

# API 编程指南

2021 年 5 月 12 日

vRealize Operations 8.3

您可以从 VMware 网站下载最新的技术文档:

<https://docs.vmware.com/cn/>。

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

**威睿信息技术（中国）有  
限公司**  
北京办公室  
北京市  
朝阳区新源南路 8 号  
启皓北京东塔 8 层 801  
[www.vmware.com/cn](http://www.vmware.com/cn)

上海办公室  
上海市  
淮海中路 333 号  
瑞安大厦 804-809 室  
[www.vmware.com/cn](http://www.vmware.com/cn)

广州办公室  
广州市  
天河路 385 号  
太古汇一座 3502 室  
[www.vmware.com/cn](http://www.vmware.com/cn)

版权所有 © 2021 VMware, Inc. 保留所有权利。 [版权和商标信息](#)

# 目录

<b>1</b>	<b>关于本编程指南</b>	<b>4</b>
<b>2</b>	<b>了解 vRealize Operations Manager API</b>	<b>5</b>
	vRealize Operations Manager API 的工作原理	6
	客户端工作流程概览	8
	vRealize Operations Manager API REST 请求	9
	vRealize Operations Manager API REST 响应	10
	将 API 与 vRealize Operations Manager 结合使用	11
<b>3</b>	<b>API 入门</b>	<b>13</b>
	获取身份验证令牌	13
	查找适配器类型和对象类型	15
	生成对象的所有衡量指标列表	16
<b>4</b>	<b>配置适配器实例</b>	<b>19</b>
	配置适配器实例请求的摘要	19
	确定解决方案及其适配器	20
	确定适配器所需的对象类型	21
	创建适配器实例	23
	提供证书有效性证明	27
	开始监控新适配器实例	30

# 关于本编程指南

# 1

《vRealize Operations Manager API 编程指南》提供了有关 vRealize Operations Manager REST API 的信息，包括如何使用 REST API 资源、进行身份验证和构建 REST API 调用。

## 目标受众

本指南中的信息主要面向希望使用 vRealize Operations Manager REST API 以编程方式配置和管理 vRealize Operations Manager 的管理员和程序员。本指南重点说明常见用例。

# 了解 vRealize Operations Manager API

## 2

开发人员可以使用 API 构建 vRealize Operations Manager 的交互式客户端。API 遵循 REST 样式，并且可供所有许可用户使用。

vRealize Operations Manager 客户端通过 HTTP 与服务器进行通信，交换 vRealize Operations Manager 对象的表示形式。这些表示形式采用 JSON 或 XML 元素形式。您可以使用 HTTP GET 请求检索对象的当前表示形式，使用 HTTP POST 和 PUT 请求创建或修改对象，以及使用 HTTP 删除请求删除对象。

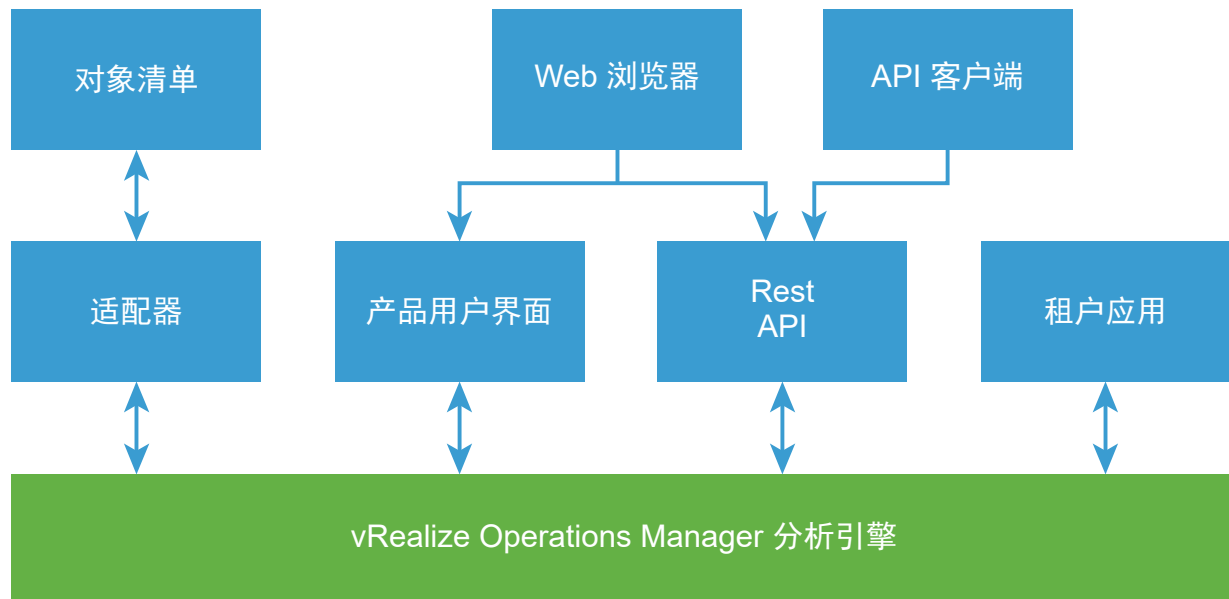
本章讨论了以下主题：

- vRealize Operations Manager API 的工作原理
- 客户端工作流程概览
- 将 API 与 vRealize Operations Manager 结合使用

## vRealize Operations Manager API 的工作原理

您可以使用 Web 浏览器通过产品用户界面或 API 调用与 vRealize Operations Manager 分析引擎进行通信。

图 2-1. vRealize Operations Manager 简化的架构



适配器实例从受监控环境中的对象收集数据。vRealize Operations Manager 分析引擎会处理数据，并在图形界面中显示完整的模型。

## 为什么使用 API

当需要自动执行定义明确的工作流（例如，重复相同的任务以配置 vRealize Operations Manager 新用户的访问控制）时，API 非常有用。在对 vRealize Operations Manager 数据存储库执行查询（例如，检索虚拟环境中特定资产的数据）时，API 也很有用。此外，您还可以使用 API 从 vRealize Operations Manager 数据存储库中提取所有数据，并将其加载到单独的分析系统中。

## vRealize Operations Manager 术语

用于描述适配器对象的 XML 语法与 API 代码语法相对应，但不同于在用户界面中找到的内容。用户界面中将显示以下术语。每个术语的说明中都包含 API 调用中使用的相应 XML 语法。

适配器类型	定义用于发现特定对象类型的适配器。例如，vCenter 适配器会发现连接到 vSphere 数据中心的对象。AWS 适配器可发现 AWS 服务和对象。 XML 语法：adapterkinds。
对象类型	表示对象或信息源的实体类。对象向 vRealize Operations Manager 分析引擎报告数据。虚拟机、数据存储和主机系统都是 vCenter 适配器模型中定义的对象类型的示例。 XML 语法：resourcekinds。

## 客户端工作流程概览

vRealize Operations Manager API 客户端实施 REST 工作流程，向服务器发出 HTTP 请求，并从服务器的响应中检索所需的信息。

## 关于 REST

REST（表征状态转移的首字母缩写）描述了使用超文本传输协议 (HTTP) 在客户端和服务器之间交换对象的序列化表示形式的程序的架构样式特性。在 vRealize Operations Manager API 中，这些表示形式为 JSON 或 XML 文档。

在 REST 工作流程中，对象的表示形式在客户端和服务器之间来回传递，明确假设任何一方都不需要解除单个请求或响应中提供的内容以外的任何对象信息。这些文档可用的 URL 通常会持续到包含它们的请求或响应的有效期之后。

## REST API 工作流程

为使用 REST API 而编写的应用程序使用 HTTP 请求，这些请求通常由脚本或其他更高级别的语言执行，以进行远程程序调用，从而创建、检索、更新或删除 API 定义的对象。在 vRealize Operations Manager REST API 中，这些对象由一组 XML 架构定义。操作本身是 HTTP 请求，因此对于所有 HTTP 客户端来说都是通用的。

要编写 REST API 客户端应用程序，您只需了解 HTTP 协议以及 JSON 或 XML（vRealize Operations Manager API 使用的传输格式）的语义。要在此类客户端中有效使用 API，您必须熟悉以下概念。

- API 支持的对象集及其代表的内容。
- API 如何表示这些对象。
- 客户端如何引用要在其上运行的对象。

API 参考包括 API 请求的完整列表。请参见[关于架构参考](#)。

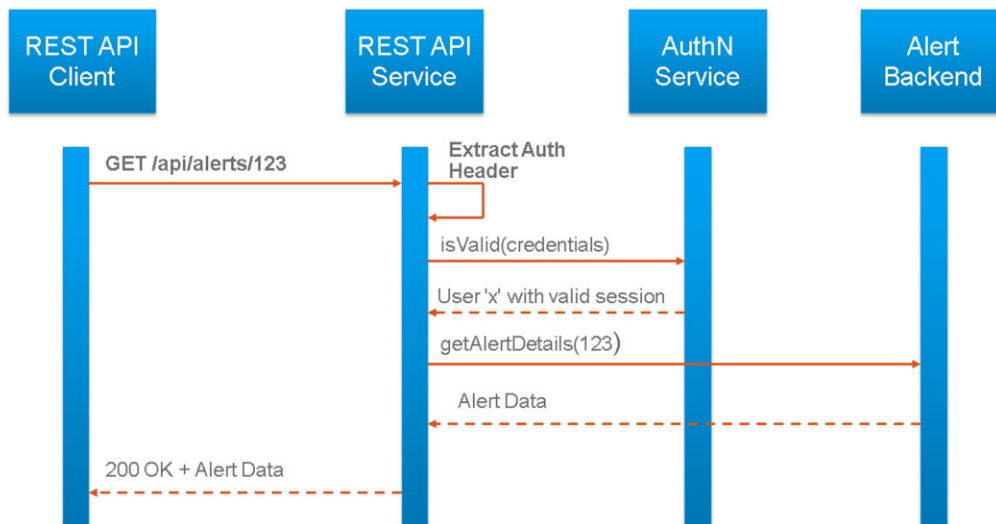
## vRealize Operations Manager API REST 请求

要检索对象表示形式，客户端会向对象引用发出 HTTP 请求。

### 安全

API 客户端和服务端之间的 HTTP 链接是使用 SSL 建立的。API 客户端配置基于令牌的身份验证以与服务端进行通信。

图 2-2. 方案：提供用户凭据以获取有关 ID 为 123 的警示的详细信息



对于基于令牌的身份验证，您可以向 vRealize Operations Manager API 服务器发布登录请求，并提供有效的用户凭据以获取身份验证令牌。以下示例说明了基于令牌的身份验证方案。

- 1 您可以获取 vRealize Operations Manager 实例的有效用户凭据。
- 2 向 REST 端点发布请求以进行身份验证。

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

请求正文包括用户名、密码和身份验证源。

- 3 在响应正文中，端点会返回令牌、到期日期和时间。

- 4 要进行进一步通信，请将令牌对象包含在授权标头中，格式为：

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

或者，如果从 SSO 源获取令牌，则授权标头的格式为：

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

- 5 您可以通过向注销端点发送 POST 请求，使令牌在过期日期和时间之前失效。

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

## 请求标头

API 请求中通常包含以下 HTTP 标头：

**Accept-Language** 要指定响应中所需的语言，请使用 `Accept-Language` 请求标头。`ErrorType` 响应中的消息字符串已本地化。要使用本地化为法语的消息字符串请求响应，请使用以下标头：

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

**授权** 用于创建 API 会话的所有请求都必须包含由您的组织使用的身份提供程序所规定格式的 `Authorization` 标头

**Content-Type** 包含正文的请求必须包括相应的 `HTTP Content-Type` 标头。

- 对于 XML 形式的请求正文，标头必须包含 `Content-Type: application/xml`
- 对于 JSON 形式的请求正文，标头必须包含 `Content-Type: application/json`

**接受** 要指定所需的响应格式，请包含 `Accept` 请求标头。

- 对于 XML 格式的响应，标头必须包含 `Accept: application/xml`
- 对于 JSON 格式的响应，标头必须包含 `Accept: application/json`

## XML 格式的请求正文

对于以 XML 格式编写的请求正文，vRealize Operations Manager 使用验证 XML 分析程序，该分析程序要求请求正文中的元素在顺序和编号上与架构一致。除非满足以下条件，否则请求正文将被拒绝为无效：

- 必须为请求中的元素所表示的所有命名空间提供 XML 命名空间属性。
- 如果请求中表示多个命名空间，则 XML 命名空间属性必须包含标识前缀，并且该前缀必须与该命名空间中的所有元素一起使用。
- 所有必需元素都必须显示在请求正文中。请求正文中出现的所有元素都必须按照架构建立的顺序显示，并且内容必须符合架构指定的类型限制。

## vRealize Operations Manager API REST 响应

所有响应都包含 HTTP 状态代码和接受标头，除非状态代码为 204（无内容）。响应内容取决于请求。某些响应包含文档正文，某些响应仅包含 URL，而某些响应为空。

## HTTP 响应代码

API 客户端可能会在响应中获得一小部分 HTTP 状态代码。

表 2-1. API 返回的 HTTP 状态代码

状态代码	状态说明
200 正常	请求有效且已完成。响应包含文档正文。
201 已创建	请求有效。请求的对象已创建，可以在位置标头中指定的 URL 处找到。
202 已接受	请求有效，并已创建任务来进行处理。此响应通常附带 <code>TaskStatus</code> 元素。
204 无内容	请求有效且已完成。响应不包含正文。
400 错误请求	请求正文格式错误、不完整或无效。
401 未授权	登录失败或身份验证令牌已过期。
403 已禁止	用户未通过身份验证，或者没有足够的特权来访问请求中指定的一个或多个对象。
404 未找到	未找到请求中指定的对象。
405 不允许使用此方法	此对象不支持请求中指定的 HTTP 方法。
406 不可接受	请求识别的资源无法生成请求的 <code>Accept</code> 标头中指定类型的响应。
415 不支持的媒体类型	请求识别的资源不支持指定的 <code>Content-Type</code> 请求和 HTTP 方法。
422 未找到	通常表示格式错误的请求 URL 或请求正文。
429 请求太多	一个客户端发送了太多请求，或者多个客户端同时发送了太多请求，并且服务器由于速率限制而无法处理这些请求。 要解决此问题，请稍后尝试再次发送请求。
500 内部服务器错误	请求已收到，但由于服务器上的内部错误，无法完成。
503 服务不可用	由于资源耗尽或服务器维护等临时状况，服务器当前无法处理请求。
504 网关超时	充当网关或代理的服务器未从请求 URL 指定的上游服务器收到及时响应。

## 将 API 与 vRealize Operations Manager 结合使用

您可以使用浏览器或 HTTP 客户端程序发送请求和接收响应。

### REST 客户端程序

可以发送 HTTPS 请求的所有客户端应用程序都是使用 vRealize Operations Manager API 开发 REST 应用程序的适用工具。REST 客户端插件可用于大多数浏览器和许多 IDE。通常使用以下开源程序：

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

此外，VMware 为 vRealize Operations Manager API 提供特定语言的客户端绑定。请参见[关于架构参考](#)。

## 关于架构参考

vRealize Operations Manager REST API 文档包括 vRealize Operations Manager API 中所有元素、类型、查询和操作的参考资料。还包括架构定义文件。

产品可提供基于 Swagger 的 API 文档，并且能够从登录页直接进行 REST API 调用。

要访问 API 文档，请使用 vRealize Operations Manager 实例的 URL。例如，如果您的实例的 URL 是 `https://vrealize.example.com`，则可从 `https://vrealize.example.com/suite-api/doc/swagger-ui.html` 获得 API 参考案例。

可从以下位置获取特定语言的客户端绑定：

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

## 关于 vRealize Operations Manager API 示例

所有示例都包含 HTTP 请求和响应。这些示例显示与操作相关联的工作流和内容，例如，创建和查询有关受监控环境中的对象的信息。

示例请求正文采用 JSON 格式。vRealize Operations Manager API 所需的请求标头包含在不属于较大示例的片段的示例请求中。

大多数示例响应仅显示与所讨论操作相关的元素和属性。省略号 (...) 表示响应正文中省略的内容。

API 客户端和 vRealize Operations Manager 服务器通过 HTTPS 进行通信，交换 API 对象的 XML 表示形式。

这一简单的 REST API 工作流示例说明了如何获取 VMware vCenter® 适配器的模型定义中包含的某种虚拟机对象类型的所有衡量指标的列表。使用 API，您可以获取任何对象类型的可用衡量指标的完整列表。

本章讨论了以下主题：

- 获取身份验证令牌
- 查找适配器类型和对象类型
- 生成对象的所有衡量指标列表

## 获取身份验证令牌

vRealize Operations Manager 需要对 API 请求进行身份验证。此工作流的第一步是获取身份验证令牌。

要获取身份验证令牌，登录请求将以基本 HTTP 身份验证所需的形式提供用户凭据。在此示例中，用户使用 URL `https://vrealize.example.com/` 登录到 vRealize Operations Manager 实例。

---

**注** 此示例使用基于令牌的身份验证。有关身份验证机制的更多信息，请参阅[安全](#)。

---

使用 `authSource`，可以导入位于其他计算机上的用户和用户组信息并对其进行身份验证。例如，可以从 LDAP、Active Directory、VMware Identity Manager、Single Sign-On 等对用户进行身份验证。导入位于其他计算机的用户帐户信息时，必须定义用于从源计算机导入用户帐户的条件。

创建身份验证源后，可以使用该身份验证源通过指定名称获取令牌。默认的身份验证源类型为“本地”。

### 前提条件

- 确保 Web 浏览器和 vRealize Operations Manager 服务器之间的通道安全。打开浏览器并输入 vRealize Operations Manager 实例的 URL，例如：

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

系统会警告您的连接不是专用连接。单击以确认安全性异常并建立 SSL 握手。

- 确认您可以访问 API。输入 vRealize Operations Manager 实例的 URL，并将 suite-api/docs/rest/index.html 添加到末尾，例如：

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

- 确认您拥有 vRealize Operations Manager 实例用户的登录凭据。

## 步骤

- 1 向登录 URL 发布请求以获取令牌。

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

请参见[登录请求和响应](#)。

- 2 检查响应。

成功的请求会返回 ops 授权令牌，您必须在后续 API 请求中包括该令牌。

## 示例：登录请求和响应

此示例说明了登录用户名为 **vRealize-user**、密码为 **vRealize-dummy-password** 的用户的请求和响应。

请求标头：

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

JSON 格式的请求正文：

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

JSON 格式的响应：

```
200 OK

{
  "token": "8f868cca-27cc-43d6-a838-c5467e73ec45::77cea9b2-1e87-490e-b626-e878beaa23b",
  "validity": 1470421325035,
  "expiresAt": "Friday, August 5, 2016 6:22:05 PM UTC",
  "roles": []
}
```

响应代码指示请求是否成功或者请求为何失败。

- 如果请求成功，服务器将返回 HTTP 响应代码 200（正常），并重新使用将在 6 小时后过期的 ops 授权令牌。在后续每个 API 请求中都必须包含此令牌。

- 如果请求缺少授权标头，则服务器将返回 HTTP 响应代码 403。
- 如果授权标头中提供的凭据无效，则服务器将返回 HTTP 响应代码 401。

## 查找适配器类型和对象类型

您的 vRealize Operations Manager 实例包含多种适配器类型。要查找 vCenter 适配器的适配器类型，请发出 GET 请求以检索所有适配器类型的列表。API 响应包含适配器监控的所有对象类型。

### 前提条件

确认您已登录到 vRealize Operations Manager 实例。

### 步骤

- 1 对所有适配器类型发出 GET 请求。

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

- 2 检查响应以查找 vCenter 适配器和受监控对象类型的列表。

请参阅 [确定 vCenter 适配器的适配器类型和对象类型](#) 的响应部分。

### 示例：确定 vCenter 适配器的适配器类型和对象类型

此示例将查找 vCenter 适配器的适配器类型以及适配器模型定义中包含的所有对象类型。

请求标头：

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

其中，*vROps\_token* 是您从 [获取身份验证令牌](#) 中的响应获取的令牌。

vCenter 适配器的 JSON 响应代码段：

```
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"
  ],
  ...
}

```

对于 vCenter 适配器，adapter-kind 键为 VMWARE。resourceKinds 是 vCenter 适配器所监控的对象类型。对于虚拟机对象类型，resourceKinds 为 VirtualMachine。

## 生成对象的所有衡量指标列表

要为 vCenter 适配器模型中定义的任何虚拟机生成完整的衡量指标列表，请向具有适配器类型和对象类型的 URL 发出 GET 请求。

### 前提条件

验证是否满足以下要求：

- 您已登录到 vRealize Operations Manager 实例。
- 您知道 vCenter 适配器的 adapterKind 值和虚拟机的 resourceKinds 值。请参见[确定 vCenter 适配器的适配器类型和对象类型](#)

### 步骤

- 1 发出 GET 请求以获取衡量指标的元数据。

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

- 2 将响应中列出的衡量指标与用户界面中显示的衡量指标进行比较。请参见[API 和用户界面中的虚拟机衡量指标](#)

## 示例： API 和用户界面中的虚拟机衡量指标

此示例说明了 XML 响应中列出的虚拟机衡量指标如何与 vRealize Operations Manager 用户界面中显示的衡量指标进行比较。

请求：

```
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* 是您从[获取身份验证令牌](#)

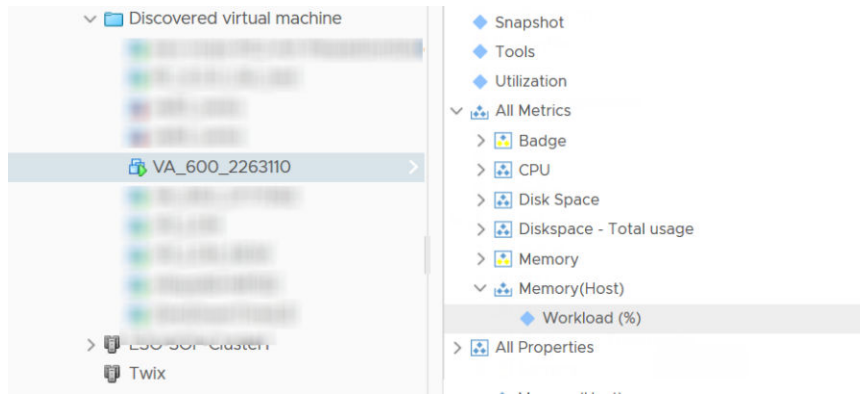
JSON 格式响应的代码段：

```
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
    },
    ...
  ]
}
```

响应中的每个 `resourceTypeAttribute` 都是一个包含虚拟机对象元数据的衡量指标。`name` 对应于 **vRealize Operations Manager** 用户界面中显示的文本。在此示例中，代码段列出了内存和主机工作负载的衡量指标。

要将响应中的衡量指标与用户界面中的衡量指标进行比较，请登录到 `vrealize.example.com` 上运行的 **vRealize Operations Manager** 实例，然后导航到虚拟机的衡量指标。以下示例说明了在何处可以找到内存（主机）和工作负载的衡量指标。



该示例说明了如何检索虚拟机对象类型的衡量指标。要检索其他对象类型的衡量指标，请将 **GET** 请求中的 `VirtualMachine` 替换为其他 `resourceKinds`。

# 配置适配器实例

# 4

在安装包含管理包和适配器的解决方案后，您必须配置适配器实例以从适配器模型定义中的对象收集数据。您可以使用 vRealize Operations Manager API 配置适配器实例。

此用例示例说明了如何为 vSphere 解决方案配置适配器实例，包括：

- 每个操作的每个特定进程的请求、请求正文和响应的
- 操作摘要

本章讨论了以下主题：

- [配置适配器实例请求的摘要](#)
- [确定解决方案及其适配器](#)
- [确定适配器所需的对象类型](#)
- [创建适配器实例](#)
- [提供证书有效性证明](#)
- [开始监控新适配器实例](#)

## 配置适配器实例请求的摘要

您可以发出连续的 API 请求来配置适配器实例。早期请求的响应会生成后续请求所需的信息。

表 4-1. 请求摘要

操作	请求	请求正文	响应
获取在产品中注册的所有解决方案并确定适配器类型	GET <API-URL>/suite-api/api/solutions	无	adapterkindkeys
获取特定适配器类型的所有对象类型。	GET <API-URL>/suite-api/api/adapterkinds/{key}/resourcekinds	无	resourceIdentifierTypes
创建适配器实例对象	POST <API-URL>/suite-api/api/adapters	resourceIdentifiers 和 credential 的值	adapterid

表 4-1. 请求摘要（续）

操作	请求	请求正文	响应
修补适配器实例以确认所提供的证书	PATCH <API-URL>/suite-api/api/adapters	POST <API-URL>/suite-api/api/adapters 的 API 响应	在没有 adapter-certificates 的情况下，POST <API-URL>/suite-api/api/adapters 的 API 响应
开始适配器监控	PUT <API-URL>/suite-api/api/adapters/{adapterid}/monitoringstate/start	无	200 正常

## 确定解决方案及其适配器

您的 vRealize Operations Manager 实例可能安装了多个解决方案。要查找 vSphere 解决方案及其适配器类型，请发出 GET 请求以检索所有解决方案的列表。响应包含解决方案中包括的所有适配器。

在此示例用例中，vRealize Operations Manager 实例的 *API-URL* 为 `vrealize.example.com`。

### 前提条件

确认您可以登录到 vRealize Operations Manager 实例的 API URL。请参见[获取身份验证令牌](#)。

### 步骤

- 1 发出 GET 请求以列出所有解决方案。

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

- 2 检查响应以查找 vSphere 解决方案及其适配器类型。

请参阅 [vSphere 解决方案的适配器类型](#) 的响应部分。

## 示例：vSphere 解决方案的适配器类型

此示例列出了所有已安装的解决方案和每个解决方案的适配器类型。

请求标头：

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

JSON 响应：

```
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"
    ]
}
]
}

```

响应显示三个已安装的解决方案：

- Management Pack for Log Insight 解决方案
- End Point Operations 解决方案
- vSphere 解决方案

vSphere 解决方案具有两种适配器类型：

- VMWARE
- PythonRemediationVcenterAdapter

对于 vCenter 适配器，适配器类型为 VMWARE。

## 确定适配器所需的对象类型

确定要创建 vCenter 适配器的实例后，您必须确定该适配器所需的对象类型。您可发出 GET 请求来检索 vCenter 适配器的所有对象类型的列表。

### 前提条件

确认您知道 vCenter 适配器的适配器类型。

## 步骤

- 1 发出 GET 请求以列出 vCenter 适配器的所有对象类型。

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

- 2 检查响应以确定所需的对象类型。

请参阅 [vCenter 适配器所需的对象类型](#) 的响应部分。

## 示例： vCenter 适配器所需的对象类型

此示例将查找 vCenter 适配器的所有对象类型。

请求标头：

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

JSON 格式响应的代码段：

```
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
    }
  ],
  ...
}

```

此代码段显示具有属性 `"resourceKindType": "ADAPTER_INSTANCE"` 的资源种类。任何具有资源标识符 `"isPartOfUniqueness": true` 的对象类型都需要该对象类型的值，并且需要 API 请求来创建适配器实例。

vCenter 适配器的适配器实例需要 VCURL 的值或 vCenter 的 URL。

## 创建适配器实例

确定适配器所需的对象类型后，您可以为对象类型提供参数值以创建适配器实例。您的 POST 请求包括请求正文以及所需参数。

要创建适配器实例，必须使用 VCURL 设置。

### 前提条件

确认您具有 vCenter 的 IP 地址和凭据。

### 步骤

- 1 发出 POST 请求以创建适配器实例。

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

- 2 检查响应以查找 vSphere 解决方案的名称及其适配器类型。

请参阅 [适配器实例](#) 的响应部分。

## 示例： 适配器实例

此示例将使用以下参数为 vCenter 创建适配器实例：

- 显示名称：VC 适配器实例
- 说明：VC 12.345.678.9 的 vCenter 适配器实例
- vCenter Server IP 地址：https://12.345.678.9
- 凭据名称：VC-Credential-1
- 用户名：administrator@vsphere.local

## ■ 密码: VC-dummy-passwd

AUTODISCOVERY 和 PROCESSCHANGEEVENTS 为可选，但包含在请求正文和响应中可显示更多资源标识符示例。

请求标头:

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

JSON 格式的请求正文:

```
{
  "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"
      }
    ]
  },
},
}
```

JSON 格式响应的代码段:

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 将创建一个新的适配器，其中包含内部生成的、可唯一标识对象的 UUID。API 响应包含 vCenter 12.345.678.9 提供的证书。适配器实例 ID 的值用于开始监控和收集数据。

## 提供证书有效性证明

在 vRealize Operations Manager 可以连接到 vCenter 并开始收集数据之前，需要验证适配器实例发现的数据源是否正在呈现有效的证书。您的 PATCH 请求通过包括请求正文（即用于创建适配器的 POST 请求的响应）来提供证书有效的证明。

### 前提条件

验证您是否具有用于创建适配器的 POST 请求的响应。请参阅 [适配器实例](#) 中的响应。

### 步骤

- ◆ 发出 PATCH 请求以通知系统用户已接受 vCenter 提供的证书。

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

### 示例：证书验证

在此示例中，PATCH 请求的请求正文与用于创建适配器实例的 POST 请求的响应相同。

请求标头：

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

JSON 格式的请求正文：

```
{
  "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"
}

```

JSON 格式的响应:

```

{
  "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"
}

```

该响应与请求正文相同，但不含 adapter-certificates 部分。

## 开始监控新适配器实例

在创建适配器实例并配置 vRealize Operations Manager 以识别有效的证书后，开始监控和收集数据。您的 PUT 请求提供用于发现新对象的适配器实例的 UUID。

### 前提条件

确认您具有新创建的适配器实例的 UUID。请参阅 [适配器实例](#) 中的响应。

## 步骤

- ◆ 发出 PUT 请求以使用新的适配器实例开始监控。

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

## 示例：发现对象并收集数据

此示例使用创建了适配器实例的 PUT 请求中的适配器实例 ID 启动适配器监控进程。

请求标头：

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