny Validator for JSON data

我們可以選擇 **Status code: Code is 200**，點擊後再編輯狀態碼為 **201**。

```
POST {{vcsa_url}}/api/session

Params Auth Headers (9) Body Pre-req. Tests Settings

1 let response = pm.response.json()
2   sessionId = response
3
4 pm.collectionVariables.set("vcsa_session", sessionId);
5
6 pm.test("Status code is 201", function () {
7   pm.response.to.have.status(201);
8 });
```

儲存變更後，重新傳送請求測試。在請求回應的 **Test Results** 選項欄位，可以看到顯示 **PASS** Status code is 201，這是 Postman 自動測試的結果，可以協助我們判斷回應是否正確。



Console 中顯示訊息

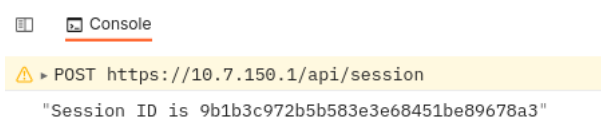
若想在 Console 中顯示訊息，可以在腳本檔中使用 `console.log()`。將以下程式加入至原有 Test 腳本檔重新測試。

```
console.log("Session ID is " + sessionId)
```

```
Params Auth Headers (9) Body Pre-req. Tests Settings

1 let response = pm.response.json()
2   sessionId = response
3
4 pm.collectionVariables.set("vcsa_session", sessionId);
5
6 pm.test("Status code is 201", function () {
7   pm.response.to.have.status(201);
8 });
9
10 console.log("Session ID is " + sessionId)
```

點選下方 **Console** 開啟紀錄，可看到剛剛寫入的紀錄訊息。



待續...

接著研究其他 API 操作應用。

- **Part 2** (https://hackmd.io/@farmer87/postman_vsphere_02).

附錄

安裝 Linux 桌面版 Postman

目前 Postman 支援以下 Linux 發行版本：

- Ubuntu 14.04 and newer
- Fedora 24
- Debian 8 and newer

目前安裝至 Fedora 35 可正常使用。

安裝步驟

1. 至[官網](https://www.postman.com/downloads/) (<https://www.postman.com/downloads/>) 下載最新版本 Postman。
2. 將壓縮檔解開至指定目錄 `opt`。

```
$ sudo tar -C /opt -xvzf Postman-linux-x86_64-9.1.5.tar.gz
```

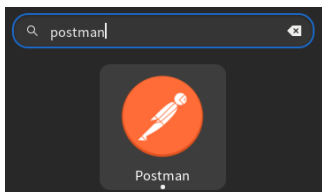
3. 若想要從起動器圖示開啟 Postman，請根據以下內容建立 `~/.local/share/applications/Postman.desktop` 檔案。

```
[Desktop Entry]
Encoding=UTF-8
Name=Postman
Exec=/opt/Postman/app/Postman %U
Icon=/opt/Postman/app/resources/app/assets/icon.png
Terminal=false
Type=Application
Categories=Development;
```

若要從終端機啟用，可建立鏈結。

```
$ sudo ln -s /opt/Postman/Postman /usr/bin/postman
```

4. 開啟 Postman 應用程式，使用註冊帳戶登入。



安裝 Postman Desktop Agent for Linux 的方式相同

建立工作區

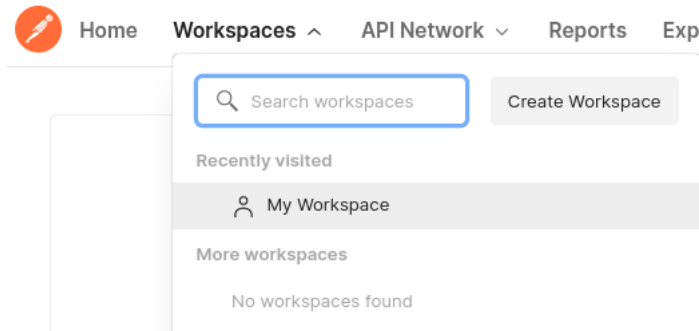
Postman 的工作區允許將工作與團隊組織起來進行協作。可將使用的項目（element）組合在一起，工作區可作為相關 API、集合、環境、模擬、監視器和其他鏈接實體的單一來源。

源。通過在工作區中進行協作，個人的編輯會與團隊即時同步。

Postman 註冊帳戶可建立無限的個人工作區。

執行步驟

1. 在 Postman 首頁的**工作區 (Workspaces)** 點選下拉選單，點擊 **Create Workspace**。



2. 根據需求輸入工作區名稱或摘要，在可視性 (Visibility) 選擇 **Personal**，點擊 **Create Workspace** 建立**個人工作區**。

免費註冊版本並不提供 **Private** 工作區。

Create workspace

Name

Summary

Add a brief summary about this workspace.

Visibility

Determines who can access this workspace.

☒ **Personal**
Only you can access

☐ Private
Only invited team members can access

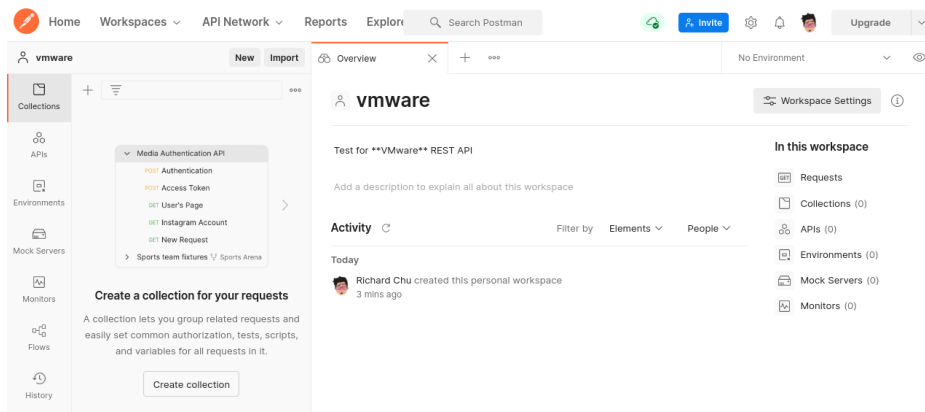
☐ Team
All team members can access

☐ Public
Everyone can view

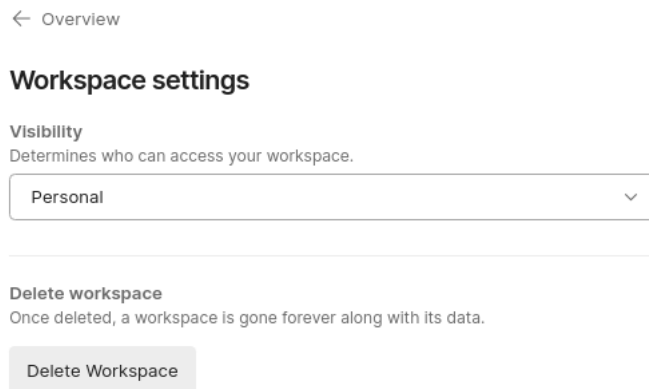
刪除工作區

執行步驟

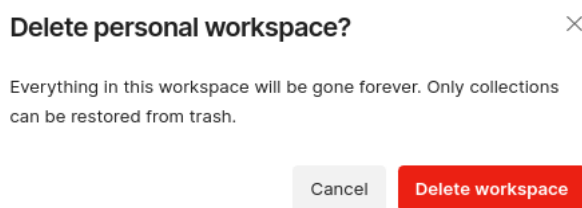
1. 選擇並進入要刪除的工作區後，在該工作區概覽 (Overview) 選項右側部份，點擊 **Workspace Settings** 進入設定。



2. 點擊 **Delete Workspace** 進行刪除工作區。



3. 點選 **Delete Workspace** 確認刪除動作。



移動集合至不同工作區

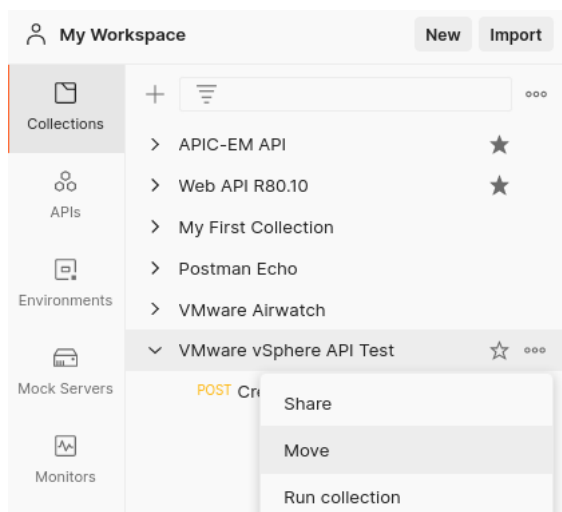
工作區可以包含集合、API、環境、模擬服務器和監視器等元素。特定元素只能存在於單個工作區中。如果需要，可將元素移動到不同的工作區。

無法移動監視器：如果移動正在監視的集合，則關聯的監視器將暫停並保留在原始工作區中。

執行步驟

以移動集合（Collections）作為範例。

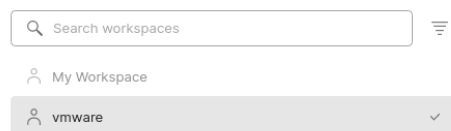
1. 從來源工作區 **My Workspace** 中選擇要移動的集合 **VMware vSphere API Test**，點擊其右側的 **...** 並選擇 **Move**。



2. 選擇要移動至的目標工作區 **vmware**，點選 **Move Collection** 遷移。

Move VMware vSphere API Test

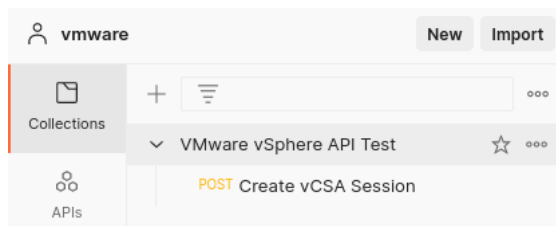
Monitors of this collection will pause and stay in the current workspace.



Cancel Move Collection

注意 無法將元素從私人、團隊或公共工作區移動到個人工作區。

3. 回到目標工作區確認移動結果。



工作區其他資訊

有關工作區相關資訊請參考**官網說明**：

- **建立工作區** (<https://learning.postman.com/docs/collaborating-in-postman/using-workspaces/creating-workspaces/>).
- **使用及管理工作區** (<https://learning.postman.com/docs/collaborating-in-postman/using-workspaces/managing-workspaces/>).

- **工作區角色** (<https://learning.postman.com/docs/collaborating-in-postman/using-workspaces/managing-workspaces/#managing-workspace-roles>).
- **Public Workspaces** (<https://learning.postman.com/docs/collaborating-in-postman/public-workspaces/>).

Desktop Agent 問題排除

發生以下問題，大多都是採用 Postman 網頁版。預設 Postman Agent 採用 **Auto-select** 選項，Postman 會自動選擇最佳的 Agent 來發送請求。

Cloud Agent Error: Can not send requests to reserved address. Make sure address is publicly accessible or select a different agent. [Use Postman's Desktop Agent](#)

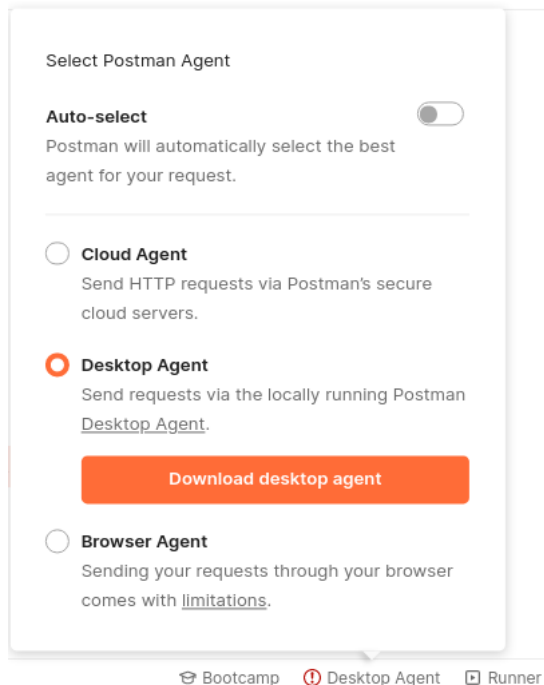
CORS Error: The request has been blocked because of the CORS policy [Use Postman's Desktop Agent](#)

根據顯示的錯誤訊息，Postman 告知使用 Desktop Agent 來執行請求。

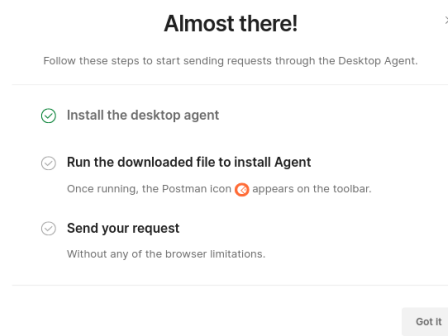
在 Postman 主頁右下方底部，**Auto-select agent** 顯示目前 Agent 的設定。

🔖 Bootcamp ☒ Auto-select agent ☐ Runner ☐ Trash ☐ ?

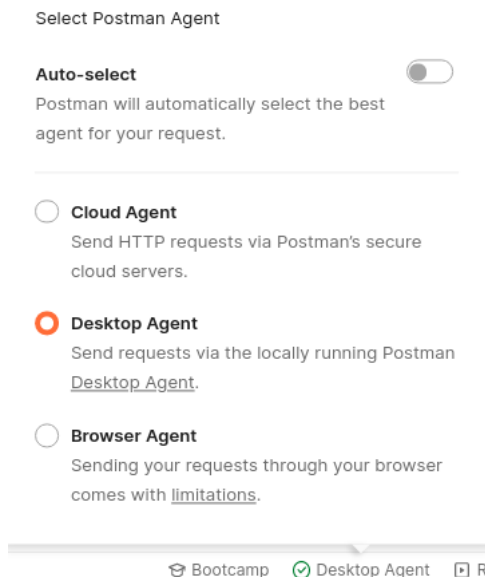
點選 **Auto-select agent** 便可展開 Postman Agent 選項。上述錯誤訊息告知，建議使用 **Desktop Agent**。所以請勾選 **Desktop Agent** 使用。



若先前沒有下載 Agent，點擊 **Download desktop agent** 下載並安裝。按照下列步驟重新執行請求。



以下是安裝並執行正常的顯示狀態。

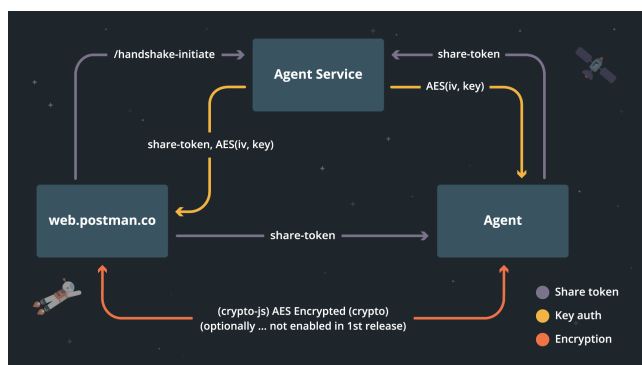


注意 使用網路版 Postman，且指定使用 Desktop Agent 的話，每次執行請求前，務必都要確認 Postman Agent 運行狀態。

詳細內容請參考 **[[Introducing the Postman Agent: Send API Requests from Your Browser without Limits](https://blog.postman.com/introducing-the-postman-agent-send-api-requests-from-your-browser-without-limits/)]**

(<https://blog.postman.com/introducing-the-postman-agent-send-api-requests-from-your-browser-without-limits/>).

o



參考

- **[Postman - Learning Center](https://learning.postman.com/docs/getting-started/introduction/)** (<https://learning.postman.com/docs/getting-started/introduction/>).

- **[Using Postman with the vCenter API](https://www.vgemba.net/vmware/VCSA-API-Postman/)**

(<https://www.vgemba.net/vmware/VCSA-API-Postman/>).

- 看來這位 Colin 也跟我遇到當初的問題，這邊使用 API Key 的方式，應該跟直接加在 Headers 的方法同義吧？後續再測試看看。這個方式應為**[正解]**

(<https://learning.postman.com/docs/sending-requests/authorization/#api-key>),

- 不過現在還沒想到要如何解決 Session ID 逾時的問題。還是每次提出請求都要將 Session ID 刪除？
- vSphere 5.5 之後預設會話超時為 120 分鐘，修改方式請參考

- **KB#2040626**(<https://kb.vmware.com/s/article/2040626>).

- **Session Timeout for Web Client**

([https://blog.ukotic.net/2021/04/20/session-timeout-now-in-the-vsphere-client-ui-vsphere-](https://blog.ukotic.net/2021/04/20/session-timeout-now-in-the-vsphere-client-ui-vsphere-7-update-2/)

[7-update-2/](https://blog.ukotic.net/2021/04/20/session-timeout-now-in-the-vsphere-client-ui-vsphere-7-update-2/)).

- **vCenter API from Postman**

(<https://www.postman.com/fabxoe/workspace/vcenter-api/overview>).

- **vSphere Automation REST Resources from Postman**(<https://www.postman.com/fabxoe/workspace/vcenter-api/collection/9432753-a70102d0-9f32-4164-bbab-971b4d499fa3>).