

# Entwickeln eines Webdienstclients für VMware vRealize Orchestrator

vRealize Orchestrator 7.4



vmware®

Die neueste technische Dokumentation finden Sie auf der VMware-Website unter:

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

Falls Sie Anmerkungen zu dieser Dokumentation haben, senden Sie Ihre Kommentare und Vorschläge an:

[docfeedback@vmware.com](mailto:docfeedback@vmware.com)

**VMware, Inc.**

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

**VMware Global, Inc.**

Zweigniederlassung Deutschland  
Freisinger Str. 3  
85716 Unterschleißheim/Lohhof  
Germany  
Tel.: +49 (0) 89 3706 17000  
Fax: +49 (0) 89 3706 17333  
[www.vmware.com/de](http://www.vmware.com/de)

Copyright © 2008–2018 VMware, Inc. Alle Rechte vorbehalten. [Informationen zu Copyright und Marken.](#)

# Inhalt

Entwickeln eines Webdienstclients für VMware vRealize Orchestrator 5

## 1 Entwickeln eines Webdienstclients 6

## 2 Verwenden der vRealize Orchestrator -REST-API 7

Authentifizierung bei Orchestrator- und Drittanbietersystemen 8

Verwenden der vCenter Single Sign-On-Authentifizierung mit der Orchestrator-REST-API 8

Zugreifen auf die Referenzdokumentation für die Orchestrator-REST-API 13

Verwenden des Java REST SDK 13

Vorgänge mit Workflows 14

Suchen eines Workflows und Abrufen seiner Definition 15

Ausführen eines Workflows 18

Ausführen eines Workflows nach der Validierung seiner Eingabeparameter anhand der Workflowpräsentation 20

Interaktion mit laufenden Workflows 24

Abrufen der Interaktionen eines Workflows 31

Zugriff auf das Schema eines Workflows 32

Arbeiten mit Aufgaben 32

Erstellen von Aufgaben 32

Ändern einer Aufgabe 34

Überprüfen des Status einer Aufgabe 35

Suchen von Objekten in der Orchestrator-Bestandsliste 35

Durchsuchen von Objekten nach Typ und ID 36

Suchen von Objekten nach Beziehungen 37

Anwenden von Filtern 38

Importieren und Exportieren eines Orchestrator-Objekts 39

Importieren eines Workflows 39

Exportieren eines Workflows 39

Importieren einer Aktion 40

Exportieren einer Aktion 40

Importieren eines Pakets 40

Exportieren von Paketen 42

Importieren einer Ressource 42

Exportieren einer Ressource 43

Importieren von Konfigurationselementen 43

Exportieren von Konfigurationselementen 44

Löschen von Orchestrator-Objekten 44

Löschen eines Workflows 44

Löschen einer Aktion	45
Löschen eines Pakets	45
Löschen einer Ressource	46
Löschen eines Konfigurationselements	46
Festlegen von Berechtigungen für Orchestrator-Objekte	46
REST-API-Berechtigungen	47
Abrufen der Berechtigungen eines Workflows	47
Löschen der Berechtigungen eines Workflows	48
Festlegen der Berechtigungen für einen Workflow	48
Abrufen der Berechtigungen einer Aktion	49
Löschen der Berechtigungen einer Aktion	49
Festlegen der Berechtigungen für eine Aktion	49
Abrufen der Berechtigungen eines Pakets	50
Löschen der Berechtigungen eines Pakets	50
Festlegen der Berechtigungen für ein Paket	51
Abrufen der Berechtigungen einer Ressource	51
Löschen der Berechtigungen einer Ressource	52
Festlegen der Berechtigungen für eine Ressource	52
Abrufen der Berechtigungen eines Konfigurationselements	53
Löschen der Berechtigungen eines Konfigurationselements	53
Festlegen der Berechtigungen für ein Konfigurationselement	54
Durchführen von Vorgängen mit Plug-Ins	54
Abrufen von Informationen zu Plug-Ins	54
Importieren von Plug-Ins	55
Exportieren eines Plug-Ins	55
Aktivieren oder Deaktivieren von Plug-Ins	56
Ausführen von Server-Konfigurationsvorgängen	56
Abrufen von Informationen zur Orchestrator-Serverkonfiguration	56
Importieren der Serverkonfiguration des Orchestrator-Servers	57
Exportieren der Serverkonfiguration eines Orchestrator-Servers	57
Ausführen von Tagvorgängen	58
Markieren eines Objekts	58
Entfernen von Tags von einem Objekt	59
Auflisten von Objekttags	59
Auflisten markierter Objekte nach Typ	59
Auflisten von Tagbesitzern	60
Auflisten von Tags nach Benutzer	60
Auflisten von Tags nach Benutzer, gefiltert nach Tagname	61
Entfernen von Tags nach Benutzer	61

# Entwickeln eines Webdienstclients für VMware vRealize Orchestrator

*Entwickeln eines Webdienstclients für VMware vRealize Orchestrator* stellt Informationen zur Entwicklung eines Webdienstclients für VMware<sup>®</sup> vRealize Orchestrator bereit.

Orchestrator stellt eine Webdienst-API bereit, sodass Sie Anwendungen entwickeln können, um auf Workflows zugreifen und sie über das Web nutzen können. Orchestrator stellt eine Representational State Transfer (REST)-API bereit, die Sie zur Durchführung verschiedener Vorgänge über Workflows verwenden können.

## Zielgruppe

Diese Informationen sind für Entwickler von Webanwendungen gedacht, die über den RESTful-Webdienst auf Orchestrator-Vorgänge über ein Netzwerk hinweg zugreifen wollen.

# Entwickeln eines Webdienstclients

1

VMware vRealize Orchestrator stellt Webdienst-APIs bereit, sodass Sie Anwendungen entwickeln können, um auf Workflows über das Web zuzugreifen. Der Hauptzweck der Orchestrator-Webdienste-APIs ist, Ihnen das Integrieren von Orchestrator-Workflows in benutzerdefinierten webbasierten Anwendungen zu ermöglichen.

Orchestrator bietet eine Webdienste-API, die auf einer Representational State Transfer-(REST)-API basiert. Die Orchestrator-REST-API stellt Objekte in der Orchestrator-Bestandsliste und den Bestandslisten der installierten Plug-Ins als Ressourcen bereit, auf die Sie über vordefinierte URLs zugreifen können. HTTP-Anforderungen für die URLs lösen Workflow-bezogene Vorgänge aus. Die Orchestrator-REST-API stellen Bestandslistenobjekte als Ressourcen über einen Satz von RESTful-Webdiensten zur Verfügung, mit denen Sie die Definitionen von Workflows abrufen, Workflows ausführen, den Status der laufenden Workflows überprüfen, Workflowausführungen abbrechen, wartende Benutzerinteraktionen verarbeiten, die Darstellung von Workflows abrufen können, usw.

# Verwenden der vRealize Orchestrator -REST-API

# 2

Mithilfe der Funktionen der Orchestrator-REST-API können Sie direkt über HTTP mit dem Orchestrator-Server kommunizieren und verschiedene Workflow-bezogene Vorgänge ausführen.

Die Orchestrator-REST-API stellt Objekte in den Bestandslisten des Orchestrator-Servers sowie die installierten Plug-Ins als Ressourcen mit vorab festgelegten URLs dar. Durch HTTP-Aufrufe an diese URLs lösen Sie Vorgänge in Orchestrator aus. Auf diese Weise können Sie verschiedene Aufgaben für Workflows durchführen:

- Workflow ausführen, Workflow planen, Ausführungen eines Workflows abrufen, Benutzerinteraktion beantworten, Workflowausführung beenden
- Details zum Workflow abrufen, etwa seine Ein- und Ausgabeparameter und Darstellung
- Details zu einer Workflowausführung abrufen, etwa den Status, erstellte Protokolle, Start- und Enddatum
- Bestandslisten und installierte Plug-Ins von Orchestrator durchsuchen
- Workflows, Aktionen und Pakete importieren und exportieren

Mithilfe der Orchestrator-REST-API können Sie Orchestrator-Workflows auf einfache Weise in benutzerdefinierte Anwendungen in jeder beliebigen Programmiersprache integrieren.

Die Orchestrator-REST-API bietet eTag-Unterstützung sowie einen Mechanismus für das Zwischenspeichern von Antwortdaten.

Dieses Kapitel behandelt die folgenden Themen:

- [Authentifizierung bei Orchestrator- und Drittanbietersystemen](#)
- [Zugreifen auf die Referenzdokumentation für die Orchestrator-REST-API](#)
- [Verwenden des Java REST SDK](#)
- [Vorgänge mit Workflows](#)
- [Arbeiten mit Aufgaben](#)
- [Suchen von Objekten in der Orchestrator-Bestandsliste](#)
- [Importieren und Exportieren eines Orchestrator-Objekts](#)
- [Löschen von Orchestrator-Objekten](#)

- [Festlegen von Berechtigungen für Orchestrator-Objekte](#)
- [Durchführen von Vorgängen mit Plug-Ins](#)
- [Ausführen von Server-Konfigurationsvorgängen](#)
- [Ausführen von Tagvorgängen](#)

## Authentifizierung bei Orchestrator- und Drittanbietersystemen

Sie müssen sich bei Orchestrator in den HTTP-Anforderungen authentifizieren, die Sie über die Orchestrator-REST-API ausführen. Wenn Sie die Orchestrator-REST-API für den Zugriff auf Ressourcen auf einem Drittanbietersystem, beispielsweise vCenter Server oder vRealize Automation, verwenden, müssen Sie sich auch bei diesem System authentifizieren.

Um beispielsweise auf alle Workflows in der Orchestrator-Bestandsliste zugreifen zu können, müssen Sie sich bei Orchestrator authentifizieren. Um aber einen Workflow für vCenter Server auszuführen, müssen Sie sich bei Orchestrator und vCenter Server authentifizieren.

Je nachdem, ob Sie Orchestrator mit vRealize Automation oder vSphere als Authentifizierungsanbieter konfigurieren, unterscheidet sich das Authentifizierungsschema für die Orchestrator-REST-API. Wenn Orchestrator vCenter Single Sign-On verwendet, können Sie je nach Ihrer Konfiguration die Authentifizierung unter Verwendung eines Holder-of-key-Tokens durchführen, der vom vCenter Single Sign-On-Server ausgestellt wurde. Wenn Orchestrator mit vRealize Automation konfiguriert ist, können Sie sich über einen OAuth-Bearer-Zugriffstoken authentifizieren.

Wenn Sie HTTP-Anforderungen unter der URL der Orchestrator-REST-API der obersten Ebene ausführen, müssen Sie sich nicht bei Orchestrator authentifizieren. Die URL der Orchestrator-REST-API der obersten Ebene lautet `https://Orchestrator_Host:Port/vco/api/`.

---

**Hinweis** Die Standardportnummer für den externen Orchestrator lautet 8281. Die Standardportnummer für die in vRealize Automation eingebettete Orchestrator-Instanz lautet 443.

---

Eine GET-Anforderung unter der URL der obersten Ebene der REST-API gibt URLs an alle Ressourcen zurück, die über die API zugänglich sind. Um HTTP-Anforderungen unter diesen URLs auszuführen, müssen Sie sich bei Orchestrator authentifizieren.

## Verwenden der vCenter Single Sign-On-Authentifizierung mit der Orchestrator-REST-API

Wenn Orchestrator mit dem vCenter Single Sign-On-Server mithilfe des vSphere-Authentifizierungsmodus konfiguriert ist, benötigen Sie einen Holder-of-key-Prinzipaltoken für den Zugriff auf Systemobjekte in Orchestrator mithilfe der Orchestrator-REST-API. Für den Zugriff auf vCenter Server oder Drittanbietersysteme, die vCenter Single Sign-On über den Orchestrator-Server nutzen, benötigen Sie einen Holder-of-key-Delegattoken für Orchestrator sowie Ihren Prinzipaltoken.

Wenn Orchestrator mit dem vCenter Single Sign-On-Server konfiguriert ist, benötigen Sie für die Authentifizierung gültige Anmeldedaten, und Orchestrator verwaltet die Holder-of-key-Token.



## Zugriff auf Systemobjekte in Orchestrator

Sie können auf Systemobjekte in Orchestrator über die URLs der Bestandsliste oder über die Katalogdienste der REST-API zugreifen.

- `https://Orchestrator_Host:Port/vco/api/inventory/System/`
- `https://Orchestrator_Host:Port/vco/api/catalog/System/`

Beim Zugriff auf Systemobjekte in Orchestrator übergeben Sie Ihren Holder-of-key-Prinzipaltoken im Authorization-Header der HTTP-Anforderung, die an die Bestandsliste oder den Katalogdienst gestellt wird.

Beispiel: Um alle Systemobjekte vom Typ Workflow abzurufen, stellen Sie eine GET-Anforderung an `https://Orchestrator_Host:Port/vco/api/catalog/System/Workflow/`. Zur Authentifizierung bei Orchestrator müssen Sie im Authorization-Header der Anforderung Ihren Holder-of-key-Prinzipaltoken übergeben.

## Zugriff auf Objekte in Drittanbietersysteme

Zur Durchführung von Vorgängen in Drittanbietersystemen, die über die Orchestrator-REST-API beim vCenter Single Sign-On-Server registriert sind, müssen Sie sich bei Orchestrator und beim Drittanbietersystem authentifizieren. Verwenden Sie bei den HTTP-Anforderungen, die Sie über die Orchestrator-REST-API ausführen, zwei Header.

- **Authorization.** In diesem Header übergeben Sie den Holder-of-key-Prinzipaltoken.
- **VCOAuthorization.** In diesem Header übergeben Sie den Holder-of-key-Delegattoken für Orchestrator. Den Delegattoken für Orchestrator beziehen Sie vom vCenter Single Sign-On-Server. Orchestrator verwendet den Delegattoken, wenn er für Sie die Authentifizierung beim Drittanbietersystem durchführt.

Beispiel: Um einen Workflow über die Orchestrator-REST-API auszuführen, in dem eine virtuelle Maschine verwendet wird, greifen Sie auf Ressourcen in Orchestrator und in vCenter Server zu. Zur Authentifizierung bei Orchestrator und vCenter Server müssen Sie im Authorization-Header der gestellten Anforderung den Holder-of-key-Prinzipaltoken und im VCOAuthorization-Header den Delegattoken übergeben. Dadurch authentifizieren Sie sich mit dem Prinzipaltoken bei Orchestrator, und Orchestrator authentifiziert Sie mithilfe des Delegattokens bei vCenter Server.

Der vCenter Single Sign-On-Server behandelt Orchestrator als eine Lösung, und jede Lösung ist mithilfe eines eindeutigen Benutzernamens beim vCenter Single Sign-On-Server registriert. Sie fordern einen Delegattoken für Orchestrator an, indem Sie den Orchestrator-Benutzernamen der Lösung und den Holder-of-key-Prinzipaltoken an den vCenter Single Sign-On-Server übergeben. Der vom vCenter Single Sign-On-Server ausgegebene Token ist ein Holder-of-key-Delegattoken für Orchestrator für Ihre Authentifizierung bei Drittanbietersystemen.

## Beispiel: Abrufen einer Sitzung im vCenter Single Sign-On-Modus

Mit dem folgenden Beispielcode wird im vCenter Single Sign-On-Modus eine Sitzung abgerufen.

```
URI uri = URI.create("https://orchestrator-server:8281/vco/api");
VcoSessionFactory sessionFactory = new DefaultVcoSessionFactory(uri);

//provide the address of the vCenter Single Sign-On server
URI ssoUri = URI.create("https://sso-server:7444/ims/STSService?wsdl");

//set the tokens to be valid for an hour
long lifeTimeSeconds = 60 * 60;

//create a factory for vCenter Single Sign-On tokens
SsoAuthenticator sso = new SsoAuthenticator(URI ssoUri, URI adminUri, VcoSessionFactory vcoSessionFactory, long lifeTimeSeconds);

//provide vCenter Single Sign-On credentials
SsoAuthentication authentication = sso.createSsoAuthentication("username", "password");

VcoSession session = sessionFactory.newSession(authentication);
//use session here
```

## Abrufen des Benutzernamens der Orchestrator-Lösung

Der vCenter Single Sign-On-Server behandelt Orchestrator als eine Lösung, und jede Lösung ist mithilfe eines eindeutigen Benutzernamens beim vCenter Single Sign-On-Server registriert. Zum Anfordern eines Holder-of-key-Delegattokens für Orchestrator vom vCenter Single Sign-On-Server wird der Benutzername der Orchestrator-Lösung benötigt.

### Voraussetzungen

Vergewissern Sie sich, dass Sie über einen gültigen Holder-of-key-Prinzipaltoken vom vCenter Single Sign-On-Server verfügen.

### Vorgehensweise

- 1 Führen Sie eine GET-Anforderung unter der URL des Benutzernamens der Orchestrator-Lösung aus:

```
GET https://{Orchestrator_Host}:{Port}/vco/api/users/
```

- 2 Geben Sie den Holder-of-key-Prinzipaltoken im Authorization-Header der Anforderung an.

Das Element `<user solution-user="OrchestratorSolutionUserName"/>` der Antwort enthält den Benutzernamen der Orchestrator-Lösung. Im Folgenden finden Sie ein Beispiel für einen Benutzernamen einer Orchestrator-Lösung.

```
<user xmlns="http://www.vmware.com/vco" admin-rights="true" solution-user="vCO-15d98795afa5b0d6f47ee3aeab3">
```

## Weiter

Mit diesem Benutzernamen und dem Holder-of-key-Prinzipaltoken können Sie einen Holder-of-key-Dele-gattoken für Orchestrator vom vCenter Single Sign-On-Server anfordern.

## Verwenden des REST-API-SDK von vRealize Orchestrator mit einer konfigurierten vRealize Automation-Authentifizierung

Sie können das REST-API-SDK mit einer konfigurierten vRA-Authentifizierung in einer Umgebung mit mehreren Mandanten oder mit einem Mandanten verwenden.

Sie finden Informationen zum Erhalt eines Authentifizierungstokens (OAuth 2.0), das für den unten darge-stellten Code erforderlich ist, im KB-Artikel [vRO REST API authorization using OAuth2.0 Authentication \(2148518\)](#).

---

**Hinweis** Abrufen einer Sitzung im vRealize Automation-Authentifizierungsmodus

---

Der folgende Beispielcode ruft eine Sitzung im vRealize Automation-Authentifizierungsmodus sowohl in einer Umgebung mit einem Mandanten als auch in einer Umgebung mit mehreren Mandanten ab.

- Wenn die Mandantenfähigkeit nicht aktiviert ist:

```
URI uri = URI.create("https://orchestrator-server:8283/vco/api");
VcoSessionFactory sessionFactory = new DefaultVcoSessionFactory(uri);
String token = "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJqdGkiOiJNzg4NWNiYS1hZTFmLTRiM2UtYmYyYi04ZmRmNzY3N"
+ "GZiZWElLCJwcm4iOiJhZG1pbmlzdHJhdG9yQFZTUeHFUKUuTE9DQUwiLCJkb21hZW4iOiJ2c3BoZXJlLmxvY2FsIi-
widXNlc1pZCI6Ij"
+ "MiLCJhdXRoX3RpbWUiOiJlMDIyMDIxMTAsImZyI6Imh0dHBzOi8vc2YtMjktMTAtMj-
kuc29mLW1idS5lbmcudm13YXJlLnN"
+ "vbS9TQUFTL3QvdnNwaGVyZS5sb2Nhbc9hdXRoIiwiaXVkiOiJoiaHR0cHM6Ly9zZi0yOS0xMC0yOS5zb2YtbWJlLmVu-
Zy52bXdhcmUuY29tL"
+ "1NBQVMvdC92c3BoZXJlLmxvY2FsL2F1dGgvd2F1dGh0b2t1biIsImN0eCI6Ilt7XCJtdGRcIjpcInVybjpvYXNpczpu-
YW1lczp"
+ "0YzpTQU1M0jIuMDphYzpjbjGFzc2VzO1Bhc3N3b3JkUHJvdGVjdGVkVHJhbnNwb3J0XCIsXCJpYXRcIjoxNTAyMjAyMTAwLW-
wiaWRcIjoxM"
+ "n1dIiwic2NwIjoidXNlciiIsImkccCI6IjAiLCJlbWwiOiJhZG1pbmlzdHJhdG9yQHNmLTl5LTUwLTI5LnNvZi1tYnU-
uZW5nLnZ"
+ "td2FyZS5jb20iLCJjaWQiOiJjYWZlX2NsaS1yRlJlNEx6bURnIiwiaXVkiOiJlMDIyMDIxMTAsImZyI6Imh0dHBzOi8vc2YtMjktMTAtMj-
TEwL2F1dGgvd2F1dGh0b2t1biIsImN0eCI6Ilt7XCJtdGRcIjpcInVybjpvYXNpczpu-
YW1lczp"
+ "E1MDIyMDIxMTAsImZyI6Imh0dHBzOi8vc2YtMjktMTAtMj-
TEwL2F1dGgvd2F1dGh0b2t1biIsImN0eCI6Ilt7XCJtdGRcIjpcInVybjpvYXNpczpu-
YW1lczp"
+ "G9gEQPtmEH5jYab-ILTK8NFYcwc3JZCEEjsmpUSH6oxLmZKEf-1JbsysBVH4ufqmGah3GMvmy6PUiTTamLRLfKCL-
wa500QCQ-qgKutZ121R"
+ "m740qBKLhmBB0NQg19ysMAVJNSxapFzirmWurf_5CKpv4WM7Y8H_bY9iNmDKQTXI";

//provide OAuth2 token obtained in step 1 here
Authentication auth = new OAuthTokenAuthentication(token);

VcoSession session = sessionFactory.newSession(auth);
//Use the session here
```

- Wenn die Mandantenfähigkeit aktiviert ist:
  - Für Benutzer von regulären Mandanten:

```
URI uri = URI.create("https://orchestrator-server:8283/vco/api");
VcoSessionFactory sessionFactory = new DefaultVcoSessionFactory(uri);
String token = "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJqdGkiOiJjNzg4NWNiYS1hZTFmLTRiM2UyYmYyY04ZmRmZnY3N"
+ "GZiZWEiLCJwcm4iOiJhZG1pbmlzdHJhdG9yQFZTUeHFUKUuTE9DQUwiLCJkb21haW4iOiJ2c3BoZXJlLmxvY2FsIi-
widXNlcl9pZCI6Ij"
+ "MiLCJhdXRoX3RpbWUiOiJlMDIyMDIxMTAsImZcyI6Imh0dHBzOi8vc2YtMjktMTAtMj-
kuc29mLWlidS5lbmcudm13YXJlLnN"
+ "vbS9TUFTL3QvdnNwaGVyZS5sb2Nhbc9hdXRoIiwiaXVkbWVkaHR0cHM6Ly9zZi0yOS0xMC0yOS5zb2YtbWJlLmVu-
Zy52bXdhcmUuY29tL"
+ "1NBQVMvdC92c3BoZXJlLmxvY2FsL2F1dGgvd2F1dGh0b2t1biIsImN0eCI6Ilt7XCJtdGRcIjpcInVybjpvYXNpczpu-
YW1lc3p"
+ "0YzptQU1MOjIuMDphYzpjbgFzc2VzOlBhc3N3b3JkUHJvdGJvdGVkVHJhbnNwb3J0XCIsXCJpYXRcIjoxNTAyMjAyM-
TEwLWlwiRcIjoxM"
+ "n1dIiwic2NwIjoidXNlciIsImk0cCI6IjAiLCJlbWwiOiJhZG1pbmlzdHJhdG9yQHNmLTI5LTEwLTI5LnNvZi1tYnU-
uZW5nLnZ"
+ "td2FyZS5jb20iLCJjaWQiOiJjYWZlX2NsaS1yRlJlNEh0bURnIiwiaXVkbWVkaHR0cHM6Ly9zZi0yOS0xMC0yOS5zb2YtbWJlLmVu-
Zy52bXdhcmUuY29tL"
+ "E1MDIyMDIxMTAsInN1YiI6IjQ1ZjQwNWUzLTNlNTgtNGJmZC1hNzMwLTQ1MjU4OWIxOGUxNyIsInByb190eXB1Ijo-
iVVNFUiJ9."
+ "G9gEQPtMEH5jYab-ILTK8NFYcwc3JZCEEjsmpUSH6oxLmZKEf-1JbsysBVH4ufqmGah3GMvmy6PUiTTamLRLfKCL-
wa500QCQ-qgKutZl21R"
+ "m740qBKLhmBB0NQg19ysMAVJNSxapFzirmWurF_5CKpv4WM7Y8H_bY9iNmDKQTXI";

//provide OAuth2 token obtained in step 1 here
Authentication auth = new OAuthTokenAuthentication(token);

VcoSession session = sessionFactory.newSession(auth);
//The operations will be executed in the scope of the tenant authenticated with the token abo-
ve.

//Use the session below
```

- Für Lösungsbenutzer:

Lösungsbenutzer können in ihrem eigenen Mandantengeltungsbereich und im Geltungsbereich der regulären Mandanten arbeiten. Sie können den Geltungsbereich der von ihnen durchgeführten Vorgänge außer Kraft setzen.

```
URI uri = URI.create("https://orchestrator-server:8283/vco/api");  
VcoSessionFactory sessionFactory = new DefaultVcoSessionFactory(uri);  
  
exapmle  
  
String token = "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJqdGkiOiJjNzg4NWwiYXN1hZTFmLTRiM2UtYyYyYiO4ZmRmNzY3N"  
+ "GZIWEiLJCwcm4iOiJhZG1pbmlzdHJhdG9yQFZTUeHFUKUuTE9DQUwiLCJkb2lhaW4iOiJ2c3BoZXJLLmxvY2FsIi-  
widXNlc1p9ZCI6Ij"  
+ "MiLCJhdXRoX3RpbWUiOjE1MDIyMDIxMTAsImZscyI6Imh0dHBzOi8vc2YtMjk tMTAtMj_-  
kuc29mLWlidS5lbmcudm13YXJLLmN"  
+ "ybS9TOUFTL3OvdnNwaGVvZS5sb2Nhbc9hdXR0IiwiaXYXVkJioiaHR0cHM6Ly9zaZi0vOS0xMC0vOS5zb2YtbWJlLmVu-
```

```

Zy52bXdhcmUuY29tL"
+ "1NBQVMvdC92c3BoZXJlLmxvY2FsL2F1dGgvd2F1dGh0b2t1biIsImN0eCI6Ilt7XCJtdGRcIjpcInVybjpvYXNpczpu-
YW1lc3p"
+ "0YzpTQU1MOjIuMDphYzpjbgFzc2VzO1Bhc3N3b3JkUHJvdGVjdGVkVHJhbnNwb3J0XCIsXCJpYXRcIjoxNTAyMjAyM-
TEwLFwiaWRcIjoxM"
+ "n1dIiwic2NwIjoidXNlciIsImlkciCI6IjAiLCJlbWwiOiJhZG1pbmlzdHJhdG9yQHNMLTi5LTewLTi5LnNvZi1tYnU-
uZW5nLnZ"
+ "td2FyZS5jb20iLCJjaWQiOiJjYWZlX2NsaS1yRlJlNEx6bURnIiwiZGlkIjoiIiwid2lkIjoiIiwiZXhwIjoxN-
TAyMjMwOTEwLCJpYXQ0j"
+ "E1MDIyMDIxMTAsInN1YiI6IjQ1ZjQwNWUzLTNlNTgtNGJmZC1hNzMwLTQ1MjU0WIX0GUxNyIsInBybl90eXBlijo-
iVVNFUiJ9."
+ "G9gEQPtEH5jYab-ILTK8NFYcwc3JZCEEjsmpUSH6oxLmZKEf-1JbsysBVH4ufqmGah3GMvmy6PUiTTamLRLfKCL-
wa500QCQ-qgKutZl21R"
+ "m740qBKLhmbB0NQg19ysMAVJNSxapFzirmWurf_5CKpv4WM7Y8H_bY9iNmDKQTXI";

//provide OAuth2 token obtained in step 1 here
Authentication auth = new OAuthTokenAuthentication(token);

// By default each tenant works in its tenant scope. However, solution users can overrde the
tenant in which they perform a given operation:
// Here, users of SDK should provide a value that is meaningful to their context.
String overrideWithTenant = "nonSolutionUserTenant";

VcoSession session = sessionFactory.newSession(auth, overrideWithTenant);

//Use session below

```

## Zugreifen auf die Referenzdokumentation für die Orchestrator-REST-API

Die Referenzdokumentation oder die Orchestrator-REST-API enthält Informationen zu den RESTful-Webdiensten der API, zu dem auf die API anwendbaren Datenmodell, zu den für die API gültigen Antwortcodes, zu Codebeispielen usw.

Die Referenzdokumentation der Orchestrator-REST-API wird zusammen mit Orchestrator installiert. Die Referenzdokumentation ist unter [https://Orchestrator\\_Host:Port/vco/api/docs/](https://Orchestrator_Host:Port/vco/api/docs/) verfügbar.

Die offizielle Swagger-Spezifikation ist verfügbar unter <https://swagger.io/specification/>.

## Verwenden des Java REST SDK

Sie können ein Java SDK-Bibliothek verwenden, um Vorgänge für die Orchestrator-REST-API in Java-Anwendungen auszuführen und direkt mit Objekten zu arbeiten.

Jeder REST-ful Webdienst des Orchestrator REST SDK verfügt über eine umhüllende Javaklasse mit Methoden, die den Vorgängen entsprechen, welche durch das Verwenden des Diensts ausgeführt werden.

Das Java REST SDK wird zusammen mit Orchestrator installiert. Die Java REST SDK-Artefakte sind an den folgenden Speicherorten verfügbar:

**Hinweis** Der Zugriff auf die Artefakte ist nur möglich, wenn die Orchestrator-Appliance installiert ist.

- [https://Orchestrator\\_Host:Port/vco-repo/com/vmware/o11n/o11n-rest-client/](https://Orchestrator_Host:Port/vco-repo/com/vmware/o11n/o11n-rest-client/)
- [https://Orchestrator\\_Host:Port/vco-repo/com/vmware/o11n/o11n-rest-client-examples/](https://Orchestrator_Host:Port/vco-repo/com/vmware/o11n/o11n-rest-client-examples/)
- [https://Orchestrator\\_Host:Port/vco-repo/com/vmware/o11n/o11n-rest-client-services/](https://Orchestrator_Host:Port/vco-repo/com/vmware/o11n/o11n-rest-client-services/)
- [https://Orchestrator\\_Host:Port/vco-repo/com/vmware/o11n/o11n-rest-client-stubs/](https://Orchestrator_Host:Port/vco-repo/com/vmware/o11n/o11n-rest-client-stubs/)

## Beispiel: Ausführen eines Workflow und Warten, bis er abgeschlossen ist

Der folgende Beispielcode führt einen Workflow aus und wartet auf seine Beendigung.

```
//start a new session to Orchestrator by using specified credentials
VcoSession session = DefaultVcoSessionFactory.newLdapSession(new URI("https://orchestrator-server:
8281/vco/api/"), "username", "password");

//create the services
WorkflowService workflowService = new WorkflowService(session);
ExecutionService executionService = new ExecutionService(session);

//find a workflow by ID
Workflow workflow = workflowService.getWorkflow("1231235");

//create an ExecutionContext from the user's input
ExecutionContext context = new ExecutionContextBuilder().addParam("name", "Jerry").addParam("age",
18).build();

//run the workflow
WorkflowExecution execution = executionService.execute(workflow, context);

//wait for the workflow to reach the user interaction state, checking every 500 milliseconds
execution = executionService.awaitState(execution, 500, 10, WorkflowExecutionState.CANCELED, WorkflowE-
xecutionState.FAILED, WorkflowExecutionState.COMPLETED);

String nameParamValue = new ParameterExtractor().fromTheOutputOf(execution).extractString("name");
System.out.println("workflow was executed with 'name' input set to" + nameParamValue);
```

## Vorgänge mit Workflows

Die Orchestrator-REST-API stellt Webdienste bereit, mit denen Sie verschiedene Vorgänge mit Workflows ausführen können.

## Suchen eines Workflows und Abrufen seiner Definition

Um einen beliebigen Vorgang auf einen Workflow anzuwenden, müssen Sie zunächst die Orchestrator-Bestandsliste nach diesem durchsuchen und seine Definition abrufen. Die Definition listet die Workflow-Eingabe- und -Ausgabeparameter auf und enthält Links zu den verfügbaren Workflowsausführungen, der Workflowpräsentation und anderen Objekten.

### Voraussetzungen

Überprüfen Sie, ob das Paket mit Beispiel-Workflows in Orchestrator importiert wurde. Das Paket ist in der ZIP-Datei mit Orchestrator-Beispielanwendungen enthalten, die Sie von der Orchestrator-Dokumentationsseite herunterladen können.

### Vorgehensweise

- 1 Suchen Sie nach dem Bestandslistenelement des Workflows.

- Wenn Ihnen der vollständige Name des Workflows oder ein Suchbegriff bekannt ist, führen Sie durch Anwendung eines Filters eine GET-Anforderung unter der URL des Workflowdiensts aus:

```
GET https://{Orchestrator_Host}:{Port}/vco/api/workflows?conditions=name={workflowFullName}
```

```
GET https://{Orchestrator_Host}:{Port}/vco/api/workflows?conditions=name~{keyword}
```

- Durchsuchen Sie den Katalog- oder Bestandslistendienst nach dem Workflow, indem Sie eine GET-Anforderung unter der URL ausführen, die als Einstiegspunkt für die Workflow-Bestandslistenelemente fungiert:

```
GET https://{Orchestrator_Host}:{Port}/vco/api/catalog/System/Workflow/
```

```
GET https://{Orchestrator_Host}:{Port}/vco/api/inventory/System/Workflows/
```

- 2 Rufen Sie das Bestandslistenelement des Workflows ab, indem Sie eine GET-Anforderung unter der URL des Workflows ausführen:

```
GET https://{Orchestrator_Host}:{Port}/vco/api/catalog/System/Workflow/{workflowID}/
```

- 3 Rufen Sie die Definition des Workflows ab, indem Sie eine GET-Anforderung unter der URL der Definition ausführen:

```
GET https://{Orchestrator_Host}:{Port}/vco/api/workflows/{workflowID}/
```

## Beispiel: Suchen nach dem Workflow „Send Hello“

Sie können nach dem Workflow „Send Hello“ suchen und dessen Definition abrufen:

- 1 Führen Sie zur Suche nach dem Workflow „Send Hello“ eine GET-Anforderung unter der URL des Workflowdiensts aus, indem Sie einen Filter anwenden:

```
GET https://localhost:8281/vco/api/workflows?conditions=name~Hello
```

Eine Liste der Workflows mit „Hello“ im Namen wird angezeigt:

```
<xml version="1.0" encoding="UTF-8" standalone="yes">
<inventory-items xmlns="http://www.vmware.com/vco" total="2">
  <link rel="down"
    href="https://localhost:8281/vco/api/catalog/System/Workflow/CF808080808080808080808080808080E6808080013086668236014a0614d16e1/">
    <attributes>
      <attribute name="id" value="CF808080808080808080808080808080E6808080013086668236014a0614d16e1"/>
      <attribute name="canExecute" value="true" />
      <attribute name="description" value="" />
      <attribute name="name" value="Interactive Hello World" />
      <attribute name="type" value="Workflow"/>
      <attribute name="canEdit" value="true"/>
    </attributes>
  </link>
  <link rel="down"
    href="https://localhost:8281/vco/api/catalog/System/Workflow/CF808080808080808080808080808080DA808080013086668236014a0614d16e1/">
    <attributes>
      <attribute name="id" value="CF808080808080808080808080808080DA808080013086668236014a0614d16e1"/>
      <attribute name="canExecute" value="true" />
      <attribute name="description" value="" />
      <attribute name="name" value="Send Hello" />
      <attribute name="type" value="Workflow"/>
      <attribute name="canEdit" value="true"/>
    </attributes>
  </link>
</inventory-items>
```

- 2 Führen Sie eine GET-Anforderung unter der URL des Bestandslistenelements des Workflows „Send Hello“ aus:

```
GET https://localhost:8281/vco/api/catalog/System/Work-  
flow/CF8080808080808080808080808080DA808080013086668236014a0614d16e1/
```



Das Bestandslistenelement des Workflows „Send Hello“ wird im Hauptteil der Antwort angezeigt:

```
<xml version="1.0" encoding="UTF-8" standalone="yes">
<inventory-item xmlns="http://www.vmware.com/vco"
  href="https://localhost:8281/vco/api/catalog/System/Work-
flow/CF808080808080808080808080808080DA808080013086668236014a0614d16e1/">
  <relations>
    <link rel="down"
      href="https://localhost:8281/vco/api/workf-
flows/CF808080808080808080808080808080DA808080013086668236014a0614d16e1/" />
  </relations>
  <attributes>
    <attribute name="id" va-
lue="CF808080808080808080808080808080DA808080013086668236014a0614d16e1"/>
    <attribute name="canExecute" value="true" />
    <attribute name="description" value="" />
    <attribute name="name" value="Send Hello" />
    <attribute name="type" value="Workflow"/>
    <attribute name="canEdit" value="true"/>
  </attributes>
</inventory-item>
```

- 3 Rufen Sie die Definition des Workflows ab, indem Sie eine GET-Anforderung unter deren URL ausführen:

```
GET https://localhost:8281/vco/api/workflows/CF8080808080808080808080808080DA808080013086668236014a0614d16e1/
```

Die Definition des Workflows „Send Hello“ wird im Hauptteil der Antwort angezeigt:

```
<xml version="1.0" encoding="UTF-8" standalone="yes">
<workflow xmlns="http://www.vmware.com/vco" customized-icon="false"
    href="https://localhost:8281/vco/api/workflows/CF8080808080808080808080808080DA808080013086668236014a0614d16e1/">
  <relations>
    <link rel="up"
      href="https://localhost:8281/vco/api/inventory/System/Workflows/Samples>HelloWorld/" />
    <link rel="add"
      href="https://localhost:8281/vco/api/workflows/CF8080808080808080808080808080DA808080013086668236014a0614d16e1/executions/" />
    <link rel="down"
      href="https://localhost:8281/vco/api/workflows/CF8080808080808080808080808080DA808080013086668236014a0614d16e1/executions/" />
    <link rel="down"
      href="https://localhost:8281/vco/api/workflows/CF8080808080808080808080808080DA808080013086668236014a0614d16e1/presentation/" />
    <link rel="down"
      href="https://localhost:8281/vco/api/workflows/CF8080808080808080808080808080DA808080013086668236014a0614d16e1/tasks/" />
    <link rel="down"
      href="https://localhost:8281/vco/api/workflows/CF8080808080808080808080808080DA808080013086668236014a0614d16e1/icon/" />
  </relations>
```

```

<input-parameters>
  <parameter name="name" type="string" />
</input-parameters>
<output-parameters>
  <parameter name="message" type="string" />
</output-parameters>
<name>Send Hello</name>
<description></description>
</workflow>

```

## Ausführen eines Workflows

Ein Workflow wird über die Orchestrator-REST-API ausgeführt, indem ein Ausführungsobjekt für den Workflow erstellt wird.

### Voraussetzungen

Überprüfen Sie, ob das Paket mit Beispiel-Workflows in Orchestrator importiert wurde. Das Paket ist in der ZIP-Datei mit Orchestrator-Beispielanwendungen enthalten, die Sie von der Orchestrator-Dokumentationsseite herunterladen können.

### Vorgehensweise

- 1 Rufen Sie die Definition des Workflows, den Sie ausführen möchten, ab und führen Sie eine GET-Anfrage unter der URL der Definition durch:

```
GET http://{Orchestrator_Host}:{Port}/vco/api/workflows/{workflowID}/
```

Als Antwort auf die Anforderung erhalten Sie die Definition des Workflows. In der Workflowdefinition können Sie die Eingabeparameter des Workflows, die Workflowbeschreibung und weitere Informationen anzeigen.

- 2 Führen Sie eine POST-Anforderung unter der URL aus, die die Ausführungsobjekte des Workflows enthält:

```
POST https://{Orchestrator_Host}:{Port}/vco/api/workflows/{workflowID}/executions/
```

- 3 Geben Sie in einem `execution-context`-Element im Hauptteil der Anforderung Werte für die Eingabeparameter des Workflows an.

Wenn Sie einen leeren `execution-context` im Hauptteil der Anforderung bereitstellen, wird der Workflow mit den Standardwerten für die Eingabeparameter ausgeführt (falls zutreffend).

War die POST-Anforderung erfolgreich, wird der Statuscode 202 mit einem leeren Antworthauptteil und einem Link zum neu erstellten Ausführungsobjekt im `Location`-Header ausgegeben.

## Beispiel: Ausführen des Workflows „Send Hello“

Sie können die Definition des Workflows „Send Hello“ abrufen und diesen ausführen.

- 1 Führen Sie eine GET-Anforderung unter der URL aus, die die Definition des Workflows „Send Hello“ enthält:

```
GET https://localhost:8281/vco/api/workflows/CF8080808080808080808080808080DA808080013086668236014a0614d16e1/
```

Sie erhalten die Workflowdefinition im Hauptteil der Antwortanforderung:

```
<xml version="1.0" encoding="UTF-8" standalone="yes">
<workflow xmlns="http://www.vmware.com/vco" customized-icon="false"
  href="https://localhost:8281/vco/api/workf-
lows/CF8080808080808080808080808080DA808080013086668236014a0614d16e1/">
  <relations>
    <link rel="up"
      href="https://localhost:8281/vco/api/inventory/System/Workflows/Samples/HelloWorld/" />
    <link rel="add"
      href="https://localhost:8281/vco/api/workf-
lows/CF8080808080808080808080808080DA808080013086668236014a0614d16e1/executions/" />
    <link rel="down"
      href="https://localhost:8281/vco/api/workf-
lows/CF8080808080808080808080808080DA808080013086668236014a0614d16e1/executions/" />
    <link rel="down"
      href="https://localhost:8281/vco/api/workf-
lows/CF8080808080808080808080808080DA808080013086668236014a0614d16e1/presentation/" />
    <link rel="down"
      href="https://localhost:8281/vco/api/workf-
lows/CF8080808080808080808080808080DA808080013086668236014a0614d16e1/tasks/" />
    <link rel="down"
      href="https://localhost:8281/vco/api/workf-
lows/CF8080808080808080808080808080DA808080013086668236014a0614d16e1/icon/" />
  </relations>
  <input-parameters>
    <parameter name="name" type="string" />
  </input-parameters>
  <output-parameters>
    <parameter name="message" type="string" />
  </output-parameters>
  <name>Send Hello</name>
  <description></description>
</workflow>
```

- 2 Führen Sie eine POST-Anforderung unter der URL aus, die die Ausführungsobjekte des Workflows enthält:

[illegible]

Übergeben Sie in einem `execution-context`-Element im Hauptteil der Anforderung Werte für die Eingabeparameter an.

```
<execution-context xmlns="http://www.vmware.com/vco">
  <parameters>
    <parameter name="name" type="string">
      <string>John Smith</string>
    </parameter>
  </parameters>
</execution-context>
```

## Ausführen eines Workflows nach der Validierung seiner Eingabeparameter anhand der Workflowpräsentation

In der Präsentation eines Workflows können Optionen für die Werte festgelegt werden, die Sie an die Eingabeparameter des Workflows übergeben, beispielsweise als vordefinierte Werteliste oder bestimmten Wertebereich. Um den erfolgreichen Ablauf des Workflows zu gewährleisten, müssen Sie die an die Eingabeparameter des Workflows übergebenen Werte anhand der Definition der Workflowpräsentation validieren.

Wenn Sie Workflows in benutzerdefinierte Anwendungen integrieren, müssen Sie vielleicht einen Assistenten für die Eingabe von Werten für die Eingabeparameter des Workflows bei seiner Ausführung erstellen. Mit dem Dienst für die Workflowpräsentation können Sie die Präsentation eines Workflows instanziiieren und Werte für seine Eingabeparameter übergeben, die den unterschiedlichen Seiten des Assistenten entsprechen. Die an die Eingabeparameter übergebenen Werte werden anhand der in der Workflowpräsentation festgelegten Optionen validiert.

### Voraussetzungen

Überprüfen Sie, ob das Paket mit Beispiel-Workflows in Orchestrator importiert wurde. Das Paket ist in der ZIP-Datei mit Orchestrator-Beispielanwendungen enthalten, die Sie von der Orchestrator-Dokumentationsseite herunterladen können.

### Vorgehensweise

- 1 Rufen Sie die Definition des auszuführenden Workflows ab, indem Sie eine GET-Anforderung unter der URL ausführen, die die Workflowdefinition enthält, ausführen.

```
GET https://{Orchestrator_Host}:{Port}/vco/api/workflows/{workflowID}/
```

Als Antwort auf die Anforderung erhalten Sie die Definition des Workflows. In der Workflowdefinition können Sie die Eingabeparameter des Workflows, die Workflow-Beschreibung und weitere Informationen anzeigen.

- 2 Rufen Sie die Definition des Workflows ab, indem Sie eine GET-Anforderung unter seiner URL ausführen:

```
GET https://{Orchestrator_Host}:{Port}/vco/api/workflows/{workflowID}/presentation/
```

- 3 Im Hauptteil der Anforderungsantwort prüfen Sie die Definition der Workflowpräsentation auf Optionen für die Werte, die Sie an die Eingabeparameter übergeben können.

So kann ein Eingabeparameter beispielsweise über eine vordefinierte Werteliste verfügen, aus der Werte ausgewählt werden.

- 4 Instanzieren Sie die Workflowpräsentation durch Ausführen einer POST-Anforderung unter der URL der Präsentationsinstanzen:

POST `https://{Orchestrator_Host}:{Port}/vco/api/workflows/{workflowID}/presentation/instances/`

- 5 Geben Sie im Hauptteil der Anforderung ein `execution-context`-Element für die Instanziierung der Präsentation an.

Sie können `execution-context` leer übergeben oder für `execution-context` Werte für einige der Eingabeparameter übergeben.

- 6 Um für die Eingabeparameter in bestimmten Teilen Werte zu übergeben, führen Sie so viele POST- oder PUT-Anforderungen wie nötig unter der URL aus, die die Präsentationsinstanz enthält:

PUT [https://{Orchestrator\\_Host}:{Port}/vco/api/workflows/{workflowID}/presentation/instances/{executionID}/](https://{Orchestrator_Host}:{Port}/vco/api/workflows/{workflowID}/presentation/instances/{executionID}/)

- 7 Prüfen Sie den Hauptteil der Antwort der von Ihnen gestellten POST- oder PUT-Anforderung.

Wenn die von Ihnen an die Eingabeparameter übergebenen Werte gültig sind, sehen Sie bei dem Attribut `valid="true"` das Tag `execution`. Wenn die Präsentation gültig ist, können Sie die im `out-parameters`-Element der Antwort aufgelisteten Werte beim Ausführen des Workflows an die Eingabeparameter übergeben.

- 8 Wenn die Werte für die Eingabeparameter gültig sind, führen Sie den Workflow aus, indem Sie eine POST-Anforderung unter der URL ausführen, die die Workflowsausführungen enthält:

POST `https://{Orchestrator_Host}:{Port}/vco/api/workflows/{workflowID}/executions/`

- 9 Geben Sie die gültigen Werte für die Eingabeparameter des Workflows in einem execution-context-Element an.

## Beispiel: Ausführen des Workflows „Send Hello“ durch Validieren seiner Eingabeparameter

Sie können den Workflow „Send Hello“ ausführen, indem Sie seine Eingabeparameter anhand der Definitionen seiner Präsentation validieren.

- 1 Führen Sie eine GET-Anforderung unter der URL aus, die die Definition des Workflows „Send Hello“ enthält:

```
GET https://localhost:8281/vco/api/workflows/CF8080808080808080808080808080DA808080013086668236014a0614d16e1/
```

Sie erhalten die Workflowdefinition im Hauptteil der Antwortanforderung:

```
<xml version="1.0" encoding="UTF-8" standalone="yes">
<workflow xmlns="http://www.vmware.com/vco" customized-icon="false"
  href="https://localhost:8281/vco/api/workf-
lows/CF8080808080808080808080808080DA808080013086668236014a0614d16e1/">
  <relations>
    <link rel="up"
      href="https://localhost:8281/vco/api/inventory/System/Workflows/Samples/HelloWorld/" />
    <link rel="add"
      href="https://localhost:8281/vco/api/workf-
lows/CF8080808080808080808080808080DA808080013086668236014a0614d16e1/executions/" />
    <link rel="down"
      href="https://localhost:8281/vco/api/workf-
lows/CF8080808080808080808080808080DA808080013086668236014a0614d16e1/executions/" />
    <link rel="down"
      href="https://localhost:8281/vco/api/workf-
lows/CF8080808080808080808080808080DA808080013086668236014a0614d16e1/presentation/" />
    <link rel="down"
      href="https://localhost:8281/vco/api/workf-
lows/CF8080808080808080808080808080DA808080013086668236014a0614d16e1/tasks/" />
    <link rel="down"
      href="https://localhost:8281/vco/api/workf-
lows/CF8080808080808080808080808080DA808080013086668236014a0614d16e1/icon/" />
  </relations>
  <input-parameters>
    <parameter name="name" type="string" />
  </input-parameters>
  <output-parameters>
    <parameter name="message" type="string" />
  </output-parameters>
  <name>Send Hello</name>
  <description></description>
</workflow>
```

- 2 Führen Sie eine GET-Anforderung unter der URL aus, die die Definition der Workflowpräsentation enthält:

```
GET https://localhost:8281/vco/api/workflows/CF808080808080808080808080808080DA808080013086668236014a0614d16e1/presentation/
```

- 3 Führen Sie eine POST-Anforderung unter der URL aus, die die Ausführungsinstanzen der Workflowpräsentation enthält:

```
POST https://localhost:8281/vco/api/workflows/CF80808080808080808080808080808080DA808080013086668236014a0614d16e1/presentation/instances/
```

Stellen Sie einen leeren `execution-context` bereit für die Instanziierung der Präsentation ohne Angabe von Werten für die Eingabeparameter:

```
<execution-context xmlns="http://www.vmware.com/vco"/>
```

Der Hauptteil der Antwort enthält Fehlermeldungen für jedes Feld darüber, dass die Werte für die Eingabeparameter ungültig sind.

```

.....
<fields>
  <field type="string" hidden="false" id="name">
    <display-name>name</display-name>
    <description>name</description>
    <messages>
      <message severity="ERROR" code="VCO-CNS0002">
        <Summary>
          The minimum number of characters allowed for this field is 3.0
        </Summary>
      </message>
    </messages>
  </constraints>
  <number-range max="15.0" min="3.0" />
</constraints>
.....

```

- 4 Führen Sie eine POST-Anforderung unter der URL aus, die die entsprechende Präsentationsinstanz enthält:

```
POST https://localhost:8281/vco/api/workflows/CF808080808080808080808080808080DA80808001308668236014a0614d16e1/presentation/instances/8880808080808080808080808080803F8080800132145338690643f66a027ec/
```

Geben Sie im Hauptteil der Anforderung Werte für die Eingabeparameter an:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<execution-context xmlns="http://www.vmware.com/vco">
  <parameters>
    <parameter name="name" type="string">
      <string>John Smith</string>
    </parameter>
  </parameters>
</execution-context>
```

Im Hauptteil der Anforderungsantwort können Sie prüfen, ob die Werte für die Eingabeparameter gültig sind:

```
<execution started-by="vcoadmin" .... valid="true".....>
```

- 5 Wenn die Präsentation gültig ist, führen Sie den Workflow aus, indem Sie eine POST-Anforderung unter der URL ausführen, die die Workflowausführungen enthält:

[illegible]

Übergeben Sie im Hauptteil der Anforderung die Werte für die Eingabeparameter des Workflows. Verwenden Sie dieselben Werte, die als Ausgabeparameter der Workflowpräsentation zurückgegeben wurden, oder verwenden Sie direkt den Hauptteil der letzten POST-Anforderung, die Sie für die Workflowpräsentation ausgeführt haben.

## Interaktion mit laufenden Workflows

Mit der Orchestrator-REST-API können Sie verschiedene Vorgänge an einem Workflow ausführen, während dieser läuft. Sie können den Status eines laufenden Workflows abrufen, eine wartende Benutzerinteraktion beantworten und eine Workflowausführung beenden.

### Abrufen von Workflowausführungsobjekten und Überprüfen des Workflowstatus

Sie können Informationen zur Ausführung eines Workflows abrufen, zum Beispiel Start- und Enddatum, Status der Ausführung und die Werte für die Eingabeparameter. Außerdem können Protokolle abgerufen werden, die zur Workflowausführung generiert werden.

#### Voraussetzungen

Überprüfen Sie, ob das Paket mit Beispiel-Workflows in Orchestrator importiert wurde. Das Paket ist in der ZIP-Datei mit Orchestrator-Beispielanwendungen enthalten, die Sie von der Orchestrator-Dokumentationsseite herunterladen können.

#### Vorgehensweise

- 1 Führen Sie eine GET-Anforderung unter der URL des Workflows aus, dessen Status Sie überprüfen möchten, um dessen Definition abzurufen:

```
GET https://{Orchestrator_Host}:{Port}/vco/api/workflows/{workflowID}/
```

Als Antwort auf die Anforderung erhalten Sie die Definition des Workflows. Die Workflowdefinition enthält einen Link zu den Ausführungsinstanzen des Workflows.

- 2 Rufen Sie die verfügbaren Ausführungsinstanzen des Workflows ab, indem Sie eine GET-Anforderung unter der URL der Workflowdefinition ausführen:

```
GET https://{Orchestrator_Host}:{Port}/vco/api/workflows/{workflowID}/executions/
```

Im Hauptteil der Antwort auf die Anforderung werden die verfügbaren Ausführungsinstanzen des Workflows aufgeführt. Hier können Sie Start- und Enddatum der einzelnen Workflowausführungen sowie deren Status und Initiator überprüfen.



- 3 (Optional) Führen Sie eine GET-Anforderung unter der URL einer bestimmten Ausführung eines Workflows aus, um Details zu dieser Ausführung zu erhalten:

GET https://{Orchestrator\_Host}:{Port}/vco/api/workflows/{workflowID}/executions/{executionID}/

Im Hauptteil der Antwort auf die Anforderung erhalten Sie die XML-Darstellung dieser bestimmten Workflowausführung. Sie können die Werte der Eingabeparameter dieser Ausführung, den ausführenden Benutzer, Start- und Enddatum und Status der Ausführung überprüfen.

- 4 (Optional) Führen Sie zum Abrufen der für diese Workflowausführung generierten Protokolle eine GET-Anforderung unter der URL mit den Protokollen aus:

```
GET https://{Orchestrator_Host}:{Port}/vco/api/workflows/{workflowID}/executions/{execution-
ID}/logs/
```

- 5 (Optional) Führen Sie zum Abrufen zusätzlicher Informationen zum Status der Ausführung eine GET-Anforderung unter der URL aus, die den Status des Workflows enthält:

```
GET https://{Orchestrator_Host}:{Port}/vco/api/workflows/{workflowID}/executions/{executionID}/state/
```

### Beispiel: Abrufen der Ausführungen des Workflows „Send Hello“ und Überprüfen des Status einer bestimmten Ausführung

Nach Ausführen des Workflows „Send Hello“ können die verfügbaren Ausführungsobjekte abgerufen und deren Details überprüft werden.

- 1 Führen Sie eine GET-Anforderung unter der URL aus, die die Definition des Workflows „Send Hello“ enthält, um diese abzurufen:

```
GET https://localhost:8281/vco/api/workf-  
lows/CF8080808080808080808080808080DA808080013086668236014a0614d16e1/
```

- 2 Führen Sie eine GET-Anforderung unter der URL aus, die die Ausführungsobjekte des Workflows enthält, um die entsprechenden Ausführungen abzurufen:

```
GET https://localhost:8281/vco/api/workflows/CF8080808080808080808080808080DA808080013086668236014a0614d16e1/executions/
```

- 3 Wählen Sie aus dem Hauptteil der Antwort auf die Anforderung eine Workflowausführung aus und führen Sie eine GET-Anforderung aus, um sie abzurufen:

```
GET https://localhost:8281/vco/api/workflows/CF8080808080808080808080808080DA808080013086668236014a0614d16e1/executions/8880808080808080808080808080803A8080800132145338690643f66a027ec/
```

Der Hauptteil der Antwort enthält die XML-Darstellung der Workflowausführung mit der eindeutigen ID, unter der Details zur Ausführung überprüft werden können:

```
.....
<input-parameters>
  <parameter name="name" type="string">
    <string>John Smith</string>
  </parameter>
</input-parameters>
<output-parameters>
  <parameter name="message" type="string">
    <string>Hello, John Smith!</string>
  </parameter>
</output-parameters>
<start-date>2012-01-31T14:28:40.223+03:00</start-date>
<end-date>2012-01-31T14:28:40.410+03:00</end-date>
<started-by>vcadmin</started-by>
<name>Send Hello</name>
.....
```

## Antwort auf eine wartende Benutzerinteraktion

Mithilfe der Orchestrator-REST-API können Sie auf eine wartende Benutzerinteraktion einer Workflowausführung antworten.

### Voraussetzungen

Überprüfen Sie, ob das Paket mit Beispiel-Workflows in Orchestrator importiert wurde. Das Paket ist in der ZIP-Datei mit Orchestrator-Beispielanwendungen enthalten, die Sie von der Orchestrator-Dokumentationsseite herunterladen können.

### Vorgehensweise

- 1 Rufen Sie die Liste aller Benutzerinteraktionsobjekte ab, indem Sie eine GET-Anforderung unter der URL ausführen, unter der die verfügbaren Benutzerinteraktionsobjekte gespeichert sind, oder indem Sie nur die wartenden Benutzerinteraktionen filtern:

URL	Beschreibung
<b>https://Orchestrator_Host:Port/vco/api/catalog/System/UserInteraction</b>	Hier sind die verfügbaren Benutzerinteraktionsobjekte in Orchestrator gespeichert.
<b>https://Orchestrator_Host:Port/vco/api/catalog/System/UserInteraction?status=0</b>	Nur die wartenden Benutzerinteraktionsobjekte werden gefiltert.

Eine Liste der verfügbaren Benutzerinteraktionsobjekte wird ausgegeben. Wartende Benutzerinteraktionen weisen ein Attribut mit den Namen `state` und dem Wert `waiting` auf.

- 2 Führen Sie eine GET-Anforderung unter der URL aus, unter der das Bestandslistenelement der wartenden Benutzerinteraktion gespeichert ist, auf die Sie antworten möchten:

GET https://{Orchestrator\_Host}:{Port}/vco/api/catalog/System/UserInteraction/{userInteractionID}/

Das Bestandslistenelement enthält einen Link zu der Benutzerinteraktionsinstanz. Diese ist mit einer bestimmten Workflowausführung verknüpft.

- 3 Führen Sie eine POST-Anforderung unter der URL der Benutzerinteraktionsinstanz für die betreffende Workflowausführung aus:

POST [https://{Orchestrator\\_Host}:{Port}/vco/api/workflows/{workflowID}/executions/{executionID}/interaction/](https://{Orchestrator_Host}:{Port}/vco/api/workflows/{workflowID}/executions/{executionID}/interaction/)

- 4 Geben Sie in einem `execution-context`-Element im Hauptteil der Anforderung Werte für die Eingabeparameter der Benutzerinteraktion an.

Die REST-API gibt den Status 204 zurück, wenn Sie erfolgreich auf eine Benutzerinteraktion antworten.

### Beispiel: Antworten auf die Benutzerinteraktion des interaktiven Workflows „Hello World“

Sie können den interaktiven Beispiel-Workflow „Hello World“ ausführen und auf dessen Benutzerinteraktion antworten.

- 1 Suchen Sie nach der wartenden Benutzerinteraktion des Workflows, indem Sie am Endpoint für die Benutzerinteraktionsobjekte des Katalogdienstes eine GET-Anforderung ausführen:

```
GET https://localhost:8281/vco/api/catalog/System/UserInteraction?status=0
```

- 2 Suchen Sie nach dem Bestandslistenelement „Benutzerinteraktion“ für den interaktiven Workflow „Hello World“ und führen Sie unter dessen URL eine GET-Anforderung aus:

[illegible]

- 3 Führen Sie eine POST-Anforderung unter der URL der Benutzerinteraktionsobjekte für die derzeit laufende Workflowausführung aus:

```
POST https://localhost:8281/vco/api/workflows/CF808080808080808080808080E6808080013086668236014a0614d16e1/executions/888080808080808080808080808080578080800132145338690643f66a027ec/interaction/
```

Geben Sie im Hauptteil der Anforderung einen Wert für den Eingabeparameter an:

```
<execution-context xmlns="http://www.vmware.com/vco">
  <parameters>
    <parameter name="name" type="string">
      <string>John Smith</string>
    </parameter>
  </parameters>
</execution-context>
```

## Antwort auf eine Benutzerinteraktion nach Validierung der Eingabeparameter

In der Präsentation einer Benutzerinteraktion können Optionen für die Werte festgelegt werden, die Sie an die Eingabeparameter des Workflows übergeben. Wenn Sie auf eine Benutzerinteraktion antworten, können Sie die an die Eingabeparameter übergebenen Werte anhand der in der Workflowpräsentation festgelegten Optionen validieren.

### Voraussetzungen

Überprüfen Sie, ob das Paket mit Beispiel-Workflows in Orchestrator importiert wurde. Das Paket ist in der ZIP-Datei mit Orchestrator-Beispielanwendungen enthalten, die Sie von der Orchestrator-Dokumentationsseite herunterladen können.

### Vorgehensweise

- 1 Rufen Sie die Liste aller Benutzerinteraktionsobjekte ab, indem Sie eine GET-Anforderung unter der URL ausführen, unter der die verfügbaren Benutzerinteraktionsobjekte gespeichert sind, oder indem Sie nur die wartenden Benutzerinteraktionen filtern:

URL	Beschreibung
<b>https://Orchestrator_Host:Port/vco/api/catalog/System/UserInteraction</b>	Hier sind die verfügbaren Benutzerinteraktionsobjekte in Orchestrator gespeichert.
<b>https://Orchestrator_Host:Port/vco/api/catalog/System/UserInteraction?status=0</b>	Nur die wartenden Benutzerinteraktionsobjekte werden gefiltert.

Eine Liste der verfügbaren Benutzerinteraktionsobjekte wird ausgegeben. Wartende Benutzerinteraktionen weisen ein Attribut mit den Namen `state` und dem Wert `waiting` auf.

- 2 Führen Sie eine GET-Anforderung unter der URL aus, unter der das Bestandslistenelement der wartenden Benutzerinteraktion gespeichert ist, auf die Sie antworten möchten:

```
GET https://{Orchestrator_Host}:{Port}/vco/api/catalog/System/UserInteraction/{userInteractionID}/
```

Der Hauptteil der Antwort enthält einen Link zur Benutzerinteraktionsinstanz. Die Benutzerinteraktionsinstanz ist mit einer bestimmten Workflowausführung verknüpft.

- 3 Führen Sie eine GET-Anforderung unter der URL der Benutzerinteraktion aus:

```
GET https://{Orchestrator_Host}:{Port}/vco/api/workflows/{workflowID}/executions/{executionID}/interaction/
```

Im Hauptteil der Antwort finden Sie einen Link zur Präsentation der Benutzerinteraktion.

- 4 Führen Sie eine GET-Anforderung unter der URL der Präsentation der Benutzerinteraktion aus:

```
GET https://{Orchestrator_Host}:{Port}/vco/api/workflows/{workflowID}/executions/{executionID}/interaction/presentation/
```

Im Hauptteil der Antwort erhalten Sie die Definition der Benutzerinteraktions-Präsentation.

- 5 Suchen Sie in der Präsentationsdefinition nach Optionen für die Werte, die Sie an die Eingabeparameter übergeben können.
- 6 Führen Sie die Benutzerinteraktions-Präsentation aus, indem Sie eine POST-Anforderung unter der URL ausführen, unter der sich die Instanzen der Präsentation befinden:

```
POST https://{Orchestrator_Host}:{Port}/vco/api/workflows/{workflowID}/executions/{executionID}/interaction/presentation/instances/
```

- 7 Geben Sie im Hauptteil der Anforderung Werte für die Eingabeparameter in einem execution-context-Element an.

Im Hauptteil der Antwort erhalten Sie die Instanz der Benutzerinteraktions-Präsentation. Wenn die von Ihnen an die Eingabeparameter übergebenen Werte gültig sind, sehen Sie das Attribut `valid="true"` im Element `execution`. Im Element `output-parameters` finden Sie die gültigen Werte für die Eingabeparameter, die Sie verwenden können, um auf die Benutzerinteraktion zu antworten.

- 8 Antworten Sie auf die Benutzerinteraktion, indem Sie eine POST-Anforderung unter der URL ausführen, unter der sich die Benutzerinteraktionsinstanz befindet:

```
POST https://{Orchestrator_Host}:{Port}/vco/api/workflows/{workflowID}/executions/{executionID}/interaction/
```

- 9 Übergeben Sie im Hauptteil der Anforderung einen execution-context-Kontext mit den Werten für die Eingabeparameter.

Sie können denselben Hauptteil der Anforderung verwenden wie den für die POST-Anforderung, die Sie unter der URL für die Benutzerinteraktions-Präsentation ausgeführt haben.

Wenn die letzte Anforderung erfolgreich war, erhalten Sie einen Statuscode 204 und einen leeren Antworthauptteil.

### Beispiel: Antworten auf die Benutzerinteraktion des interaktiven Workflows „Hello World“ durch Validierung der Eingabeparameter

Sie können auf die Benutzerinteraktion des interaktiven Workflows „Hello World“ antworten, indem Sie die Werte der Eingabeparameter anhand der in der Präsentation der Benutzerinteraktion festgelegten Optionen validieren.

- 1 Suchen Sie nach den wartenden Benutzerinteraktionen des Workflows, indem Sie am Endpoint für die Benutzerinteraktionsobjekte des Katalogdienstes eine GET-Anforderung ausführen:

```
GET https://localhost:8281/vco/api/catalog/System/UserInteraction?status=0
```

- 2 Suchen Sie nach dem Bestandslistenelement „Benutzerinteraktion“ für den interaktiven Workflow „Hello World“ und führen Sie unter dessen URL eine GET-Anforderung aus:

[illegible]

- 3 Führen Sie eine GET-Anforderung unter der URL der Benutzerinteraktion aus:

[illegible]

- 4 Führen Sie eine GET-Anforderung unter der URL der Benutzerinteraktions-Präsentation aus:

[illegible]

Die Präsentation definiert die Eingabeparameter als obligatorisch und enthält eine Option für die Länge der Zeichenfolge, die Sie übergeben können.

- 5 Führen Sie eine POST-Anforderung unter der URL aus, die die Instanzen der Benutzerinteraktions-Präsentation enthält:

[illegible]

Geben Sie im Hauptteil der Anforderung einen Wert für den Eingabeparameter an:

```
<execution-context xmlns="http://www.vmware.com/vco">
  <parameters>
    <parameter name="name" type="string">
      <string>John Smith</string>
    </parameter>
  </parameters>
</execution-context>
```

Das execution-Element des Hauptteils der Antwort enthält ein valid="true"-Attribut, das angibt, dass der Eingabeparameterwert anhand der Optionen in der Benutzerinteraktions-Präsentation gültig ist. Der gültige Wert wird im Element output-parameters aufgelistet:

```
.....
<output-parameters>
  <parameter name="name" type="string">
    <string>John Smith</string>
  </parameter>
</output-parameters>
.....
```

- 6 Führen Sie eine POST-Anforderung unter der URL der Benutzerinteraktionsinstanz aus, indem Sie denselben Hauptteil der Anforderung wie in der POST-Anforderung in Schritt 5 übergeben.

[illegible]

## Abbrechen einer Workflowausführung

Sie können eine Workflowausführung mithilfe der Orchestrator-REST-API importieren.

## Vorgehensweise

- 1 Rufen Sie die Definition des Workflows ab, indem Sie eine GET-Anforderung unter der URL der Workflowdefinition ausführen:

GET https://{Orchestrator\_Host}:{Port}/vco/api/workflows/{workflowID}/

Die Workflowdefinition enthält einen Link zu den verfügbaren Ausführungsobjekten des Workflows.

- 2 Führen Sie eine GET-Anforderung unter der URL aus, die die Ausführungsobjekte des Workflows enthält, um die verfügbaren Workflowausführungen abzurufen:

GET https://{Orchestrator\_Host}:{Port}/vco/api/workflows/{workflowID}/executions/

- 3** Wählen Sie in der Liste der verfügbaren Workflowausführungen die aus, die Sie abbrechen möchten, und führen Sie eine DELETE-Anforderung unter der URL durch:

```
DELETE https://{Orchestrator_Host}:{Port}/vco/api/workflows/{workflowID}/executions/{execution-
ID}/state
```

## Abrufen der Interaktionen eines Workflows

Mithilfe der Orchestrator-REST-API können Sie die Liste aller Benutzerinteraktionen für einen Workflow abrufen.

### Vorgehensweise

- 1 Rufen Sie die Definition des Workflows ab, indem Sie eine GET-Anforderung unter der URL der Workflowdefinition ausführen:

```
GET https://{Orchestrator_Host}:{Port}/vco/api/workflows/{workflowID}/
```

- 2 Rufen Sie die Liste der Workflowinteraktionen ab, indem Sie eine GET-Anforderung unter der URL der Workflowinteraktionen ausführen:

```
GET https://{Orchestrator_Host}:{Port}/vco/api/workflows/{workflowID}/interactions/
```

War die GET-Anforderung erfolgreich, werden der Statuscode 200 und eine Liste aller für den Workflow verfügbaren Benutzerinteraktionen ausgegeben.

## Zugriff auf das Schema eines Workflows

Mithilfe der REST-API von Orchestrator können Sie auf das Schemaimage eines Workflows zugreifen.

### Vorgehensweise

- 1 Rufen Sie die Definition des Workflows ab, indem Sie eine GET-Anforderung unter der URL der Workflowdefinition ausführen:

```
GET https://{Orchestrator_Host}:{Port}/vco/api/workflows/{workflowID}/
```

- 2 Rufen Sie das Schemaimage des Workflows ab, indem Sie eine GET-Anforderung unter der URL des Workflowschemas ausführen:

```
GET https://{Orchestrator_Host}:{Port}/vco/api/workflows/{workflowID}/schema/
```

Wenn die GET-Anforderung erfolgreich ist, erhalten Sie den Statuscode 200 und die binären Daten des Images, die das Workflowschema repräsentieren. Der Inhaltstyp der Antwort wird auf einen richtigen Medientyp festgelegt, z. B. Content-Type: image/png.

## Arbeiten mit Aufgaben

Mit dem Aufgabendienst der Orchestrator-REST-API können Sie alle Vorgänge ausführen, die mit dem Verwalten von Aufgaben in Orchestrator zusammenhängen. Sie können eine Aufgabe für das Planen eines Workflows erstellen, die Eigenschaften einer bereits vorhandenen Aufgabe ändern, eine Aufgabe löschen und so weiter.

Die maximale Anzahl von geplanten Aufgaben, die von Orchestrator unterstützt wird, ist 50.

## Erstellen von Aufgaben

Mithilfe der REST-API von Orchestrator können Sie eine Aufgabe zur Zeitplanung von Workflows erstellen.



## Voraussetzungen

Überprüfen Sie, ob das Paket mit Beispiel-Workflows in Orchestrator importiert wurde. Das Paket ist in der ZIP-Datei mit Orchestrator-Beispielanwendungen enthalten, die Sie von der Orchestrator-Dokumentationsseite herunterladen können.

## Vorgehensweise

- 1 Rufen Sie die Definition des Workflows, für den Sie eine Aufgabe erstellen möchten, ab und führen Sie eine GET-Anforderung unter der URL des Workflows durch:

GET https://{Orchestrator\_Host}:{Port}/vco/api/workflows/{workflowID}/

In der Workflowdefinition können Sie den Namen und die ID des Workflows sowie seine Eingabeparameter anzeigen.

- 2 Um eine neue Aufgabe für den Workflow zu erstellen, führen Sie eine POST-Anforderung unter der URL des Aufgabendienstes aus:

POST https://{Orchestrator\_Host}:{Port}/vco/api/tasks/

- 3 Geben Sie im Hauptteil der Anforderung die Parameter für die neue Aufgabe in einem task-Element an.

Bei einer erfolgreichen Anforderung antwortet die API mit dem Statuscode 202 und einem leeren Hauptteil.

### Beispiel: Erstellen einer Aufgabe für den Workflow „Send Hello“

Sie können eine Aufgabe zur Zeitplanung des Workflows „Send Hello“ erstellen, die in der fünfzehnten Minute jeder Stunde ausgeführt wird und ab einem bestimmten Datum startet.

- 1 Führen Sie eine GET-Anforderung unter der URL des Workflows „Send Hello“ aus, um dessen Definition abzurufen:

```
GET https://localhost:8281/vco/api/workflows/CF808080808080808080808080808080DA808080013086668236014a0614d16e1/
```

- 2 Führen Sie eine POST-Anforderung unter der URL des Aufgabendienstes aus und geben Sie dabei die Parameter der neuen Aufgabe im Hauptteil der Anforderung an:

POST https://localhost:8281/vco/api/tasks/

```
<task xmlns="http://www.vmware.com/vco">
  <name>Send Hello Task</name>
  <recurrence-cycle>every-hours</recurrence-cycle>
  <recurrence-start-date>2012-01-31T11:00:00+00:00</recurrence-start-date>
  <recurrence-end-date>2012-02-05T11:00:00+00:00</recurrence-end-date>
  <recurrence-pattern>15:15</recurrence-pattern>
  <input-parameters>
```

```
<parameter name="name" type="string">
  <string>John Smith</string>
</parameter>
</input-parameters>
<workflow href="https://localhost:8281/vco/api/workflows/CF8080808080808080808080808080DA808080013086668236014a0614d16e1/">
  <name>Send Hello</name>
</workflow>
<start-mode>normal</start-mode>
</task>
```

## Ändern einer Aufgabe

Sie können die Eigenschaften einer vorhandenen Aufgabe mithilfe der Orchestrator-REST-API ändern.

Sie können einer Aufgabe nur neue Planungsaufgaben hinzufügen oder die Werte der bereits vorhandenen Eigenschaften ändern. Wenn Sie die Planungseigenschaften einer Aufgabe ersetzen möchten, müssen Sie die Aufgabe löschen und eine neue erstellen.

## Voraussetzungen

Überprüfen Sie, ob das Paket mit Beispiel-Workflows in Orchestrator importiert wurde. Das Paket ist in der ZIP-Datei mit Orchestrator-Beispielanwendungen enthalten, die Sie von der Orchestrator-Dokumentationsseite herunterladen können.

## Vorgehensweise

- 1 Führen Sie eine GET-Anforderung unter der URL der Aufgabe aus, die Sie ändern möchten:

GET https://{Orchestrator\_Host}:{Port}/vco/api/tasks/{Aufgaben-ID}/

- 2 Überprüfen Sie die Eigenschaften der Aufgabe im Hauptteil der Antwort auf die Anforderung.
- 3 Wenn Sie eine Aufgabe ändern möchten, führen Sie eine POST-Anforderung unter der URL der Aufgabe aus und geben Sie dabei die neuen Eigenschaften der Aufgabe in einem `task-data`-Element des Hauptteils der Anforderung an.

War die POST-Anforderung erfolgreich, gibt die API den Statuscode 200 und die aktualisierte Aufgabe im Hauptteil der Antwort zurück.

## Beispiel: Aktualisieren der Beispielaufgabe „Send Hello“

Sie können die Start- und Enddaten einer Aufgabe aktualisieren. Sie können die in [Erstellen von Aufgaben](#) vorgestellte Beispielaufgabe ändern. Sie müssen eine POST-Anforderung unter der URL der Aufgabe ausführen und dabei das neue Start- und Enddatum im Hauptteil der Anforderung eingeben:

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<task-data xmlns="http://www.vmware.com/vco">
  <recurrence-start-date>2012-02-01T14:00:00+02:00</recurrence-start-date>
  <recurrence-end-date>2012-02-05T14:00:00+02:00</recurrence-end-date>
</task-data>
```

## Überprüfen des Status einer Aufgabe

Sie können den Status der momentan verfügbaren Aufgaben oder den Status aller Ausführungsinstanzen einer bestimmten Aufgabe überprüfen.

### Voraussetzungen

Überprüfen Sie, ob das Paket mit Beispiel-Workflows in Orchestrator importiert wurde. Das Paket ist in der ZIP-Datei mit Orchestrator-Beispielanwendungen enthalten, die Sie von der Orchestrator-Dokumentationsseite herunterladen können.

### Vorgehensweise

- Führen Sie eine GET-Anforderung unter der URL des Aufgabendienstes aus, um den Status aller momentan verfügbaren Aufgaben zu überprüfen:

```
GET https://{Orchestrator_Host}:{Port}/vco/api/tasks/
```

Der Hauptteil der Antwort enthält die Definitionen der momentan in Orchestrator verfügbaren Aufgaben. Der Status jeder Aufgabe ist in einem attribute-Element verfügbar, dessen Name `state` lautet. Dementsprechend kann der Wert für das Element `finished`, `pending`, `running` etc. sein.

- Um den Status aller Ausführungen einer bestimmten Aufgabe zu überprüfen, führen Sie eine GET-Anforderung unter der URL aus, unter der sich die Aufgabenausführung befindet:

```
GET https://{Orchestrator_Host}:{Port}/vco/api/tasks/{taskID}/executions/
```

Sie erhalten eine Liste verfügbarer Ausführungen für diese Aufgabe im Hauptteil der Antwort. Der Status jeder Ausführung ist im `state`-Element des Aufgabenausführungsobjekts verfügbar.

## Suchen von Objekten in der Orchestrator-Bestandsliste

Sie können beliebige Objekte in einer Orchestrator-Bestandsliste suchen, indem Sie die Katalog- oder Bestandslistendienste verwenden. Sie können nur auf eine bestimmte Teilmenge von Objekten zugreifen, indem Sie Filterparameter am Ende der URLs anwenden, unter denen Sie HTTP-Anforderungen ausführen.

Sie können den Katalogdienst zum Suchen von Objekten eines bestimmten Typs in der Orchestrator-Bestandsliste verwenden oder ein bestimmtes Objekt nach seinem Typ und seiner ID abrufen. Beispielsweise können Sie alle Objekte, die vom Typ `workflow` oder `action` sind, abrufen, oder Sie können einen bestimmten Workflow oder eine bestimmte Aktion abrufen.

Mit dem Bestandslistendienst können Sie die Orchestrator-Bestandsliste nach übergeordneten/untergeordneten Elementen durchsuchen. Unter Verwendung des Bestandslistendienstes können Sie auf Objekte zugreifen, die an einem bestimmten Ort in der Orchestrator-Bestandsliste verfügbar sind. Beispielsweise können Sie alle Workflows für die Datacenter-Verwaltung abrufen, indem Sie zum Speicherort in der Orchestrator-Bestandsliste navigieren: `Library/vCenter/Datacenter`.

Alle Dienste der Orchestrator-REST-API unterstützen Filterparameter, die Sie beim Ausführen von HTTP-Anforderungen am Ende der URLs hinzufügen können. Durch die Verwendung von Filterparametern können Sie die Ergebnisse, die Sie im Hauptteil der Antwort unter einer bestimmten URL erhalten, einschränken.

## Durchsuchen von Objekten nach Typ und ID

Mithilfe des Katalogdiensts der REST-API können Sie die Objekte in Orchestrator nach Typs und ID suchen.

### Voraussetzungen

Überprüfen Sie, ob das Paket mit Beispiel-Workflows in Orchestrator importiert wurde. Das Paket ist in der ZIP-Datei mit Orchestrator-Beispielanwendungen enthalten, die Sie von der Orchestrator-Dokumentationsseite herunterladen können.

### Vorgehensweise

- 1 Führen Sie eine GET-Anforderung unter der URL des Katalogdiensts aus:

```
GET https://{Orchestrator_Host}:{Port}/vco/api/catalog/
```

Der Hauptteil der Antwort auf die Anforderung enthält Downlinks zu den Katalogeinstiegspunkten der Plug-Ins, deren Bestandslisten in Orchestrator verfügbar sind, sowie zu den Systemobjekten in Orchestrator:

- `https://{Orchestrator_Host}:{Port}/vco/api/catalog/{Plug-In-Namespace}/`
- `https://{Orchestrator_Host}:{Port}/vco/api/catalog/System/`

- 2 Führen Sie zum Zugriff auf Objekte, die von Plug-Ins verfügbar gemacht werden, oder die Systemobjekte in Orchestrator eine GET-Anforderung unter der URL des Katalogeinstiegspunkts für das Plug-In bzw. die Systemobjekte in Orchestrator aus.

Der Hauptteil der Antwort auf die Anforderung enthält Links zu den verfügbar gemachten Objekttypen.

- 3 Führen Sie eine GET-Anforderung unter der URL des Objekttyps aus, auf den Sie zugreifen möchten.

```
GET https://{Orchestrator_Host}:{Port}/vco/api/catalog/{Namespace}/{objectType}/
```

- 4 Führen Sie eine GET-Anforderung unter der URL des bestimmten Objekts aus, das Sie suchen:

```
GET https://{Orchestrator_Host}:{Port}/vco/api/catalog/{Namespace}/{objectType}/{objectId}/
```

## Beispiel: Suchen des Workflows „Send Hello“

Mithilfe des Katalogdiensts können Sie nach dem Beispiel-Workflow „Send Hello“ suchen.

- 1 Führen Sie eine GET-Anforderung unter der URL des Katalogdiensts aus:

```
GET https://localhost:8281/vco/api/catalog/
```

- 2 Führen Sie eine GET-Anforderung unter der URL aus, wo sich sämtliche Systemobjekte in Orchestrator befinden:

```
GET https://localhost:8281/vco/api/catalog/System/
```

- 3 Führen Sie eine GET-Anforderung unter der URL aus, unter der sich sämtliche Workflows befinden:

```
GET https://localhost:8281/vco/api/catalog/Workflow/
```

- 4 Führen Sie eine GET-Anforderung unter der URL des Workflows „Send Hello“ aus:

```
GET https://localhost:8281/vco/api/catalog/Work-  
flow/CF8080808080808080808080808080DA808080013086668236014a0614d16e1/
```

## Suchen von Objekten nach Beziehungen

Mithilfe des Inventory Service der Orchestrator-REST-API können Sie die Orchestrator- und Plug-In-Be-  
standslisten als Hierarchie durchsuchen.

## Voraussetzungen

Überprüfen Sie, ob das Paket mit Beispiel-Workflows in Orchestrator importiert wurde. Das Paket ist in der ZIP-Datei mit Orchestrator-Beispielanwendungen enthalten, die Sie von der Orchestrator-Dokumentationsseite herunterladen können.

## Vorgehensweise

- 1 Führen Sie eine GET-Anforderung unter der URL des Inventory Service aus:

```
GET https://{Orchestrator_Host}:{Port}/vco/api/inventory/
```

Der Hauptteil der Antwort enthält Downlinks zu den registrierten Bestandslisten der installierten Plugins sowie zu den Systemobjekten in Orchestrator unter „System“.

- 2 Führen Sie eine GET-Anforderung unter dem Downlink der Bestandsliste aus, auf die Sie zugreifen möchten.
- 3 Führen Sie solange GET-Anforderungen unter den Up- und Downlinks der Bestandslistenelemente aus, bis Sie das gesuchte Objekt gefunden haben.

## Beispiel: Suchen des Workflows „Send Hello“

Sie können die Orchestrator-Bestandsliste nach dem Workflow „Send Hello“ durchsuchen.

- 1 Führen Sie eine GET-Anforderung unter der URL des Inventory Service aus:

```
GET https://localhost:8281/vco/api/inventory/
```

- 2 Führen Sie eine GET-Anforderung unter der URL aus, unter der sich die Systemobjekte in Orchestrator befinden:

```
GET https://localhost:8281/vco/api/inventory/System/
```

- 3 Führen Sie eine GET-Anforderung unter der URL aus, unter der sich sämtliche Workflows in Orchestrator befinden:

```
GET https://localhost:8281/vco/api/inventory/System/Workflows/
```

- 4 Führen Sie eine GET-Anforderung unter der URL der Workflow-Kategorie „Beispiele“ aus:

```
GET https://localhost:8281/vco/api/inventory/System/Workflows/Samples/
```

- 5 Verwenden Sie den Downlink der Workflowkategorie „Hello World“, unter dem sich der Workflow „Send Hello“ befindet.

## Anwenden von Filtern

Die Dienste der Orchestrator-REST-API unterstützen zusätzliche URL-Parameter, mit deren Hilfe Sie die Objekte eingrenzen können, die auf HTTP-Anforderungen an die API zurückgegeben werden.

Für die URL zu einer Ressource, auf die Sie über die REST-API zugreifen können, werden jeweils unterschiedliche Abfrageparameter unterstützt. Weitere Informationen zu den Abfrageparametern, die für eine bestimmte URL gelten, finden Sie in der Referenzdokumentation *vRealize Orchestrator-REST-API*.

### Vorgehensweise

- ◆ Wenden Sie zum Eingrenzen der Ergebnisse einer Anforderung für eine bestimmte URL Filter auf das Ende der URL an:  
  
*URL?Filter\_1&Filter\_2&Filter\_3&....&Filter\_N*. Jeder Filter enthält Abfrageparameter, die für die jeweilige URL gültig sind. Weitere Informationen zu allgemein gültigen Abfrageparametern für URLs finden Sie in der Referenzdokumentation zur Orchestrator-REST-API.

## Beispiel: Filtern von Workflows

Wenn Sie nach einem Workflow mit einem bestimmten Wort im Namen suchen (zum Beispiel „Datenspeicher“), können Sie folgenden Filter auf die entsprechende Anforderung an den Katalogdienst anwenden:

```
GET https://localhost:8281/vco/api/catalog/System/Workflow?conditions=name~datastore
```

Wenden Sie zum Einschränken der zurückgegebenen Workflows auf eine bestimmte Anzahl (zum Beispiel fünf) einen weiteren Filter auf die Anforderung an:

```
GET https://localhost:8281/vco/api/catalog/System/Workflow?conditions=name~datastore&maxResult=5
```

## Importieren und Exportieren eines Orchestrator-Objekts

Die Orchestrator-REST-API stellt Webdienste bereit, mit denen Sie Workflows, Aktionen, Pakete, Ressourcen und Konfigurationselemente importieren und exportieren können.

### Importieren eines Workflows

Sie können Workflows mithilfe der Orchestrator-REST-API importieren.

Abhängig von der Bibliothek Ihrer REST-Clientanwendung können Sie benutzerdefinierten Code verwenden, der die Eigenschaften der Workflows definiert.

#### Voraussetzungen

Der binäre Inhalt des Workflows muss als mehrteiliger Inhalt verfügbar sein. Weitere Informationen finden Sie in RFC 2387.

#### Vorgehensweise

- 1 Fügen Sie in der REST-Clientanwendung Anforderungsheader hinzu, um die Eigenschaften des zu importierenden Workflows festzulegen.
- 2 Führen Sie eine POST-Anforderung unter der URL für die Workflowobjekte aus:

```
POST http://{Orchestrator_Host}:{Port}/vco/api/workflows/
```

War die POST-Anforderung erfolgreich, wird der Statuscode 202 ausgegeben.

### Exportieren eines Workflows

Sie können Workflows mithilfe der Orchestrator-REST-API exportieren und als Datei herunterladen.

#### Vorgehensweise

- 1 Fügen Sie in der REST-Clientanwendung einen Anforderungsheader mit folgenden Werten hinzu:
  - **Name:** `accept`
  - **Wert:** `application/zip`
- 2 Führen Sie eine GET-Anforderung unter der URL des Workflows aus, den Sie exportieren möchten:

```
GET http://{Orchestrator_Host}:{Port}/vco/api/workflows/{workflowID}/
```

War die GET-Anforderung erfolgreich, wird der Statuscode 200 ausgegeben. Der binäre Inhalt des Workflows ist als Anhang mit dem Standarddateinamen *Workflow\_Name.workflow* verfügbar. Sie können die Datei mit einer REST-Clientanwendung speichern.

## Importieren einer Aktion

Sie können Aktionen mithilfe der Orchestrator-REST-API importieren.

Abhängig von der Bibliothek Ihrer REST-Clientanwendung können Sie benutzerdefinierten Code verwenden, der die Eigenschaften der Aktion definiert.

### Voraussetzungen

Der binäre Inhalt der Aktion muss als mehrteiliger Inhalt verfügbar sein. Weitere Informationen finden Sie in RFC 2387.

### Vorgehensweise

- 1 Fügen Sie in der REST-Clientanwendung Anforderungsheader hinzu, um die Eigenschaften der zu importierenden Aktion festzulegen.
- 2 Führen Sie eine POST-Anforderung unter der URL der Aktionsobjekte aus:

```
POST http://{Orchestrator_Host}:{Port}/vco/api/actions/
```

War die POST-Anforderung erfolgreich, wird der Statuscode 202 ausgegeben.

## Exportieren einer Aktion

Sie können Aktionen mithilfe der Orchestrator-REST-API exportieren und als Datei herunterladen.

### Vorgehensweise

- 1 Fügen Sie in der REST-Clientanwendung einen Anforderungsheader mit folgenden Werten hinzu:
  - **Name:** `accept`
  - **Wert:** `application/zip`
- 2 Führen Sie eine GET-Anforderung unter der URL der Aktion aus, die Sie exportieren möchten:

```
GET http://{Orchestrator_Host}:{Port}/vco/api/actions/{actionID}/
```

War die GET-Anforderung erfolgreich, wird der Statuscode 200 ausgegeben. Der binäre Inhalt der Aktion ist als Anhang mit dem Standarddateinamen *Action\_Name.action* verfügbar. Sie können die Datei mit einer REST-Clientanwendung speichern.

## Importieren eines Pakets

Sie können Pakete mithilfe der Orchestrator-REST-API importieren.



Abhängig von der Bibliothek Ihrer REST-Clientanwendung können Sie benutzerdefinierten Code verwenden, der die Eigenschaften des Pakets definiert.

Wenn Sie ein Orchestrator-Paket mit einem doppelten Namen importieren, wird das vorhandene Paket standardmäßig nicht überschrieben. Mithilfe eines Parameters in der Anforderung können Sie festlegen, ob vorhandene Pakete überschrieben werden sollen.

Standardmäßig werden Orchestrator-Pakete mit den Attributwerten der Konfigurationselemente importiert. Pakete ohne Attributwerte können mithilfe der Parameter in der Anforderung importiert werden.

In Orchestrator-Paketen enthaltene Tags werden standardmäßig importiert. Wenn dieselben Tags auf dem Orchestrator-Server bereits vorhanden sind, werden die Werte der vorhandenen Tags beibehalten. Mithilfe von Parametern in der Anforderung können Sie festlegen, ob vorhandene Tagwerte beibehalten werden sollen.

### Voraussetzungen

Der binäre Inhalt der Pakete muss als mehrteiliger Inhalt verfügbar sein. Weitere Informationen finden Sie in RFC 2387.

### Vorgehensweise

- 1 Fügen Sie in der REST-Clientanwendung Anforderungsheader hinzu, um die Eigenschaften des zu importierenden Pakets festzulegen.
- 2 Führen Sie eine POST-Anforderung unter der URL für die Paketobjekte aus:

```
POST http://{Orchestrator_Host}:{Port}/vco/api/packages/
```

- 3 (Optional) Um ein Paket zu importieren und ein vorhandenes Paket mit demselben Namen zu überschreiben, verwenden Sie den Parameter `overwrite` in der POST-Anforderung:

```
POST http://{Orchestrator_Host}:{Port}/vco/api/packages/?overwrite=true
```

- 4 (Optional) Durch Angabe des Parameters `importConfigurationAttributeValues` in der POST-Anforderung können Sie ein Paket ohne die Attributwerte des Konfigurationselements importieren:

```
POST http://{Orchestrator_Host}:{Port}/vco/api/packages/?importConfigurationAttributeValues=false
```

- 5 (Optional) Durch Angabe des Parameters `tagImportMode` in der POST-Anforderung können Sie ein Paket ohne die darin enthaltenen Tags importieren:

```
POST http://{Orchestrator_Host}:{Port}/vco/api/packages/?tagImportMode=DoNotImport
```

- 6 (Optional) Durch Angabe des Parameters `tagImportMode` in der POST-Anforderung können Sie ein Paket ohne die darin enthaltenen Tags importieren und vorhandene Tagwerte überschreiben:

```
POST http://{Orchestrator_Host}:{Port}/vco/api/packages/?tagImportMode=ImportAndOverwriteExisting-Value
```

War die POST-Anforderung erfolgreich, wird der Statuscode 202 ausgegeben.

## Exportieren von Paketen

Sie können Pakete mithilfe der Orchestrator-REST-API exportieren und als Datei herunterladen.

Standardmäßig werden Orchestrator-Pakete mit den Attributwerten der Konfigurationselemente und globalen Tags exportiert. Pakete ohne Attributwerte oder globale Tags können durch Angabe der Parameter in der Anforderung exportiert werden. Sie können der heruntergeladenen Datei auch einen benutzerdefinierten Namen geben.

### Vorgehensweise

- 1 Fügen Sie in der REST-Clientanwendung einen Anforderungsheader mit folgenden Werten hinzu:

- **Name:** `accept`
- **Wert:** `application/zip`

- 2 Führen Sie eine GET-Anforderung unter der URL des Pakets aus, das Sie exportieren möchten:

```
GET http://{Orchestrator_Host}:{Port}/vco/api/packages/{Paket_Name}/
```

- 3 (Optional) Durch Angabe des `packageName`-Parameters in der GET-Anforderung können Sie dem exportierten Paket einen benutzerdefinierten Namen geben:

```
GET http://{Orchestrator_Host}:{Port}/vco/api/packages/{Paket_Name}?packageName={benutzerdefinierter_Name}
```

- 4 (Optional) Durch Angabe des Parameters `exportConfigurationAttributeValues` in der GET-Anforderung können Sie ein Paket ohne die Attributwerte des Konfigurationselements exportieren:

```
GET http://{Orchestrator_Host}:{Port}/vco/api/packages/{Paket_Name}?exportConfigurationAttributeValues=false
```

- 5 (Optional) Durch Angabe des Parameters `exportGlobalTags` in der GET-Anforderung können Sie ein Paket ohne globale Tags exportieren:

```
GET http://{Orchestrator_Host}:{Port}/vco/api/packages/{Paket_Name}?exportGlobalTags=false
```

War die GET-Anforderung erfolgreich, wird der Statuscode 200 ausgegeben. Der binäre Inhalt des Pakets ist als Anhang mit dem Standarddateinamen `Paket_Name.package` verfügbar. Sie können die Datei mit einer REST-Clientanwendung speichern.

## Importieren einer Ressource

Sie können Ressourcen mithilfe der Orchestrator-REST-API importieren.

Abhängig von der Bibliothek Ihrer REST-Clientanwendung können Sie benutzerdefinierten Code verwenden, der die Eigenschaften der Ressourcen definiert.

## Voraussetzungen

Der binäre Inhalt der Ressource muss als mehrteiliger Inhalt verfügbar sein. Weitere Informationen finden Sie in RFC 2387.

## Vorgehensweise

- 1 Fügen Sie in der REST-Clientanwendung Anforderungsheader hinzu, um die Eigenschaften der Ressource festzulegen, das Sie importieren möchten.
- 2 Führen Sie eine POST-Anforderung unter der URL für die Ressourcenobjekte aus:

```
POST http://{Orchestrator_Host}:{Port}/vco/api/resources/
```

War die POST-Anforderung erfolgreich, wird der Statuscode 202 ausgegeben.

## Exportieren einer Ressource

Sie können Ressourcen mithilfe der Orchestrator-REST-API exportieren.

## Vorgehensweise

- 1 Fügen Sie in der REST-Clientanwendung einen Anforderungsheader mit folgenden Werten hinzu:
  - **Name:** `accept`
  - **Wert:** `application/octet-stream`
- 2 Führen Sie eine GET-Anforderung unter der URL der Ressource aus, die Sie exportieren möchten:

```
GET http://{Orchestrator_Host}:{Port}/vco/api/resources/{resourceID}/
```

War die GET-Anforderung erfolgreich, wird der Statuscode 200 ausgegeben. Der Inhalt der Ressource wird im Hauptteil der Antwort angezeigt.

## Importieren von Konfigurationselementen

Sie können Konfigurationselemente mithilfe der Orchestrator-REST-API importieren.

Abhängig von der Bibliothek Ihrer REST-Clientanwendung können Sie benutzerdefinierten Code verwenden, der die Eigenschaften des Konfigurationselements definiert.

## Voraussetzungen

Der binäre Inhalt des Konfigurationselements muss als mehrteiliger Inhalt verfügbar sein. Weitere Informationen finden Sie in RFC 2387.

## Vorgehensweise

- 1 Fügen Sie in der REST-Clientanwendung Anforderungsheader hinzu, um die Eigenschaften des zu importierenden Konfigurationselements festzulegen.

- 2 Führen Sie eine POST-Anforderung unter der URL der Konfigurationselementobjekte aus:

```
POST http://{Orchestrator_Host}:{Port}/vco/api/configurations/
```

War die POST-Anforderung erfolgreich, wird der Statuscode 202 ausgegeben.

## Exportieren von Konfigurationselementen

Sie können Konfigurationselemente mithilfe der Orchestrator-REST-API exportieren.

### Vorgehensweise

- 1 Fügen Sie in der REST-Clientanwendung einen Anforderungsheader mit folgenden Werten hinzu:
  - **Name:** `accept`
  - **Wert:** `application/vcoobject+xml`
- 2 Führen Sie eine GET-Anforderung unter der URL des Konfigurationselements aus, das Sie exportieren möchten:

```
GET http://{Orchestrator_Host}:{Port}/vco/api/configurations/{configuration_elementID}/
```

War die GET-Anforderung erfolgreich, wird der Statuscode 200 ausgegeben. Der Inhalt des Konfigurationselements wird im Hauptteil der Antwort angezeigt.

## Löschen von Orchestrator-Objekten

Die Orchestrator-REST-API stellt Webdienste bereit, mit denen Sie Workflows, Aktionen, Pakete, Ressourcen und Konfigurationselemente löschen können.

### Löschen eines Workflows

Sie können Workflows mithilfe der Orchestrator-REST-API löschen.

### Vorgehensweise

- 1 Führen Sie eine GET-Anforderung durch und rufen Sie die ID des Workflow aus der Liste der zurückgegebenen Workflows ab:

```
GET http://{orchestrator_host}:{port}/vco/api/workflows/
```

- 2 Führen Sie eine DELETE-Anforderung unter der URL für den Workflows aus:

```
DELETE http://{Orchestrator_Host}:{Port}/vco/api/workflows/{workflowID}/
```

War die DELETE-Anforderung erfolgreich, wird der Statuscode 200 ausgegeben, und der Hauptteil der Antwort ist leer.

## Löschen einer Aktion

Sie können Aktionen mithilfe der Orchestrator-REST-API löschen.

### Vorgehensweise

- 1 Führen Sie eine GET-Anforderung durch und rufen Sie die ID der Aktion aus der Liste der zurückgegebenen Aktionen ab:

```
GET http://{orchestrator_host}:{port}/vco/api/actions/
```

- 2 Führen Sie eine DELETE-Anforderung unter der URL für die Aktion aus:

```
DELETE http://{Orchestrator_Host}:{Port}/vco/api/actions/{actionID}/
```

War die DELETE-Anforderung erfolgreich, wird der Statuscode 200 ausgegeben, und der Hauptteil der Antwort ist leer.

## Löschen eines Pakets

Sie können Pakete mithilfe der Orchestrator-REST-API löschen.

Wenn Sie ein Paket löschen, werden die Elemente des Pakets nicht gelöscht. Wenn Sie den Inhalt eines Pakets löschen möchten, müssen Sie einen Optionsparameter angeben.

### Vorgehensweise

- 1 Führen Sie eine GET-Anforderung durch und rufen Sie den Namen des Pakets aus der Liste der zurückgegebenen Pakete ab:

```
GET http://{orchestrator_host}:{port}/vco/api/packages/
```

- 2 Führen Sie eine DELETE-Anforderung unter der URL des Pakets aus. Wenn Sie Elemente aus diesem Paket löschen möchten, geben Sie einen Optionsparameter am Ende der Anforderung an:

```
DELETE http://{Orchestrator_Host}:{Port}/vco/api/packages/{Paket_Name}/?option={Parameter}
```

Parameter	Beschreibung
<b>deletePackage</b>	Nur das Paket wird gelöscht, der Inhalt bleibt erhalten.
<b>deletePackageWithContent</b>	Das Paket und sein gesamter Inhalt werden gelöscht. Wenn andere Pakete Elemente des gelöschten Pakets ebenfalls nutzen, werden die geteilten Elemente aus den anderen Paketen gelöscht.
<b>deletePackageKeepingShared</b>	Das Paket und nicht gemeinsam genutzte Inhalte werden gelöscht. Mit anderen Paketen gemeinsam genutzte Elemente werden nicht gelöscht.

Wenn Sie keinen Optionsparameter angeben, wird der standardmäßige Parameter `deletePackage` verwendet.

War die DELETE-Anforderung erfolgreich, wird der Statuscode 200 ausgegeben, und der Hauptteil der Antwort ist leer.

## Löschen einer Ressource

Sie können Ressourcen mithilfe der Orchestrator-REST-API löschen.

### Vorgehensweise

- 1 Führen Sie eine GET-Anforderung durch und rufen Sie die ID der Ressource aus der Liste der zurückgegebenen Ressourcen ab:

```
GET http://{orchestrator_host}:{port}/vco/api/resources/
```

- 2 Führen Sie eine DELETE-Anforderung unter der URL für die Ressource aus:

```
DELETE http://{Orchestrator_Host}:{Port}/vco/api/resources/{resourceID}/
```

War die DELETE-Anforderung erfolgreich, wird der Statuscode 200 ausgegeben, und der Hauptteil der Antwort ist leer.

## Löschen eines Konfigurationselements

Sie können Konfigurationselemente mithilfe der Orchestrator-REST-API löschen.

### Vorgehensweise

- 1 Führen Sie eine GET-Anforderung durch und rufen Sie die ID des Konfigurationselements aus der Liste der zurückgegebenen Konfigurationselemente ab:

```
GET http://{orchestrator_host}:{port}/vco/api/configurations/
```

- 2 Führen Sie eine DELETE-Anforderung unter der URL des Konfigurationselementobjekts aus:

```
DELETE http://{Orchestrator_Host}:{Port}/vco/api/configurations/{configuration_elementID}/
```

War die DELETE-Anforderung erfolgreich, wird der Statuscode 200 ausgegeben, und der Hauptteil der Antwort ist leer.

## Festlegen von Berechtigungen für Orchestrator-Objekte

Berechtigungen für Orchestrator-Objekte können mithilfe der REST-API festgelegt werden. Zum Festlegen der Berechtigungen müssen Sie eine POST-Anforderung unter der URL der Objektberechtigungen ausführen und die Berechtigungen im Hauptteil der Anforderung definieren.

Sie können die Orchestrator-REST-API auch zum Abrufen von Informationen über die Berechtigungen eines Objekts oder zum Löschen der bestehenden Berechtigungen verwenden.

## REST-API-Berechtigungen

Beim Festlegen von Berechtigungen mithilfe der Orchestrator-REST-API müssen Sie einen Satz von Zeichen verwenden, um die Berechtigungen zu definieren.

Sie können die Berechtigungen für ein Element definieren, indem Sie eine Sequenz von Zeichen in das `<rights>`-Tag im Hauptteil einer POST-Anforderung einbeziehen.

Die Zeichen, die Sie zum Festlegen von Berechtigungen über die Orchestrator-REST-API verwenden können, haben bestimmte Bedeutungen.

**Tabelle 2-1. Zeichensatz der Orchestrator-REST-API für Berechtigungen**

Zeichen	Beschreibung
r	Erteilt Anzeigeberechtigungen.
x	Erteilt Ausführungsberechtigungen.
i	Erteilt Überprüfungsberechtigungen.
c	Erteilt Bearbeitungsberechtigungen.
a	Erteilt Administratorberechtigungen.

### Beispiel: Syntax zum Festlegen von Berechtigungen

Im Hauptteil einer POST-Anforderung unter der URL für die Berechtigungen eines Orchestrator-Elements können Sie die folgende Beispielsyntax verwenden.

```
<permissions xmlns="http://www.vmware.com/vco">
  <permission>
    <principal>cn=vcousers,ou=vco,dc=appliance</principal>
    <rights>ric</rights>
  </permission>
</permissions>
```

Indem Sie `ric`-Berechtigungen im Tag `<rights>` im Hauptteil der Anforderung festlegen, erteilen Sie Mitgliedern der Benutzergruppe `vcousers` die Berechtigung, das Orchestrator-Element anzuzeigen, zu überprüfen und zu bearbeiten.

## Abrufen der Berechtigungen eines Workflows

Mithilfe der Orchestrator-REST-API können Sie Informationen zu den Berechtigungen eines Workflows abrufen.

### Vorgehensweise

- 1 Führen Sie eine GET-Anforderung durch und rufen Sie die ID des Workflow aus der Liste der zurückgegebenen Workflows ab:

```
GET http://{orchestrator_host}:{port}/vco/api/workflows/
```

- 2 Führen Sie eine GET-Anforderung unter der URL für die Berechtigungen des Workflows aus:

```
GET http://{Orchestrator_Host}:{Port}/vco/api/workflows/{workflowID}/permissions/
```

War die GET-Anforderung erfolgreich, wird der Statuscode 200 ausgegeben. Informationen zu den Berechtigungen des Workflows werden im Hauptteil der Antwort angezeigt.

## Löschen der Berechtigungen eines Workflows

Sie können die Berechtigungen eines Workflows mithilfe der Orchestrator-REST-API löschen. Sie können die vorhandenen Berechtigungen eines Workflows löschen, bevor Sie neue Berechtigungen festlegen.

### Vorgehensweise

- 1 Führen Sie eine GET-Anforderung durch und rufen Sie die ID des Workflow aus der Liste der zurückgegebenen Workflows ab:

```
GET http://{orchestrator_host}:{port}/vco/api/workflows/
```

- 2 Führen Sie eine DELETE-Anforderung unter der URL für die Berechtigungen des Workflows aus:

```
DELETE http://{Orchestrator_Host}:{Port}/vco/api/workflows/{workflowID}/permissions/
```

War die DELETE-Anforderung erfolgreich, wird der Statuscode 204 ausgegeben, und der Hauptteil der Antwort ist leer.

## Festlegen der Berechtigungen für einen Workflow

Sie können die Berechtigungen für Workflows mithilfe der Orchestrator-REST-API bearbeiten.

### Voraussetzungen

Überprüfen Sie die Berechtigungen und die Syntax, die im Hauptteil einer Anforderung verwenden können. Weitere Informationen finden Sie unter [REST-API-Berechtigungen](#).

### Vorgehensweise

- 1 Führen Sie eine GET-Anforderung durch und rufen Sie die ID des Workflow aus der Liste der zurückgegebenen Workflows ab:

```
GET http://{orchestrator_host}:{port}/vco/api/workflows/
```

- 2 Fügen Sie in einer REST-Clientanwendung Anforderungsheader zum Festlegen der Eigenschaften des Workflows, dessen Berechtigungen Sie einstellen möchten, ein.
- 3 Legen Sie im Hauptteil der Anforderung die gewünschten Berechtigungen fest.



- 4 Führen Sie eine POST-Anforderung unter der URL für die Berechtigungen des Workflows aus:

```
POST http://{Orchestrator_Host}:{Port}/vco/api/workflows/{workflowID}/permissions/
```

War die POST-Anforderung erfolgreich, wird der Statuscode 201 ausgegeben. Informationen zu den Berechtigungen des Workflows werden im Hauptteil der Antwort angezeigt.

## Abrufen der Berechtigungen einer Aktion

Mithilfe der Orchestrator-REST-API können Sie Informationen zu den Berechtigungen einer Aktion abrufen.

### Vorgehensweise

- 1 Führen Sie eine GET-Anforderung durch und rufen Sie die ID der Aktion aus der Liste der zurückgegebenen Aktionen ab:

```
GET http://{orchestrator_host}:{port}/vco/api/actions/
```

- 2 Führen Sie eine GET-Anforderung unter der URL für die Berechtigungen der Aktion aus:

```
GET http://{Orchestrator_Host}:{Port}/vco/api/actions/{actionID}/permissions/
```

War die GET-Anforderung erfolgreich, wird der Statuscode 200 ausgegeben. Informationen zu den Berechtigungen der Aktion werden im Hauptteil der Antwort angezeigt.

## Löschen der Berechtigungen einer Aktion

Sie können die Berechtigungen einer Aktion mithilfe der Orchestrator-REST-API löschen. Sie können die vorhandenen Berechtigungen einer Aktion löschen, bevor Sie neue Berechtigungen festlegen.

### Vorgehensweise

- 1 Führen Sie eine GET-Anforderung durch und rufen Sie die ID der Aktion aus der Liste der zurückgegebenen Aktionen ab:

```
GET http://{orchestrator_host}:{port}/vco/api/actions/
```

- 2 Führen Sie eine DELETE-Anforderung unter der URL für die Berechtigungen der Aktion aus:

```
DELETE http://{Orchestrator_Host}:{Port}/vco/api/actions/{actionID}/permissions/
```

War die DELETE-Anforderung erfolgreich, wird der Statuscode 204 ausgegeben, und der Hauptteil der Antwort ist leer.

## Festlegen der Berechtigungen für eine Aktion

Sie können die Berechtigungen für Aktionen mithilfe der Orchestrator-REST-API bearbeiten.

## Voraussetzungen

Überprüfen Sie die Berechtigungen und die Syntax, die im Hauptteil einer Anforderung verwenden können. Weitere Informationen finden Sie unter [REST-API-Berechtigungen](#).

## Vorgehensweise

- 1 Führen Sie eine GET-Anforderung durch und rufen Sie die ID der Aktion aus der Liste der zurückgegebenen Aktionen ab:

```
GET http://{orchestrator_host}:{port}/vco/api/actions/
```

- 2 Fügen Sie in einer REST-Clientanwendung Anforderungsheader zum Festlegen der Eigenschaften der Aktion, deren Berechtigungen Sie einstellen möchten, ein.
- 3 Legen Sie im Hauptteil der Anforderung die gewünschten Berechtigungen fest.
- 4 Führen Sie eine POST-Anforderung unter der URL für die Berechtigungen der Aktion aus:

```
POST http://{Orchestrator_Host}:{Port}/vco/api/actions/{actionID}/permissions/
```

War die POST-Anforderung erfolgreich, wird der Statuscode 201 ausgegeben. Informationen zu den Berechtigungen der Aktion werden im Hauptteil der Antwort angezeigt.

## Abrufen der Berechtigungen eines Pakets

Mithilfe der Orchestrator-REST-API können Sie Informationen zu den Berechtigungen eines Pakets abrufen.

## Vorgehensweise

- 1 Führen Sie eine GET-Anforderung durch und rufen Sie den Namen des Pakets aus der Liste der zurückgegebenen Pakete ab:

```
GET http://{orchestrator_host}:{port}/vco/api/packages/
```

- 2 Führen Sie eine GET-Anforderung unter der URL für die Berechtigungen des Pakets aus:

```
GET http://{Orchestrator_Host}:{Port}/vco/api/packages/{Paket_Name}/permissions/
```

War die GET-Anforderung erfolgreich, wird der Statuscode 200 ausgegeben. Informationen zu den Berechtigungen des Pakets werden im Hauptteil der Antwort angezeigt.

## Löschen der Berechtigungen eines Pakets

Sie können die Berechtigungen eines Pakets mithilfe der Orchestrator-REST-API löschen. Sie können die vorhandenen Berechtigungen eines Pakets löschen, bevor Sie neue Berechtigungen festlegen.

### Vorgehensweise

- 1 Führen Sie eine GET-Anforderung durch und rufen Sie den Namen des Pakets aus der Liste der zurückgegebenen Pakete ab:

```
GET http://{orchestrator_host}:{port}/vco/api/packages/
```

- 2 Führen Sie eine DELETE-Anforderung unter der URL für die Berechtigungen des Pakets aus:

```
DELETE http://{Orchestrator_Host}:{Port}/vco/api/packages/{Paket_Name}/permissions/
```

War die DELETE-Anforderung erfolgreich, wird der Statuscode 204 ausgegeben, und der Hauptteil der Antwort ist leer.

## Festlegen der Berechtigungen für ein Paket

Sie können die Berechtigungen für Pakete mithilfe der Orchestrator-REST-API bearbeiten.

### Voraussetzungen

Überprüfen Sie die Berechtigungen und die Syntax, die im Hauptteil einer Anforderung verwenden können. Weitere Informationen finden Sie unter [REST-API-Berechtigungen](#).

### Vorgehensweise

- 1 Führen Sie eine GET-Anforderung durch und rufen Sie den Namen des Pakets aus der Liste der zurückgegebenen Pakete ab:

```
GET http://{orchestrator_host}:{port}/vco/api/packages/
```

- 2 Fügen Sie in einer REST-Clientanwendung Anforderungsheader zum Festlegen der Eigenschaften des Pakets, dessen Berechtigungen Sie einstellen möchten, ein.
- 3 Legen Sie im Hauptteil der Anforderung die gewünschten Berechtigungen fest.
- 4 Führen Sie eine POST-Anforderung unter der URL für die Berechtigungen des Pakets aus:

```
POST http://{Orchestrator_Host}:{Port}/vco/api/packages/{Paket_Name}/permissions/
```

War die POST-Anforderung erfolgreich, wird der Statuscode 201 ausgegeben. Informationen zu den Berechtigungen des Pakets werden im Hauptteil der Antwort angezeigt.

## Abrufen der Berechtigungen einer Ressource

Mithilfe der Orchestrator-REST-API können Sie Informationen zu den Berechtigungen einer Ressource abrufen.

### Vorgehensweise

- 1 Führen Sie eine GET-Anforderung durch und rufen Sie die ID der Ressource aus der Liste der zurückgegebenen Ressourcen ab:

```
GET http://{orchestrator_host}:{port}/vco/api/resources/
```

- 2 Führen Sie eine GET-Anforderung unter der URL für die Berechtigungen der Ressource aus:

```
GET http://{Orchestrator_Host}:{Port}/vco/api/resources/{resourceID}/permissions/
```

War die GET-Anforderung erfolgreich, wird der Statuscode 200 ausgegeben. Informationen zu den Berechtigungen der Ressource werden im Hauptteil der Antwort angezeigt.

## Löschen der Berechtigungen einer Ressource

Sie können die Berechtigungen einer Ressource mithilfe der Orchestrator-REST-API löschen. Sie können die vorhandenen Berechtigungen einer Ressourcen löschen, bevor Sie neue Berechtigungen festlegen.

### Vorgehensweise

- 1 Führen Sie eine GET-Anforderung durch und rufen Sie die ID der Ressource aus der Liste der zurückgegebenen Ressourcen ab:

```
GET http://{orchestrator_host}:{port}/vco/api/resources/
```

- 2 Führen Sie eine DELETE-Anforderung unter der URL für die Berechtigungen der Ressource aus:

```
DELETE http://{Orchestrator_Host}:{Port}/vco/api/resources/{resourceID}/permissions/
```

War die DELETE-Anforderung erfolgreich, wird der Statuscode 204 ausgegeben, und der Hauptteil der Antwort ist leer.

## Festlegen der Berechtigungen für eine Ressource

Sie können die Berechtigungen für Ressourcen mithilfe der Orchestrator-REST-API bearbeiten.

### Voraussetzungen

Überprüfen Sie die Berechtigungen und die Syntax, die im Hauptteil einer Anforderung verwenden können. Weitere Informationen finden Sie unter [REST-API-Berechtigungen](#).

### Vorgehensweise

- 1 Führen Sie eine GET-Anforderung durch und rufen Sie die ID der Ressource aus der Liste der zurückgegebenen Ressourcen ab:

```
GET http://{orchestrator_host}:{port}/vco/api/resources/
```

- 2 Fügen Sie in einer REST-Clientanwendung Anforderungsheader zum Festlegen der Eigenschaften der Ressource, deren Berechtigungen Sie einstellen möchten, ein.
- 3 Legen Sie im Hauptteil der Anforderung die gewünschten Berechtigungen fest.
- 4 Führen Sie eine POST-Anforderung unter der URL für die Berechtigungen der Ressource aus:

```
POST http://{Orchestrator_Host}:{Port}/vco/api/resources/{resourceID}/permissions/
```

War die POST-Anforderung erfolgreich, wird der Statuscode 201 ausgegeben. Informationen zu den Berechtigungen der Ressource werden im Hauptteil der Antwort angezeigt.

## Abrufen der Berechtigungen eines Konfigurationselements

Mithilfe der Orchestrator-REST-API können Sie Informationen zu den Berechtigungen eines Konfigurationselements abrufen.

### Vorgehensweise

- 1 Führen Sie eine GET-Anforderung durch und rufen Sie die ID des Konfigurationselements aus der Liste der zurückgegebenen Konfigurationselemente ab:

```
GET http://{orchestrator_host}:{port}/vco/api/configurations/
```

- 2 Führen Sie eine GET-Anforderung unter der URL für die Berechtigungen des Konfigurationselements aus:

```
GET http://{Orchestrator_Host}:{Port}/vco/api/configurations/{configuration_elementID}/permissions/
```

War die GET-Anforderung erfolgreich, wird der Statuscode 200 ausgegeben. Informationen zu den Berechtigungen des Konfigurationselements werden im Hauptteil der Antwort angezeigt.

## Löschen der Berechtigungen eines Konfigurationselements

Sie können die Berechtigungen eines Konfigurationselements mithilfe der Orchestrator-REST-API löschen. Sie können die vorhandenen Berechtigungen eines Konfigurationselements löschen, bevor Sie neue Berechtigungen festlegen.

### Vorgehensweise

- 1 Führen Sie eine GET-Anforderung durch und rufen Sie die ID des Konfigurationselements aus der Liste der zurückgegebenen Konfigurationselemente ab:

```
GET http://{orchestrator_host}:{port}/vco/api/configurations/
```

- 2 Führen Sie eine DELETE-Anforderung unter der URL für die Berechtigungen des Konfigurationselements aus:

```
DELETE http://{Orchestrator_Host}:{Port}/vco/api/configurations/{configuration_elementID}/permissions/
```

War die DELETE-Anforderung erfolgreich, wird der Statuscode 204 ausgegeben, und der Hauptteil der Antwort ist leer.

## Festlegen der Berechtigungen für ein Konfigurationselement

Sie können die Berechtigungen für Konfigurationselemente mithilfe der Orchestrator-REST-API bearbeiten.

### Voraussetzungen

Überprüfen Sie die Berechtigungen und die Syntax, die im Hauptteil einer Anforderung verwenden können. Weitere Informationen finden Sie unter [REST-API-Berechtigungen](#).

### Vorgehensweise

- 1 Führen Sie eine GET-Anforderung durch und rufen Sie die ID des Konfigurationselements aus der Liste der zurückgegebenen Konfigurationselemente ab:

```
GET http://{orchestrator_host}:{port}/vco/api/configurations/
```

- 2 Fügen Sie in einer REST-Clientanwendung Anforderungsheader zum Festlegen der Eigenschaften des Konfigurationselements, dessen Berechtigungen Sie einstellen möchten, ein.
- 3 Legen Sie im Hauptteil der Anforderung die gewünschten Berechtigungen fest.
- 4 Führen Sie eine POST-Anforderung unter der URL für die Berechtigungen des Konfigurationselements aus:

```
POST http://{Orchestrator_Host}:{Port}/vco/api/configurations/{configuration_elementID}/permissions/
```

War die POST-Anforderung erfolgreich, wird der Statuscode 201 ausgegeben. Informationen zu den Berechtigungen des Konfigurationselements werden im Hauptteil der Antwort angezeigt.

## Durchführen von Vorgängen mit Plug-Ins

Die Orchestrator-REST-API stellt Webdienste bereit, mit denen Sie verschiedene Vorgänge mit Plug-Ins ausführen können.

### Abrufen von Informationen zu Plug-Ins

Mithilfe der Orchestrator-REST-API können Sie Metadateninformationen für alle installierten Plug-Ins abrufen.

### Vorgehensweise

- 1 Fügen Sie in der REST-Clientanwendung Anforderungsheader hinzu, um die Eigenschaften des Plug-Ins zu definieren.
- 2 Führen Sie eine GET-Anforderung unter der URL der Plug-In-Objekte aus:

```
GET http://{Orchestrator_Host}:{Port}/vco/api/plugins/
```

War die GET-Anforderung erfolgreich, wird der Statuscode 200 ausgegeben.

## Importieren von Plug-Ins

Sie können Plug-Ins mithilfe der Orchestrator-REST-API importieren.

Abhängig von der Bibliothek Ihrer REST-Clientanwendung können Sie benutzerdefinierten Code verwenden, der die Eigenschaften des Plug-Ins definiert.

---

**Hinweis** Sie können ein Plug-In nicht importieren, wenn ein Plug-In mit demselben Namen bereits installiert ist.

---

### Voraussetzungen

Der binäre Inhalt des Plug-Ins muss als mehrteiliger Inhalt verfügbar sein. Weitere Informationen finden Sie in RFC 2387.

### Vorgehensweise

- 1 Fügen Sie in der REST-Clientanwendung Anforderungsheader hinzu, um die Eigenschaften des Plug-Ins festzulegen, das Sie importieren möchten.
- 2 Führen Sie eine POST-Anforderung unter der URL der Plug-In-Objekte aus:

```
POST http://{Orchestrator_Host}:{Port}/vco/api/plugins/
```

War die POST-Anforderung erfolgreich, wird der Statuscode 200 ausgegeben.

## Exportieren eines Plug-Ins

Sie können Plug-Ins mithilfe der Orchestrator-REST-API exportieren.

### Vorgehensweise

- 1 Fügen Sie in der REST-Clientanwendung einen Anforderungsheader mit folgenden Werten hinzu:
  - **Name:** `accept`
  - **Wert:** `application/dar`

- 2 Führen Sie eine GET-Anforderung unter der URL des Plug-Ins aus, das Sie exportieren möchten:

```
GET http://{Orchestrator_Host}:{Port}/vco/api/plugins/{Plug-In_Name}/
```

War die GET-Anforderung erfolgreich, wird der Statuscode 200 ausgegeben. Der Inhalt des Plug-Ins wird im Hauptteil der Antwort angezeigt.

## Aktivieren oder Deaktivieren von Plug-Ins

Sie können Plug-Ins mithilfe der Orchestrator-REST-API aktivieren oder deaktivieren.

Führen Sie eine PUT-Anforderung unter der URL des Plug-Ins aus, um den Status des Plug-Ins von „Aktiviert“ auf „Deaktiviert“ zu setzen oder umgekehrt. Sie können den aktuellen Status eines Plug-Ins durch Abrufen der Informationen zu Orchestrator-Plug-Ins überprüfen. Weitere Informationen finden Sie unter [Abrufen von Informationen zu Plug-Ins](#).

### Voraussetzungen

Der binäre Inhalt des Plug-Ins muss als mehrteiliger Inhalt verfügbar sein. Weitere Informationen finden Sie in RFC 2387.

### Vorgehensweise

- 1 Fügen Sie in der REST-Clientanwendung Anforderungsheader hinzu, um die Eigenschaften des Plug-Ins festzulegen, das Sie aktivieren oder deaktivieren möchten.
- 2 Führen Sie eine PUT-Anforderung unter der URL des Plug-Ins aus, das Sie aktivieren oder deaktivieren möchten:

```
PUT http://{Orchestrator_Host}:{Port}/vco/api/plugins/{Plug-In_Name}/state/
```

War die PUT-Anforderung erfolgreich, wird der Statuscode 200 ausgegeben.

## Ausführen von Server-Konfigurationsvorgängen

Die Orchestrator-REST-API stellt Webdienste bereit, die Sie zum Ausführen verschiedener Vorgänge zur Orchestrator-Serverkonfiguration verwenden können.

## Abrufen von Informationen zur Orchestrator-Serverkonfiguration

Mithilfe der Orchestrator-REST-API können Sie Informationen zur Orchestrator-Serverkonfiguration abrufen.

### Vorgehensweise

- 1 Fügen Sie in der REST-Clientanwendung Anforderungsheader hinzu, um die Eigenschaften des Servers zu definieren, für den Sie Informationen abrufen möchten.



- 2 Führen Sie eine GET-Anforderung unter der URL der Plug-In-Objekte aus:

```
GET http://{Orchestrator_Host}:{Port}/vco/api/server-configuration/
```

War die GET-Anforderung erfolgreich, wird der Statuscode 200 ausgegeben.

## Importieren der Serverkonfiguration des Orchestrator-Servers

Sie können eine gespeicherte Konfiguration mithilfe der Orchestrator-REST-API importieren.

### Voraussetzungen

Der binäre Inhalt der Konfiguration muss als mehrteiliger Inhalt verfügbar sein. Weitere Informationen finden Sie in RFC 2387.

### Vorgehensweise

- 1 Fügen Sie in der REST-Clientanwendung einen Anforderungsheader mit folgenden Werten hinzu:
  - **Name:** `content-type`
  - **Wert:** `multipart/form-data`
- 2 Führen Sie eine POST-Anforderung unter der URL der Serverkonfiguration aus:

```
POST http://{Orchestrator_Host}:{Port}/vco/api/server-configuration/
```

War die POST-Anforderung erfolgreich, wird der Statuscode 200 ausgegeben.

## Exportieren der Serverkonfiguration eines Orchestrator-Servers

Sie können die Serverkonfiguration mithilfe der Orchestrator-REST-API exportieren.

### Voraussetzungen

Der binäre Inhalt der Konfiguration muss als mehrteiliger Inhalt verfügbar sein. Weitere Informationen finden Sie in RFC 2387.

### Vorgehensweise

- 1 Fügen Sie in der REST-Clientanwendung einen Anforderungsheader mit folgenden Werten hinzu:
  - **Name:** `content-type`
  - **Wert:** `multipart/form-data`
- 2 Fügen Sie einen weiteren Anforderungsheader mit folgenden Werten hinzu:
  - **Name:** `accept`
  - **Wert:** `*/*`

- 3 Führen Sie eine POST-Anforderung unter der URL der Serverkonfiguration aus:

```
POST http://{Orchestrator_Host}:{Port}/vco/api/server-configuration/
```

War die POST-Anforderung erfolgreich, wird der Statuscode 200 ausgegeben.

## Ausführen von Tagvorgängen

Die Orchestrator-REST-API stellt Webdienste bereit, mit denen Sie verschiedene Vorgänge für Objekte ausführen können, um ein besseres Auffinden in Orchestrator dank Tags zu ermöglichen.

Objekte werden durch das Hinzufügen von Tags besser auffindbar. Tags sind Zeichenfolgen mit einer Länge von 3 bis 64 Zeichen. Leerzeichen sind nicht zulässig.

Es können globale und private Tags hinzugefügt werden. Globale Tags sind für alle Orchestrator-Benutzer sichtbar. Private Tags sind nur für den jeweiligen Ersteller sichtbar. Globale Tags können nur von Benutzern mit Administratorrechten erstellt und entfernt werden.

## Markieren eines Objekts

Tags können Objekten mithilfe der Orchestrator-REST-API zugewiesen werden.

Sie können private und globale Tags erstellen. Ob ein Tag privat oder global ist, wird im Hauptteil der Anforderung festgelegt.

---

**Hinweis** Zum Erstellen globaler Tags müssen Sie als Benutzer mit Administratorrechten angemeldet sein.

---

Dem Tag kann auch ein Wert zugewiesen werden. Ein Tagwert ist ein optionaler Parameter, der zum Filtern von Tags verwendet werden kann.

### Vorgehensweise

- 1 Definieren Sie den Hauptteil der Anforderung mithilfe der folgenden Syntax:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<tag-instance xmlns="http://www.vmware.com/vco" global="false">
  <name>Tag_Name</name>
  <value>Tag_Wert</value>
</tag-instance>
```

---

**Hinweis** Ein globales Tag wird erstellt, wenn Sie die Variable **global** auf **"true"** einstellen.

---

- 2 Führen Sie eine POST-Anforderung unter der URL des Objekts aus:

```
POST http://{Orchestrator_Host}:{Port}/vco/api/catalog/{namespace}/{objectType}/{objectId}/tags
```

War die POST-Anforderung erfolgreich, wird der Statuscode 200 ausgegeben.

## Entfernen von Tags von einem Objekt

Einem Objekt zugewiesene Tags können mithilfe der Orchestrator-REST-API entfernt werden.

Sie können private und globale Tags entfernen.

---

**Hinweis** Zum Entfernen globaler Tags müssen Sie als Benutzer mit Administratorrechten angemeldet sein.

---

### Vorgehensweise

- ◆ Führen Sie zum Entfernen privater oder globaler Tags eine DELETE-Anforderung aus.
  - Zum Entfernen eines privaten Tags führen Sie eine DELETE-Anforderung unter der URL des Objekts aus, wobei Sie die folgende Syntax verwenden:

```
DELETE http://{Orchestrator_Host}:{Port}/vco/api/catalog/{namespace}/{objectType}/{objectId}/tag/{Tag_Name}
```

- Zum Entfernen eines globalen Tags führen Sie eine DELETE-Anforderung unter der URL des Objekts aus, wobei Sie die folgende Syntax verwenden:

```
DELETE http://{Orchestrator_Host}:{Port}/vco/api/catalog/{namespace}/{objectType}/{objectId}/tag/{Tag_Name}
```

War die DELETE-Anforderung erfolgreich, wird der Statuscode 200 ausgegeben.

## Auflisten von Objekttags

Eine dem Objekt zugewiesene Liste von Tags kann mithilfe der Orchestrator-REST-API abgerufen werden.

### Vorgehensweise

- ◆ Führen Sie eine GET-Anforderung unter der URL des Objekts aus:

```
GET http://{Orchestrator_Host}:{Port}/vco/api/catalog/{namespace}/{objectType}/{objectId}/tags
```

War die GET-Anforderung erfolgreich, wird der Statuscode 200 ausgegeben.

## Auflisten markierter Objekte nach Typ

Mithilfe der Orchestrator-REST-API können Sie eine Liste von Objekten abrufen, die mit einem speziellen Tag versehen sind, und sie nach Objekttyp filtern.

**Vorgehensweise**

- ◆ Führen Sie eine GET-Anforderung unter der URL des Objekttyps aus:

```
GET http://{Orchestrator_Host}:{Port}/vco/api/catalog/{namespace}/{objectType}?
tags=tag1&tags=:tag2=value
```

War die GET-Anforderung erfolgreich, wird der Statuscode 200 ausgegeben.

**Auflisten von Tagbesitzern**

Eine Liste der Tagbesitzer kann mithilfe der Orchestrator-REST-API abgerufen werden. Tagbesitzer sind Benutzer, die mindestens ein Tag erstellt haben.

**Vorgehensweise**

- ◆ Führen Sie eine GET-Anforderung unter der folgenden URL aus:

```
GET http://{Orchestrator_Host}:{Port}/vco/api/tags
```

War die GET-Anforderung erfolgreich, wird der Statuscode 200 ausgegeben. Die von Ihnen abgerufene Liste enthält Benutzer, die mindestens ein Tag erstellt haben. Globale Tags werden unter dem Systembenutzernamen `__GLOBAL__` aufgelistet.

**Auflisten von Tags nach Benutzer**

Mithilfe der Orchestrator-REST-API können Sie die von einem bestimmten Benutzer erstellte Liste von Tags entfernen.

Sie können auch globale Tags abrufen. Globale Tags werden unter dem Systembenutzernamen `__GLOBAL__` aufgelistet.

**Vorgehensweise**

- ◆ Führen Sie eine GET-Anforderung unter der URL des Benutzers aus.
  - Wenn Sie die von einem bestimmten Benutzer erstellte Liste von Tags abrufen möchten, führen Sie eine GET-Anforderung mit der folgenden Syntax aus:

```
GET http://{Orchestrator_Host}:{Port}/vco/api/tags/{Benutzer_Name}
```

- Wenn Sie eine Liste globaler Tags abrufen möchten, führen Sie eine GET-Anforderung mit der folgenden Syntax aus:

```
GET http://{Orchestrator_Host}:{Port}/vco/api/tags/__GLOBAL__
```

War die GET-Anforderung erfolgreich, wird der Statuscode 200 ausgegeben.

## Auflisten von Tags nach Benutzer, gefiltert nach Tagname

Mithilfe der Orchestrator-REST-API können Sie die von einem bestimmten Benutzer erstellte Liste von Taginstanzen abrufen und die Tags nach Tagname filtern.

Sie können auch globale Taginstanzen abrufen. Globale Tags werden unter dem Systembenutzernamen `__GLOBAL__` aufgelistet.

### Vorgehensweise

- ◆ Führen Sie eine GET-Anforderung unter der URL des Benutzers aus.
  - Wenn Sie die von einem bestimmten Benutzer erstellte gefilterte Liste von Taginstanzen abrufen möchten, führen Sie eine GET-Anforderung mit der folgenden Syntax aus:

```
GET http://{Orchestrator_Host}:{Port}/vco/api/tags/{Benutzer_Name}/{Tag_Name}
```

- Wenn Sie eine gefilterte Liste globaler Taginstanzen abrufen möchten, führen Sie eine GET-Anforderung mit der folgenden Syntax aus:

```
GET http://{Orchestrator_Host}:{Port}/vco/api/tags/__GLOBAL__/{Tag_Name}
```

War die GET-Anforderung erfolgreich, wird der Statuscode 200 ausgegeben. Die von Ihnen abgerufenen Informationen enthalten einen Verweis auf das mit einem Tag versehene Objekt, auf den Tagnamen, auf den Tagwert sowie einen Hinweis darauf, ob die Taginstanz global oder privat ist.

## Entfernen von Tags nach Benutzer

Mithilfe der Orchestrator-REST-API können Sie alle von einem bestimmten Benutzer erstellten Tags entfernen.

Außerdem können Sie globale Tags entfernen. Globale Tags werden unter dem Systembenutzernamen `__GLOBAL__` aufgelistet.

---

**Hinweis** Zum Entfernen globaler Tags müssen Sie als Benutzer mit Administratorrechten angemeldet sein.

---

## Vorgehensweise

- ◆ Führen Sie eine DELETE-Anforderung unter der URL des Benutzers aus.
  - Wenn Sie die von einem bestimmten Benutzer erstellten Tags entfernen möchten, führen Sie eine DELETE-Anforderung mit der folgenden Syntax aus:

```
DELETE http://{Orchestrator_Host}:{Port}/vco/api/tags/{Benutzer_Name}
```

- Wenn Sie die globalen Tags entfernen möchten, führen Sie eine DELETE-Anforderung mit der folgenden Syntax aus:

```
DELETE http://{Orchestrator_Host}:{Port}/vco/api/tags/__GLOBAL__
```

War die DELETE-Anforderung erfolgreich, wird der Statuscode 204 ausgegeben.