

# API 程式設計指南

2022 年 3 月 30 日  
vRealize Operations 8.4

您可以在 VMware 網站上找到最新的技術文件，網址如下：

<https://docs.vmware.com/tw/>

**VMware, Inc.**  
3401 Hillview Ave.  
Palo Alto, CA 94304  
[www.vmware.com](http://www.vmware.com)

Copyright © 2022 VMware, Inc. 保留所有權利。 [版權與商標資訊](#)。

# 目錄

- 1 關於此程式設計指南** 4
  
- 2 瞭解 vRealize Operations Manager API** 5
  - vRealize Operations Manager API 的運作方式 5
  - 用戶端工作流程概觀 7
    - vRealize Operations Manager API REST 要求 8
    - vRealize Operations Manager API REST 回應 9
  - 將 API 與 vRealize Operations Manager 搭配使用 10
  
- 3 開始使用 API** 12
  - 取得驗證 Token 12
  - 尋找介面卡類型和物件類型 14
  - 產生物件所有度量的清單 15
  
- 4 設定介面卡執行個體** 18
  - 設定介面卡執行個體要求的摘要 18
  - 識別解決方案及其介面卡 19
  - 識別介面卡所需的物件類型 20
  - 建立介面卡執行個體 22
  - 提供憑證有效的證明 25
  - 啟動監控新的介面卡執行個體 29

# 關於此程式設計指南

# 1

vRealize Operations Manager API 程式設計指南提供 vRealize Operations Manager REST API 的相關資訊，其中包括如何使用 REST API 資源、驗證和建構 REST API 呼叫。

## 預定對象

此資訊適用於想要使用 vRealize Operations Manager REST API 以程式設計方式設定和管理 vRealize Operations Manager 的管理員和程式設計師。本指南著重於常見的使用案例。

# 瞭解 vRealize Operations Manager API

# 2

開發人員可以使用 API 建立 vRealize Operations Manager 的互動式用戶端。API 遵循 REST 樣式，並可供所有授權使用者使用。

vRealize Operations Manager 用戶端透過 HTTP 與伺服器進行通訊，交換 vRealize Operations Manager 物件的表示方式。這些表示方式採用 JSON 或 XML 元素的形式。您可以使用 HTTP GET 要求來擷取物件的目前表示方式、HTTP POST 和 PUT 要求以建立或修改物件，以及 HTTP DELETE 要求以刪除物件。

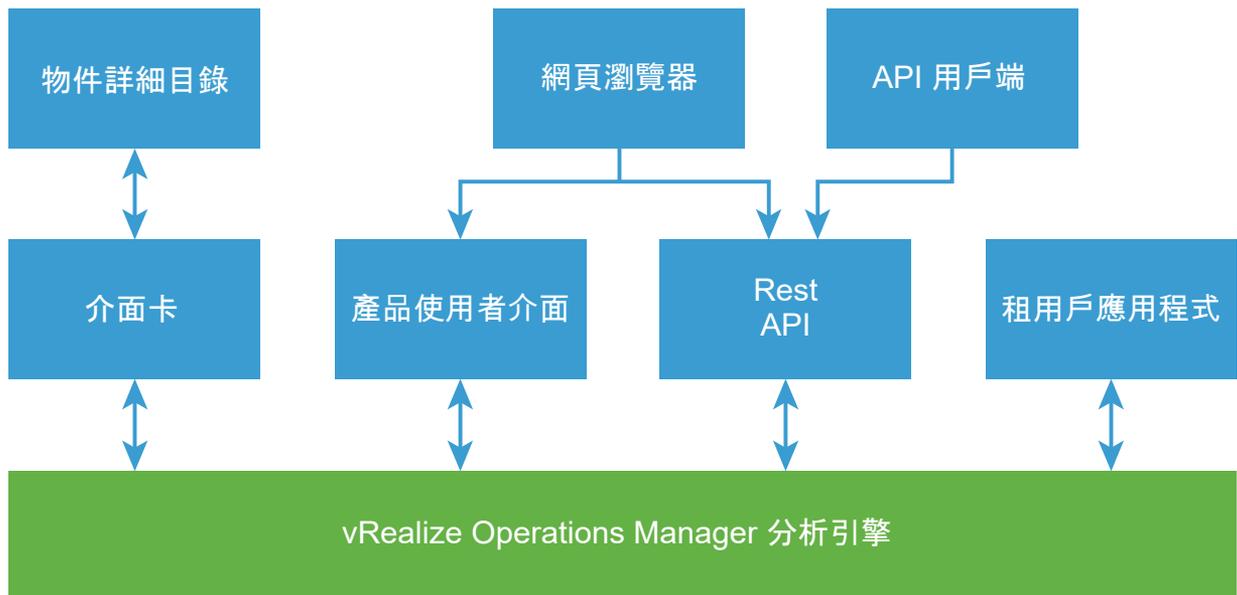
本章節討論下列主題：

- [vRealize Operations Manager API 的運作方式](#)
- [用戶端工作流程概觀](#)
- [將 API 與 vRealize Operations Manager 搭配使用](#)

## vRealize Operations Manager API 的運作方式

請使用網頁瀏覽器，透過產品使用者介面或 API 呼叫與 vRealize Operations Manager 分析引擎進行通訊。

圖 2-1. vRealize Operations Manager 簡化的架構



介面卡執行個體會從受監控環境中的物件收集資料。vRealize Operations Manager 分析引擎會處理資料，並在圖形化介面中顯示完整模型。

## 為何使用 API

需要將定義完善的工作流程自動化時 (例如重複相同的工作，以為新的 vRealize Operations Manager 使用者設定存取控制)，API 最為有用。在 vRealize Operations Manager 資料存放庫上執行查詢時 (例如在虛擬環境中擷取特定資產的資料時)，API 也很有用。此外，您可以使用 API 從 vRealize Operations Manager 資料存放庫中擷取所有資料，並將其載入到單獨的分析系統中。

## vRealize Operations Manager 術語

用來說明介面卡物件的 XML 語法與 API 代碼語法相對應，但與您在使用者介面中找到的內容不同。下列術語會顯示在使用者介面中。包含在每個術語的說明中的，是 API 呼叫中使用的對應 XML 語法。

介面卡類型	定義用於探索特定物件類型的介面卡。例如，vCenter 介面卡會探索連線至 vSphere 資料中心的物件。AWS 介面卡會探索 AWS 服務和物件。 XML 語法： <code>adapterkinds</code> 。
物件類型	代表物件或資訊來源的實體類別。物件會將資料報告到 vRealize Operations Manager 分析引擎。虛擬機器、資料存放區和主機系統是 vCenter 介面卡型號中所定義物件類型的範例。 XML 語法： <code>resourcekinds</code> 。

## 用戶端工作流程概觀

vRealize Operations Manager API 用戶端會執行 REST 工作流程、向伺服器發出 HTTP 要求，以及從伺服器的回應中擷取所需的資訊。

### 關於 REST

REST 是 Representational State Transfer 的縮寫，說明使用超文字傳輸通訊協定 (HTTP) 來交換用戶端與伺服器之間物件序列化表示的程式架構樣式特性。在 vRealize Operations Manager API 中，這些表示方式為 JSON 或 XML 文件。

在 REST 工作流程中，物件的表示方式會在用戶端與伺服器之間來回傳遞，並明確假設沒有任何一方需要瞭解單一要求或回應中所顯示物件以外的任何物件。包含 URL 之要求或回應的生命週期過後，提供這些文件的 URL 通常仍會存在。

### REST API 工作流程

寫來使用 REST API 的應用程式會使用 HTTP 要求，該要求通常由指令碼或其他較高階語言執行遠端程序呼叫，以建立、擷取、更新或刪除 API 定義的物件。在 vRealize Operations Manager REST API 中，這些物件是由 XML 配置的集合來定義。作業本身為 HTTP 要求，因此對所有 HTTP 用戶端來說是通用的。

若要寫入 REST API 用戶端應用程式，您僅需瞭解 HTTP 通訊協定，以及 JSON 或 XML 的語義，也就是 vRealize Operations Manager API 使用的傳輸格式。若要在此類用戶端中有效使用 API，您必須熟悉下列概念。

- API 支援的物件集及其表示的內容。
- API 表示這些物件的方式。
- 用戶端如何參照其想要運作的物件。

包含 API 要求完整清單的 API 參考。請參閱 [關於配置參考](#)。

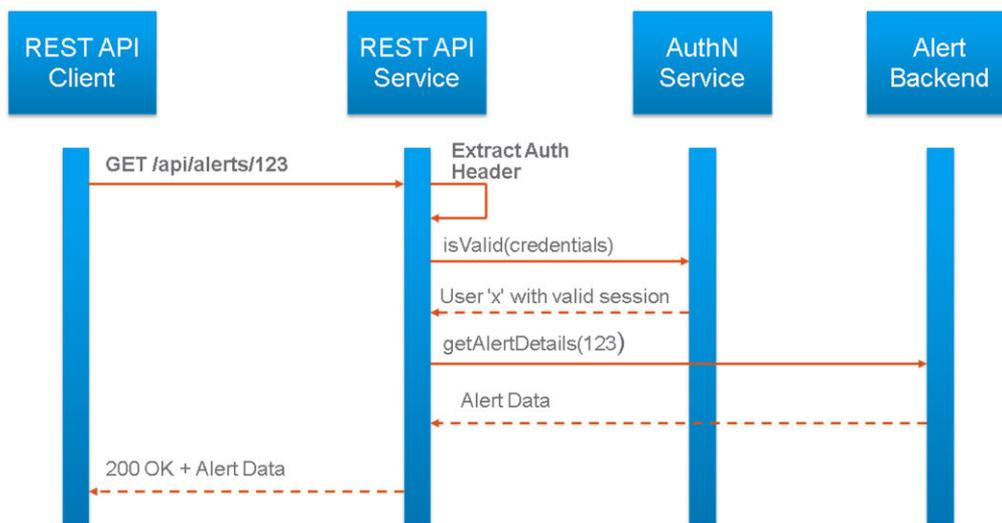
## vRealize Operations Manager API REST 要求

若要擷取物件表示方式，用戶端會向物件參考發出 HTTP 要求。

### 安全性

API 用戶端與伺服器之間的 HTTP 連結是使用 SSL 建立的。API 用戶端會設定以 Token 為基礎的驗證，以與伺服器進行通訊。

圖 2-2. 案例：為使用者提供認證，以取得有關識別碼 123 警示的詳細資料



使用以 Token 為基礎的驗證 POST 登入要求至 vRealize Operations Manager API 伺服器，提供有效的使用者認證以取得驗證 Token。下列範例提供以 Token 為基礎的驗證案例。

- 1 取得 vRealize Operations Manager 執行個體的有效使用者認證。
- 2 向 REST 端點 POST 要求以進行驗證。

```
https://RESTendpoint.example.com/suite-api/api/auth/token/acquire
```

要求內文包括使用者名稱、密碼和驗證來源。

- 3 在回應內文中，端點會傳回 Token、到期日和時間。

- 4 如需進一步的通訊，請在授權標頭中包含具有下列格式的 Token 物件：

```
Authorization: vRealizeOpsToken <vROps_token>
```

或者，如果您從 SSO 來源取得 Token，則授權標頭的格式為：

```
Authorization: SSO2Token <SSO_SAML_TOKEN>
```

- 5 您可以將 POST 要求傳送至登出端點，以在到期日和時間之前使 Token 失效。

```
POST https://RESTendpoint.example.com/suite-api/api/auth/token/release
```

## 要求標頭

API 要求中通常包含下列 HTTP 標頭：

Accept-Language 若要指定回應中所需的語言，請使用 `Accept-Language` 要求標頭。 `ErrorType` 回應中的訊息字串已進行當地語系化。若要要求使用當地語系化為法文的訊息字串來回應，請使用下列標頭：

```
Accept-Language: fr-FR
```

授權 所有建立 API 工作階段的要求，必須包含由組織使用的身分識別提供者所規定的 `Authorization` 標頭格式

Content-Type 包含內文的要求必須包含適當的 HTTP `Content-Type` 標頭。

- 對於以 XML 撰寫的要求內文，標頭必須包含 `Content-Type: application/xml`
- 對於以 JSON 撰寫的要求內文，標頭必須包含 `Content-Type: application/json`

接受 若要指定所需的回應格式，請包括 `Accept` 要求標頭。

- 對於以 XML 撰寫的回應，標頭必須包含 `Accept: application/xml`
- 對於以 JSON 撰寫的回應，標頭必須包含 `Accept: application/json`

## 以 XML 撰寫的要求內文

對於以 XML 撰寫的要求內文，vRealize Operations Manager 會使用驗證 XML 剖析器，該剖析器需要在要求內文中的元素符合配置的順序和編號。除非符合下列準則，否則要求內文會因被視為無效而遭到拒絕：

- 必須為要求中的元素所表示的所有命名空間提供 XML 命名空間屬性。
- 如果要求中表示了多個命名空間，則 XML 命名空間屬性必須包含識別前置詞，且該前置詞必須與該命名空間中的所有元素搭配使用。
- 所有所需元素必須出現在要求內文中。出現在要求內文中的所有元素都必須依照配置建立的順序出現，並顯示符合配置所指定類型限制的內容。

## vRealize Operations Manager API REST 回應

所有回應均包含 HTTP 狀態碼和 (除非狀態碼為 204 (無內容)) 接受標頭。回應內容取決於要求。某些回應包含文件內文，有些僅包含 URL，而有些則為空白。

## HTTP 回應代碼

API 用戶端可以在回應中預期看到 HTTP 狀態碼的子集。

表 2-1. API 傳回的 HTTP 狀態碼

狀態碼	狀態說明
[200 完成]	要求有效且已完成。此回應包含文件內文。
[201 已建立]	要求有效。已建立要求的物件，並且可在位置標頭中指定的 URL 找到。
[202 已接受]	要求有效，且已建立工作以進行處理。此回應通常會與 <code>TaskStatus</code> 元素一起顯示。
[204 無內容]	要求有效且已完成。此回應不包含內文。
[400 要求不正確]	要求內文格式錯誤、不完整或無效。
[401 未經授權]	登入失敗或驗證 Token 已到期。
[403 禁止]	使用者未經驗證，或者沒有足夠的權限可存取要求中指定的一或多個物件。
[404 找不到]	找不到要求中指定的物件。
[405 方法不被允許]	此物件不支援要求中指定的 HTTP 方法。
[406 無法接受]	要求所識別的資源無法產生要求的 <code>Accept</code> 標頭中所指定類型的回應。
[415 不支援的媒體類型]	要求所識別的資源不支援指定之 <code>Content-Type</code> 和 HTTP 方法的要求。
[422 找不到]	通常表示要求 URL 或要求內文格式錯誤。
[429 過多的要求]	用戶端已傳送過多的要求，或者多個用戶端傳送過多同時的要求，且伺服器受到速率限制而無法處理這些要求。 若要解決此問題，請稍後嘗試再次傳送要求。
[500 內部伺服器錯誤]	已收到要求，但因為伺服器上發生內部錯誤而無法完成。
[503 服務無法使用]	伺服器因為發生暫時的情況 (例如資源用盡或伺服器維護)，目前無法處理要求。
[504 閘道逾時]	作為閘道或 Proxy 的伺服器未收到來自要求 URL 所指定上游伺服器的及時回應。

## 將 API 與 vRealize Operations Manager 搭配使用

您可以使用瀏覽器或 HTTP 用戶端程式來傳送要求並接收回應。

### REST 用戶端程式

任何可以傳送 HTTPS 要求的用戶端應用程式，都是使用 vRealize Operations Manager API 開發 REST 應用程式的適當工具。REST 用戶端外掛程式適用於大多數瀏覽器和許多 IDE。通常會使用下列開放原始碼程式：

- cURL。 <http://curl.haxx.se>
- Postman 應用程式。 <http://www.getpostman.com>

此外，VMware 為 vRealize Operations Manager API 提供特定語言的用戶端繫結。請參閱 [關於配置參考](#)。

## 關於配置參考

vRealize Operations Manager REST API 說明文件包含 vRealize Operations Manager API 中所有元素、類型、查詢和作業的參考資料。其中也包含配置定義檔案。

Swagger 型 API 說明文件隨附於產品中，具備直接從登陸頁面進行 REST API 通話的功能。

若要存取 API 說明文件，請使用 vRealize Operations Manager 執行個體的 URL。例如，如果執行個體的 URL 是 `https://vrealize.example.com`，可從下列位置取得 API 參考：`https://vrealize.example.com/suite-api/doc/swagger-ui.html`。

特定語言的用戶端繫結位於：

```
https://vrealize.example.com/suite-api/
```

## 關於 vRealize Operations Manager API 範例

所有範例包括 HTTP 要求和回應。這些範例會顯示與作業相關聯的工作流程和內容，例如建立和查詢受監控環境中物件的相關資訊。

範例要求內文以 JSON 撰寫。範例要求中包含 vRealize Operations Manager API 所需的要求標頭，這些並非較大範例的片段。

大多數的範例回應僅顯示與所討論作業相關的元素和屬性。省略符號 (...) 代表回應內文中省略的內容。

# 開始使用 API

# 3

API 用戶端和 vRealize Operations Manager 伺服器透過 HTTPS 進行通訊，交換 API 物件的 XML 表示方式。

此簡單的 REST API 工作流程範例，顯示了如何取得 VMware vCenter® 介面卡包含在型號定義中之虛擬機器物件類型所有度量的清單。您可以使用 API 取得任何物件類型可用度量的完整清單。

本章節討論下列主題：

- 取得驗證 Token
- 尋找介面卡類型和物件類型
- 產生物件所有度量的清單

## 取得驗證 Token

vRealize Operations Manager 需要驗證 API 要求。此工作流程的第一步驟是取得驗證 Token。

若要取得驗證 Token，登入要求會以基本 HTTP 驗證所需的格式提供使用者認證。在此範例中，使用者使用 URL `https://vrealize.example.com/` 登入 vRealize Operations Manager 執行個體。

---

**備註** 此範例使用以 Token 為基礎的驗證。如需關於驗證機制的詳細資訊，請參閱 [安全性](#)。

---

透過使用 `authSource`，可以匯入並驗證位於其他機器上的使用者和使用者群組資訊。例如，可以從 LDAP、Active Directory、VMware Identity Manager、Single Sign-on 等驗證使用者。當您匯入存放於另一台機器的使用者帳戶資訊時，必須定義用於從來源機器匯入使用者帳戶的準則。

建立驗證來源後，您可以使用該驗證來源透過指定名稱取得 Token。預設驗證來源類型為「本機」。

必要條件

- 確保網頁瀏覽器與 vRealize Operations Manager 伺服器之間有安全的通道。開啟瀏覽器並輸入 vRealize Operations Manager 執行個體的 URL，例如：

```
https://vrealize.example.com
```

系統會警告您的連線並非私人連線。按一下以確認安全性例外狀況，然後建立 SSL 交握。

- 確認您可以存取 API。輸入 vRealize Operations Manager 執行個體的 URL，並將 `suite-api/docs/rest/index.html` 新增至尾端，例如：

```
https://vrealize.example.com/suite-api/docs/rest/index.html
```

- 確認您有 vRealize Operations Manager 執行個體使用者的登入認證。

## 程序

- 1 POST 要求至登入 URL 以取得 Token。

```
POST https://vrealize.example.com/suite-api/api/auth/token/acquire
```

請參閱 [登入要求和回應](#)。

- 2 檢查回應。

成功的要求會傳回 ops 授權 Token，您必須在後續的 API 要求中將其包含在內。

## 範例：登入要求和回應

此範例顯示登入使用者名稱為：`vRealize-user` 和密碼為：`vRealize-dummy-password` 之使用者的要求和回應。

要求標頭：

```
POST https://vrealize.example.com/suite-api/api/auth/token/acquire
Content-Type: application/json
Accept: application/json
```

以 JSON 格式撰寫的要求內文：

```
{
  "username" : "vRealize-user",
  "password" : "vRealize-dummy-password"
}
```

以 JSON 撰寫的回應：

```
200 OK

{
  "token": "8f868cca-27cc-43d6-a838-c5467e73ec45::77cea9b2-1e87-490e-b626-e878beea23b",
  "validity": 1470421325035,
  "expiresAt": "Friday, August 5, 2016 6:22:05 PM UTC",
  "roles": []
}
```

回應代碼表示要求是否成功，或其失敗原因。

- 如果要求成功，伺服器會傳回 HTTP 回應代碼 200 (OK)，以及在六個小時後到期並可重複使用的 ops 授權 Token。此 Token 必須包含在每個後續的 API 要求中。

- 如果要求的授權標頭遺失，則伺服器會傳回 HTTP 回應代碼 403。
- 如果授權標頭中提供的認證無效，則伺服器會傳回 HTTP 回應代碼 401。

## 尋找介面卡類型和物件類型

vRealize Operations Manager 執行個體包含多個介面卡類型。若要尋找 vCenter 介面卡的介面卡類型，請建立 GET 要求以擷取所有介面卡類型的清單。API 回應包含介面卡監控的所有物件類型。

### 必要條件

確認您已登入 vRealize Operations Manager 執行個體。

### 程序

- 1 針對所有介面卡類型提出 GET 要求。

```
GET https://vrealize.example.com/suite-api/api/adapterkinds
```

- 2 檢查回應以尋找 vCenter 介面卡和受監控的物件類型清單。

請參閱 [判定 vCenter 介面卡的介面卡類型和物件類型](#) 的回應部分。

## 範例：判定 vCenter 介面卡的介面卡類型和物件類型

此範例會尋找 vCenter 介面卡的介面卡類型，以及介面卡型號定義中包含的所有物件類型。

要求標頭：

```
GET https://vRealize.example.com/suite-api/api/adapterkinds
Content-Type: application/json
Authorization: vRealizeOpsToken <vROps_token>
Accept: application/json
```

其中 *vROps\_token* 是您從 [取得驗證 Token](#) 中的回應取得的 Token。

vCenter 介面卡以 JSON 撰寫回應的程式碼片段：

```
200 OK

{
  "key": "VMWARE",
  "name": "vCenter Adapter",
  "description": "Provides the connection information and credentials required...",
  "adapterKindType": "GENERAL",
  "describeVersion": 573,
  "identifiers": [],
  "resourceKinds": [
    "ClusterComputeResource",
    "ComputeResource",
    "CustomDatacenter",
    "Datacenter",
    "Datastore",
    "StoragePod",
```

```

    "DatastoreFolder",
    "VM Entity Status",
    "Folder",
    "HostFolder",
    "HostSystem",
    "NetworkFolder",
    "ResourcePool",
    "VMwareAdapter Instance",
    "VirtualMachine",
    "VMFolder",
    "DistributedVirtualPortgroup",
    "VmwareDistributedVirtualSwitch",
    "vSphere World"
  ],
  ...
}

```

針對 vCenter 介面卡，adapter-kind 金鑰為 VMWARE。resourceKinds 是 vCenter 介面卡監控的物件類型。針對虛擬機器物件類型，resourceKinds 為 VirtualMachine。

## 產生物件所有度量的清單

若要為 vCenter 介面卡型號中定義的任何虛擬機器產生度量的完整清單，請使用介面卡類型和物件類型對 URL 提出 GET 要求。

### 必要條件

確認已滿足下列需求：

- 您已登入 vRealize Operations Manager 執行個體。
- 您知道 vCenter 介面卡的 adapterKind 值，以及虛擬機器的 resourceKinds 值。請參閱判定 [vCenter 介面卡的介面卡類型和物件類型](#)

### 程序

- 1 提出 GET 要求以取得度量的中繼資料。

```

GET https://vrealize.example.com/suite-api/api/adapterkinds/VMWARE/resourcekinds/
VirtualMachine/statkeys

```

- 2 將回應中列出的度量與使用者介面中顯示的度量進行比較。請參閱 [API 和使用者介面中的虛擬機器度量](#)

### 範例：API 和使用者介面中的虛擬機器度量

此範例顯示在 XML 回應中列出的虛擬機器度量如何與 vRealize Operations Manager 使用者介面中顯示的度量相比較。

## 要求：

```
GET https://vrealize.example.com/suite-api/api/adapterkinds/VMWARE/resourcekinds/
VirtualMachine/statkeys
Content-Type: application/json
Authorization: vRealizeOpsToken <vROps_token>
Accept: application/json
```

## 其中：

- VMWARE 是 adapterKindKey。
- VirtualMachine 是 resourceKindKey。
- *vROps\_token* 是您從取得驗證 Token

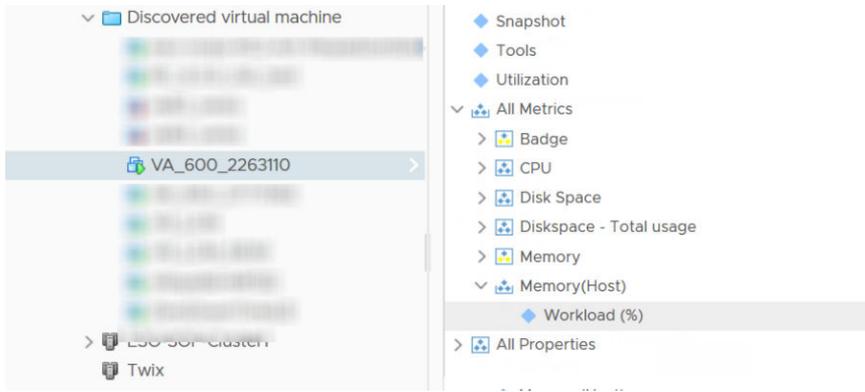
以 JSON 撰寫回應的程式碼片段：

```
200 OK
```

```
{
  "resourceTypeAttributes": [
    ...
    {
      "key": "mem|host_workload",
      "name": "Memory|Host Workload",
      "description": "Host Workload (%)",
      "defaultMonitored": false,
      "rollupType": "AVG",
      "instanceType": "INSTANCED",
      "unit": "%",
      "dataType2": "FLOAT",
      "monitoring": false,
      "property": false
    },
    ...
  ]
}
```

回應中的每個 `resourceTypeAttribute` 是具有虛擬機器物件之中繼資料的度量。name 會對應 vRealize Operations Manager 使用者介面中顯示的文字。在此範例中，程式碼片段會列出記憶體和主機工作負載的度量。

若要將回應中的度量與使用者介面中的度量進行比較，請登入 `vrealize.example.com` 上執行的 vRealize Operations Manager 執行個體，並導覽至虛擬機器的度量。下列範例顯示找到記憶體 (主機) 和工作負載度量的位置。



此範例顯示如何擷取虛擬機器物件類型的度量。若要擷取其他物件類型的度量，請將 GET 要求中的 `VirtualMachine` 取代為其他 `resourceKinds`。

## 設定介面卡執行個體

安裝包含管理套件和介面卡的解決方案後，您必須設定介面卡執行個體，以從介面卡型號定義中的物件收集資料。您可以使用 vRealize Operations Manager API 來設定介面卡執行個體。

此使用案例範例顯示如何設定 vSphere 解決方案的介面卡執行個體，包括：

- 含每個操作各自之特定程序要求、要求內文和回應的
- 作業摘要

本章節討論下列主題：

- [設定介面卡執行個體要求的摘要](#)
- [識別解決方案及其介面卡](#)
- [識別介面卡所需的物件類型](#)
- [建立介面卡執行個體](#)
- [提供憑證有效的證明](#)
- [啟動監控新的介面卡執行個體](#)

### 設定介面卡執行個體要求的摘要

您可以建立有順序的 API 要求以設定介面卡執行個體。來自先前要求的回應會產生後續要求所需的資訊。

表 4-1. 要求的摘要

作業	要求	要求內文	回應
取得所有已向產品登錄的解決方案，並識別介面卡類型	GET <API-URL>/suite-api/api/solutions	無	adapterkindkeys
取得特定介面卡類型的所有物件類型。	GET <API-URL>/suite-api/api/adapterkinds/{key}/resourcekinds	無	resourceIdentifierTypes
建立介面卡執行個體物件	POST <API-URL>/suite-api/api/adapters	resourceIdentifiers 和 credential 的值	adapterid

表 4-1. 要求的摘要 (續)

作業	要求	要求內文	回應
修補介面卡執行個體以確認所提供的憑證	PATCH <API-URL>/suite-api/api/adapters	POST <API-URL>/suite-api/api/adapters 的 API 回應	POST <API-URL>/suite-api/api/adapters 的 API 回應，不含 adapter-certificates
啟動介面卡監控	PUT <API-URL>/suite-api/api/adapters/{adapterid}/monitoringstate/start	無	200 OK

## 識別解決方案及其介面卡

vRealize Operations Manager 執行個體可能已安裝多個解決方案。若要尋找 vSphere 解決方案及其介面卡類型，請建立 GET 要求以擷取所有解決方案的清單。此回應包括解決方案中隨附的所有介面卡。

在此範例使用案例中，vRealize Operations Manager 執行個體的 API-URL 為 `vrealize.example.com`。

### 必要條件

確認您可以登入 vRealize Operations Manager 執行個體的 API URL。請參閱 [取得驗證 Token](#)。

### 程序

- 1 提出 GET 要求以列出所有解決方案。

```
GET https://vrealize.example.com/suite-api/api/solutions
```

- 2 檢查回應以尋找 vSphere 解決方案及其介面卡類型。

請參閱 [vSphere 解決方案的介面卡類型](#) 的回應部分。

### 範例：vSphere 解決方案的介面卡類型

此範例會列出所有已安裝的解決方案，以及各自的介面卡類型。

要求標頭：

```
GET https://vrealize.example.com/suite-api/api/solutions
```

以 JSON 撰寫的回應：

```
200 OK
```

```
{
  "solution":
  {
    "id": "MPforLogInsight",
    "name": "VMware vRealize Operations Management Pack for Log Insight",
    "version": "6.0.3171089",
```

```

    "description": "VMware vRealize Operations Management Pack for Log Insight... ",
    "vendor": "VMware Inc.",
    "adapterKindKeys": [
      "LogInsightAdapter"
    ]
  },
  {
    "id": "ep-ops-os-and-availability",
    "name": "Operating Systems / Remote Service Monitoring",
    "version": "1.0.4071095",
    "description": "The End Point Operations Management Solution for Operating... ",
    "vendor": "VMware Inc.",
    "adapterKindKeys": [
      "ep-ops-os-and-availability-kind"
    ]
  },
  {
    "id": "VMware vSphere",
    "name": "VMware vSphere",
    "version": "6.0.7496664",
    "description": "Manages vSphere objects such as Clusters, Hosts...",
    "vendor": "VMware Inc.",
    "adapterKindKeys": [
      "VMWARE",
      "PythonRemediationVcenterAdapter"
    ]
  }
]
}

```

此回應顯示三個已安裝的解決方案：

- Log Insight 解決方案的管理套件
- 端點作業解決方案
- vSphere 解決方案

vSphere 解決方案有兩種介面卡類型：

- VMWARE
- PythonRemediationVcenterAdapter

針對 vCenter 介面卡，介面卡類型為 VMWARE。

## 識別介面卡所需的物件類型

決定要建立 vCenter 介面卡的執行個體後，您必須識別該介面卡所需的物件類型。建立 GET 要求，以擷取 vCenter 介面卡所有物件類型的清單。

### 必要條件

確認您知道 vCenter 介面卡的介面卡類型。

## 程序

- 1 建立 GET 要求，以列出 vCenter 介面卡的所有物件類型。

```
GET https://vrealize.example.com/suite-api/api/adapterkinds/VMWARE/resourcekinds
```

- 2 檢查回應以識別所需的物件類型。

請參閱 [vCenter 介面卡所需的物件類型](#) 的回應部分。

## 範例：vCenter 介面卡所需的物件類型

此範例會尋找 vCenter 介面卡的所有物件類型。

要求標頭：

```
GET https://vrealize.example.com/suite-api/api/adapterkinds/VMWARE/resourcekinds
```

以 JSON 撰寫回應的程式碼片段：

```
200 OK
```

```
{
  "key": "VmwareAdapter Instance",
  "name": "vCenter Server",
  "adapterKind": "VMWARE",
  "resourceKindType": "ADAPTER_INSTANCE",
  "resourceKindSubType": "NONE",
  "resourceIdentifierTypes": [
    {
      "name": "AUTODISCOVERY",
      "dataType": "STRING",
      "isPartOfUniqueness": false
    },
    {
      "name": "DISABLE_COMPUTATION_BASED_ON_CONSUMERS",
      "dataType": "STRING",
      "isPartOfUniqueness": false
    },
    {
      "name": "DV_PORT_GROUP_DISABLED",
      "dataType": "STRING",
      "isPartOfUniqueness": false
    },
    {
      "name": "DVS_DISABLED",
      "dataType": "STRING",
      "isPartOfUniqueness": false
    },
    {
      "name": "PROCESSCHANGEEVENTS",
      "dataType": "STRING",
      "isPartOfUniqueness": false
    },
  ],
}
```

```

    {
      "name": "VCURL",
      "dataType": "STRING",
      "isPartOfUniqueness": true
    },
    ...
    {
      "name": "VM_LIMIT",
      "dataType": "INTEGER",
      "isPartOfUniqueness": false
    }
  ],
  ...
}

```

此程式碼片段顯示屬性為 "resourceKindType": "ADAPTER\_INSTANCE" 的資源種類。資源識別碼為 "isPartOfUniqueness": true 的任何物件類型都需要具有 API 要求的該物件類型值，才能建立介面卡執行個體。

vCenter 介面卡的介面卡執行個體需要 VCURL 的值或 vCenter 的 URL。

## 建立介面卡執行個體

識別介面卡所需的物件類型之後，請為物件類型提供參數值以建立介面卡執行個體。您的 POST 要求包含具有所需參數的要求內文。

若要建立介面卡執行個體，VCURL 設定為必填。

### 必要條件

確認您擁有 vCenter 的 IP 位址和認證。

### 程序

- 1 提出 POST 要求，以建立介面卡執行個體。

```
POST https://vrealize.example.com/suite-api/api/adapters
```

- 2 檢查回應以尋找 vSphere 解決方案的名稱及其介面卡類型。

請參閱 [介面卡執行個體](#) 的回應部分。

## 範例：介面卡執行個體

此範例會使用下列參數建立 vCenter 的介面卡執行個體：

- 顯示名稱：VC 介面卡執行個體
- 說明：VC 12.345.678.9 的 vCenter 介面卡執行個體
- vCenter Server IP 位址：https://12.345.678.9
- 認證名稱：VC-Credential-1
- 使用者名稱：administrator@vsphere.local

- 密碼：VC-dummy-passwd

AUTODISCOVERY 和 PROCESSCHANGEEVENTS 為選用，但會包含在內，以在要求內文和回應中顯示資源識別碼的其他範例。

要求標頭：

```
POST https://vrealize.example.com/suite-api/api/adapters
```

以 JSON 格式撰寫的要求內文：

```
{
  "name" : "VC Adapter Instance",
  "description" : "A vCenter Adapter Instance for VC 12.345.678.9",
  "collectorId" : "1",
  "adapterKindKey" : "VMWARE",
  "resourceIdentifiers" : [
    {
      "name" : "AUTODISCOVERY",
      "value" : "true"
    },
    {
      "name" : "PROCESSCHANGEEVENTS",
      "value" : "true"
    },
    {
      "name" : "VCURL",
      "value" : "https://12.345.678.9"
    }
  ],
  "credential" : {
    "id" : null,
    "name" : "VC-Credential-1",
    "adapterKindKey" : "VMWARE",
    "credentialKindKey" : "PRINCIPALCREDENTIAL",
    "fields" : [
      {
        "name" : "USER",
        "value" : "administrator@vsphere.local"
      },
      {
        "name" : "PASSWORD",
        "value" : "VC-dummy-passwd"
      }
    ]
  }
}
```

以 JSON 撰寫回應的程式碼片段：

201 Created

```
{
  "resourceKey": {
    "name": "VC Adapter Instance",
    "adapterKindKey": "VMWARE",
    "resourceKindKey": "VMwareAdapter Instance",
    "resourceIdentifiers": [
      {
        "identifierType": {
          "name": "AUTODISCOVERY",
          "dataType": "STRING",
          "isPartOfUniqueness": false
        },
        "value": "true"
      },
      {
        "identifierType": {
          "name": "DISABLE_COMPUTATION_BASED_ON_CONSUMERS",
          "dataType": "STRING",
          "isPartOfUniqueness": false
        },
        "value": ""
      },
      {
        "identifierType": {
          "name": "DV_PORT_GROUP_DISABLED",
          "dataType": "STRING",
          "isPartOfUniqueness": false
        },
        "value": ""
      },
      {
        "identifierType": {
          "name": "DVS_DISABLED",
          "dataType": "STRING",
          "isPartOfUniqueness": false
        },
        "value": ""
      },
      {
        "identifierType": {
          "name": "PROCESSCHANGEEVENTS",
          "dataType": "STRING",
          "isPartOfUniqueness": false
        },
        "value": "true"
      },
      {
        "identifierType": {
          "name": "VCURL",
          "dataType": "STRING",
```

```

        "isPartOfUniqueness": true
    },
    "value": "https://12.345.678.9"
},
{
    "identifierType": {
        "name": "VM_FOLDER_DISABLED",
        "dataType": "STRING",
        "isPartOfUniqueness": false
    },
    "value": ""
},
{
    "identifierType": {
        "name": "VM_LIMIT",
        "dataType": "STRING",
        "isPartOfUniqueness": false
    },
    "value": ""
}
]
},
"description": "A vCenter Adapter Instance for VC 12.345.678.9",
"collectorId": 1,
"collectorGroupId": "909c2fbf-2c2c-4957-9a75-21bf2a887d31",
"credentialInstanceId": "65081a8d-d462-43b2-b4e0-596eaf3d497e",
"monitoringInterval": 5,
"adapter-certificates": [
    {
        "thumbprint": "2520fb4351bc91ee7b82ef7cc54a8d88fa893da9",
        "certificateDetails": "[
            Version: V3 Subject: C=US, CN=12.345.678.9
            Signature Algorithm: SHA256withRSA, OID = 1.2.840.113549.1.1.11
            Key: Sun RSA public key, 2048 bits modulus: ...
            Validity: [From: Wed Jul 15 19:26:51 UTC 2015, To: Tue Jul 08 11:26:30 UTC 2025]
            Issuer: O=W12R2UINanduVC, C=US, DC=local, DC=vsphere, CN=CA ...
            ...
        ]"
    }
],
...
"id": "a97bd204-e3e5-404b-a219-e2b20cf158d2"
}

```

API 會建立一個新的介面卡，當中包含由內部產生的 UUID，可唯一識別物件。API 回應包括 vCenter 12.345.678.9 提供的憑證。介面卡執行個體識別碼的值，可用於啟動監控和收集資料。

## 提供憑證有效的證明

在 vRealize Operations Manager 可以連線至 vCenter 並開始收集資料之前，需要確認介面卡執行個體探索到的資料來源提供的是有效的憑證。您的 PATCH 要求包括要求內文 (即用於建立介面卡之 POST 要求的回應)，提供了憑證有效的證明。

## 必要條件

確認您有用於建立介面卡之 POST 要求的回應。請參閱 [介面卡執行個體](#) 中的回應。

## 程序

- ◆ 提出 PATCH 要求，以向系統通知使用者已接受 vCenter 提供的憑證。

```
PATCH https://vrealize.example.com/suite-api/api/adapters
```

## 範例：憑證驗證

在此範例中，PATCH 要求的要求內文與用於建立介面卡執行個體之 POST 要求的回應相同。

要求標頭：

```
PATCH https://vrealize.example.com/suite-api/api/adapters
```

以 JSON 格式撰寫的要求內文：

```
{
  "resourceKey": {
    "name": "VC Adapter Instance",
    "adapterKindKey": "VMWARE",
    "resourceKindKey": "VMwareAdapter Instance",
    "resourceIdentifiers": [
      {
        "identifierType": {
          "name": "AUTODISCOVERY",
          "dataType": "STRING",
          "isPartOfUniqueness": false
        },
        "value": "true"
      },
      {
        "identifierType": {
          "name": "DISABLE_COMPUTATION_BASED_ON_CONSUMERS",
          "dataType": "STRING",
          "isPartOfUniqueness": false
        },
        "value": ""
      },
      {
        "identifierType": {
          "name": "DV_PORT_GROUP_DISABLED",
          "dataType": "STRING",
          "isPartOfUniqueness": false
        },
        "value": ""
      },
      {
        "identifierType": {
          "name": "DVS_DISABLED",
          "dataType": "STRING",
          "isPartOfUniqueness": false
        }
      }
    ]
  }
}
```

```

    },
    "value": ""
  },
  {
    "identifierType": {
      "name": "PROCESSCHANGEEVENTS",
      "dataType": "STRING",
      "isPartOfUniqueness": false
    },
    "value": "true"
  },
  {
    "identifierType": {
      "name": "VCURL",
      "dataType": "STRING",
      "isPartOfUniqueness": true
    },
    "value": "https://12.345.678.9"
  },
  {
    "identifierType": {
      "name": "VM_FOLDER_DISABLED",
      "dataType": "STRING",
      "isPartOfUniqueness": false
    },
    "value": ""
  },
  {
    "identifierType": {
      "name": "VM_LIMIT",
      "dataType": "STRING",
      "isPartOfUniqueness": false
    },
    "value": ""
  }
]
},
"description": "A vCenter Adapter Instance for VC 12.345.678.9",
"collectorId": 1,
"collectorGroupId": "909c2fbf-2c2c-4957-9a75-21bf2a887d31",
"credentialInstanceId": "65081a8d-d462-43b2-b4e0-596eaf3d497e",
"monitoringInterval": 5,
"adapter-certificates": [
  {
    "thumbprint": "2520fb4351bc91ee7b82ef7cc54a8d88fa893da9",
    "certificateDetails": "[
      Version: V3 Subject: C=US, CN=12.345.678.9
      Signature Algorithm: SHA256withRSA, OID = 1.2.840.113549.1.1.11
      Key: Sun RSA public key, 2048 bits modulus: ...
      Validity: [From: Wed Jul 15 19:26:51 UTC 2015, To: Tue Jul 08 11:26:30 UTC 2025]
      Issuer: O=W12R2UINanduVC, C=US, DC=local, DC=vsphere, CN=CA ...
      ...
    ]"
  }
]
}

```

```

],
...
"id": "a97bd204-e3e5-404b-a219-e2b20cf158d2"
}

```

以 JSON 撰寫的回應：

```

{
  "resourceKey": {
    "name": "VC Adapter Instance",
    "adapterKindKey": "VMWARE",
    "resourceKindKey": "VMwareAdapter Instance",
    "resourceIdentifiers": [
      {
        "identifierType": {
          "name": "AUTODISCOVERY",
          "dataType": "STRING",
          "isPartOfUniqueness": false
        },
        "value": "true"
      },
      {
        "identifierType": {
          "name": "DISABLE_COMPUTATION_BASED_ON_CONSUMERS",
          "dataType": "STRING",
          "isPartOfUniqueness": false
        },
        "value": ""
      },
      {
        "identifierType": {
          "name": "DV_PORT_GROUP_DISABLED",
          "dataType": "STRING",
          "isPartOfUniqueness": false
        },
        "value": ""
      },
      {
        "identifierType": {
          "name": "DVS_DISABLED",
          "dataType": "STRING",
          "isPartOfUniqueness": false
        },
        "value": ""
      },
      {
        "identifierType": {
          "name": "PROCESSCHANGEEVENTS",
          "dataType": "STRING",
          "isPartOfUniqueness": false
        },
        "value": "true"
      },
      {
        "identifierType": {

```

```

    "name": "VCURL",
    "dataType": "STRING",
    "isPartOfUniqueness": true
  },
  "value": "https://12.345.678.9"
},
{
  "identifierType": {
    "name": "VM_FOLDER_DISABLED",
    "dataType": "STRING",
    "isPartOfUniqueness": false
  },
  "value": ""
},
{
  "identifierType": {
    "name": "VM_LIMIT",
    "dataType": "STRING",
    "isPartOfUniqueness": false
  },
  "value": ""
}
]
},
"description": "A vCenter Adapter Instance for VC 12.345.678.9",
"collectorId": 1,
"collectorGroupId": "909c2fbf-2c2c-4957-9a75-21bf2a887d31",
"credentialInstanceId": "65081a8d-d462-43b2-b4e0-596eaf3d497e",
"monitoringInterval": 5,
...
"id": "a97bd204-e3e5-404b-a219-e2b20cf158d2"
}

```

此回應與要求內文相同，不含 `adapter-certificates` 區段。

## 啟動監控新的介面卡執行個體

在建立介面卡執行個體並設定 vRealize Operations Manager 以辨識有效憑證後，啟動監控和收集資料。您的 PUT 要求提供用於探索新物件之介面卡執行個體的 UUID。

### 必要條件

確認您擁有新建立介面卡執行個體的 UUID。請參閱 [介面卡執行個體](#) 中的回應。

### 程序

- ◆ 提出 PUT 要求，以使用新的介面卡執行個體啟動監控。

```

PUT https://vrealize.example.com/suite-api/api/adapters/<adapter_UUID>/monitoringstate/
start

```

## 範例：探索物件並收集資料

此範例會使用建立介面卡執行個體的 PUT 要求中的介面卡執行個體識別碼，啟動介面卡監控程序。

要求標頭：

```
PUT https://vrealize.example.com/suite-api/api/adapters/a97bd204-e3e5-404b-a219-e2b20cf158d2/monitoringstate/start
```