Overview

This chapter contains the following sections:

Getting Started with Cisco UCS Director REST API

The Cisco UCS Director REST API allows an application to interact with Cisco UCS Director, programmatically. These requests provide access to resources in Cisco UCS Director. With an API call, you can execute Cisco UCS Director workflows and change the configuration of switches, adapters, policies, and other hardware and software components.

For more information on how to setup the development environment, refer the Cisco UCS Director REST API Getting Started Guide.

Structure of an Example

Under a descriptive title, each example comprises the following sections:

Objective

What the example is designed to accomplish.

Context

When you would use the example, when you would not use it, and why.

Prerequisites

What conditions have to exist for the example to work.

REST URL

What is the REST URL to pass the REST API.

Components

Which objects and methods are used in the example, and what the input variables represent.

Code

The example code.

Results

What output is expected from the example code.

Implementation

Notes on implementing the example, including what modifications might be necessary to implement it.

See Also

Related Examples

How to Use the Examples

This document is a collection of examples-recipes, if you will-for using REST API, a server-side scripting solution for use with Cisco UCS Director Orchestrator. Like a cookbook, you can use this document in at least three ways:

- You can follow the examples as written (substituting your own variables, of course) to complete tasks without necessarily knowing everything about the steps you are following.
- You can use the examples as templates and adapt them to similar tasks in your work.
- You can study the examples to figure out "how things are done" in REST API and, along with the REST API Javadoc reference, generalize to using different methods for other tasks you need to script.

The examples are chosen to illustrate common use cases and are intended to facilitate all three of these modes of use.

Example: Self-Service Provisioning of Virtual Machines

This example shows how to use REST APIs to perform a straightforward task of enabling a user to self-service provision virtual machines (VMs).

The REST API calls involved in this use case are summarized, and the requests and the system responses are detailed. The responses are typical and will not exactly match your implementation, depending on the state of your Cisco UCS Director database. You need to extract required parameters from response and pass them through the API requests in your application.

You can provision virtual machines (VMs) using predefined catalog items. To accomplish this, you need to view a list of catalogs and choose an appropriate service container catalog. You can create a service container using the chosen catalog and submit a service request for self-provisioning the VMs on the service container.

To implement self-service VM provisioning, execute the following REST APIs in sequence:

1. UserAPIGetAllCatalogs—Retrieve a list of catalogs containing the cloud name and the group name to choose a catalog for provisioning a VM.
2. UserAPIServiceContainerCatalogRequest—Submit a service request to create a service container for provisioning VM, using the chosen catalog.
3. UserAPIGetServiceRequestWorkFlow—Optional. View the workflow details of the service request.
4. userAPISubmitServiceRequest—Submit the service request to provision a VM.

Step 1 Retrieve a list of catalogs containing the cloud name and the group name to which the VM is bound using the userAPIGetAllCatalogs API. You can then choose a catalog from the list that is returned.

Request

```
/app/api/rest?formatType=json&opName=userAPIGetAllCatalogs&opData={}
```

Response

```
{
"serviceResult":{"rows":[{"Catalog_ID":"5","Catalog_Name":"VNX_Pranita","Folder":"Advanced","Catalog_Type":"Advanced",
"Template_Name":"Not Applicable","Catalog_Description":"","Cloud":"","Image":"","Group":"Default
Group","Icon":
"/app/images/temp/1436492144835_ibm.png","OS":"","Additional_OS_Info":"","Applications":"","Additional_Application_Details":"",
"Status":"OK"},{"Catalog_ID":"6","Catalog_Name":"MSP_CAT","Folder":"Advanced","Catalog_Type":"Advanced","Template_Name":"Not
 Applicable",
"Catalog_Description":"","Cloud":"","Image":"","Group":"Default
Group","Icon":"/app/images/temp/1436492144835_ibm.png","OS":"","Additional_OS_Info":"","Applications":"",
"Additional_Application_Details":"","Status":"OK"},{"Catalog_ID":"7","Catalog_Name":"VNX_Update_Tenant","Folder":"Advanced",
"Catalog_Type":"Advanced","Template_Name":"Not
Applicable","Catalog_Description":"","Cloud":"","Image":"","Group":"Default Group",
"Icon":"/app/images/temp/1436492144835_ibm.png","OS":"","Additional_OS_Info":"","Applications":"","Additional_Application_Details":"",
"Status":"OK"},{"Catalog_ID":"8","Catalog_Name":"Pja_cat","Folder":"Advanced","Catalog_Type":"Advanced","Template_Name":"Not
 Applicable",
"Catalog_Description":"","Cloud":"","Image":"","Group":"Default
Group","Icon":"/app/images/temp/1436492144835_ibm.png",
"OS":"","Additional_OS_Info":"","Applications":"","Additional_Application_Details":"","Status":"OK"},
{"Catalog_ID":"10","Catalog_Name":"pja_upd","Folder":"Advanced","Catalog_Type":"Advanced","Template_Name":"Not
 Applicable",
"Catalog_Description":"","Cloud":"","Image":"","Group":"pja_sep","Icon":"/app/images/temp/1436492144835_ibm.png","OS":"",
"Additional_OS_Info":"","Applications":"","Additional_Application_Details":"","Status":"OK"},{"Catalog_ID":"4","Catalog_Name":"zmn_con",
"Folder":"Service Container","Catalog_Type":"Service
Container","Template_Name":"zmnACT","Catalog_Description":"","Cloud":"","Image":"",
"Group":"All
Groups","Icon":"/app/images/temp/1436514875643_container_clear_64x64.png","OS":"","Additional_OS_Info":"","Applications":"",
"Additional_Application_Details":"","Status":"OK"},{"Catalog_ID":"9","Catalog_Name":"ayc_LB","Folder":"Service
 Container","Catalog_Type":
"Service
Container","Template_Name":"aycACT","Catalog_Description":"","Cloud":"","Image":"","Group":"apREGsep9_org1","Icon":
"/app/images/temp/1436514875643_container_clear_64x64.png","OS":"","Additional_OS_Info":"",
"Applications":"","Additional_Application_Details":"","Status":"OK"},{"Catalog_ID":"11","Catalog_Name":"pja_con","Folder":
"Service Container","Catalog_Type":"Service
Container","Template_Name":"pja_tmp","Catalog_Description":"","Cloud":"","Image":"",
"Group":"pja_sep","Icon":"/app/images/temp/1436514875643_container_clear_64x64.png","OS":"","Additional_OS_Info":"","Applications":"",
"Additional_Application_Details":"","Status":"OK"},{"Catalog_ID":"12","Catalog_Name":"prsConCat","Folder":"Service
```

```
 Container",
"Catalog_Type":"Service
Container","Template_Name":"prsACT","Catalog_Description":"","Cloud":"","Image":"","Group":"arpAPIReg2_Org",
"Icon":"/app/images/temp/1436514875643_container_clear_64x64.png","OS":"","Additional_OS_Info":"","Applications":"",
"Additional_Application_Details":"","Status":"OK"}],"columnMetaData":null,"reportParams":{}},
"serviceError":null, "serviceName":"InfraMgr",
 "opName":"userAPIGetAllCatalogs" }
```


Step 2 Choose a catalog (for example, pja_con) that is used for creating a service container to provision a VM and submit a service request using the userAPIServiceContainerCatalogRequest API to create a service container using the chosen catalog.
In this example, a service container called SCN_Name is created using the pja_con catalog.

Request


```
/app/api/rest?formatType=json&opName=userAPIServiceContainerCatalogRequest&opData={param0:
{"catalogName":"pja_con","groupName":"Default
Group","serviceContainerName":"SCN_Name","apiResourceLimits":
null,"networkThroughput":"1G","enableNetworkMgmt":true,"customTierLabels":[{"name":"web","value":"web"}],
"comments":"test"}}
```


Response


```
{"serviceResult":728, "serviceError":null, "serviceName":"InfraMgr",
"opName":"userAPIServiceContainerCatalogRequest"}
```


The URL returns the service request ID. The service request ID for creating the service container is 728.

Step 3 (Optional) After creating the service request, get the details regarding the service request and the related workflow steps using the userAPIGetServiceRequestWorkFlow API. The SR ID (728) is passed from the Step 2 response.

Request


```
/app/api/rest?formatType=json&opName=userAPIGetServiceRequestWorkFlow&opData={param0:728}
```


Response


```
{
"serviceResult":{"requestId":728,"workflowCreated":1442274648591,"submittedTime":1442274648981,"cancelledTime":-1,
"cancelledByUser":null,"adminStatus":1,"executionStatus":2,"futureStartTime":1442274648591,"entries":[{"stepId":
"Initiated by
aks","executionStatus":3,"statusMessage":null,"handlerId":4,"startedTime":-1,"completedTime":1442274649606,
"validTill":-1,"startAfter":-1},{"stepId":"GetResourceRequirementFromThroughput","executionStatus":3,"statusMessage":"",
"handlerId":12,"startedTime":-1,"completedTime":1442274657097,"validTill":-1,"startAfter":-1},{"stepId":"Allocate
 APIC Container
Resources","executionStatus":2,"statusMessage":"Execution of the task resulted in
errors","handlerId":12,"startedTime":-1,
"completedTime":1442274662674,"validTill":-1,"startAfter":-1},{"stepId":"Verify Container Resource
Limits","executionStatus":0,
"statusMessage":null,"handlerId":12,"startedTime":-1,"completedTime":-1,"validTill":-1,"startAfter":-1},{"stepId":"If
 Else",
"executionStatus":0,"statusMessage":null,"handlerId":12,"startedTime":-1,"completedTime":-1,"validTill":-1,"startAfter":-1},
{"stepId":"APIC Reterive Secondary
Container","executionStatus":0,"statusMessage":null,"handlerId":12,"startedTime":-1,
"completedTime":-1,"validTill":-1,"startAfter":-1},{"stepId":"Trigger APIC Container - DR
Site","executionStatus":0,"statusMessage":null,
"handlerId":12,"startedTime":-1,"completedTime":-1,"validTill":-1,"startAfter":-1},{"stepId":"Create
 Tenant Application Profile",
"executionStatus":0,"statusMessage":null,"handlerId":12,"startedTime":-1,"completedTime":-1,"validTill":-1,"startAfter":-1},
{"stepId":"Create Private Network
","executionStatus":0,"statusMessage":null,"handlerId":12,"startedTime":-1,"completedTime":-1,
"validTill":-1,"startAfter":-1},{"stepId":"Trigger Multiple Container Tier
Creation","executionStatus":0,"statusMessage":null,
"handlerId":12,"startedTime":-1,"completedTime":-1,"validTill":-1,"startAfter":-1},{"stepId":"Wait
For Service Requests",
"executionStatus":0,"statusMessage":null,"handlerId":12,"startedTime":-1,"completedTime":-1,"validTill":-1,"startAfter":-1},
{"stepId":"Setup APIC Container Network
Connection","executionStatus":0,"statusMessage":null,"handlerId":12,"startedTime":-1,
"completedTime":-1,"validTill":-1,"startAfter":-1},{"stepId":"Create APIC Container
Contracts","executionStatus":0,"statusMessage":null,
"handlerId":12,"startedTime":-1,"completedTime":-1,"validTill":-1,"startAfter":-1},{"stepId":"Child
 workflow
(APIC Container Attached L4L7
Configuration)","executionStatus":0,"statusMessage":null,"handlerId":12,"startedTime":-1,"completedTime":-1,
```

```
"validTill":-1,"startAfter":-1},{"stepId":"Provision APIC Container
VMs","executionStatus":0,"statusMessage":null,"handlerId":12,
"startedTime":-1,"completedTime":-1,"validTill":-1,"startAfter":-1},{"stepId":"Re-Sync Container
VMs","executionStatus":0,"statusMessage":
null,"handlerId":12,"startedTime":-1,"completedTime":-1,"validTill":-1,"startAfter":-1},{"stepId":"If
 Else","executionStatus":0,
"statusMessage":null,"handlerId":12,"startedTime":-1,"completedTime":-1,"validTill":-1,"startAfter":-1},{"stepId":"Wait
 For Service

Requests","executionStatus":0,"statusMessage":null,"handlerId":12,"startedTime":-1,"completedTime":-1,"validTill":-1,"startAfter":-1},
{"stepId":"Child workflow
(APICContainerSRMSettings)","executionStatus":0,"statusMessage":null,"handlerId":12,"startedTime":-1,
"completedTime":-1,"validTill":-1,"startAfter":-1},{"stepId":"Initiate APIC Container BM
Provisioning","executionStatus":0,"statusMessage":
null,"handlerId":12,"startedTime":-1,"completedTime":-1,"validTill":-1,"startAfter":-1},{"stepId":"Send
 Container Email","executionStatus":
0,"statusMessage":null,"handlerId":12,"startedTime":-1,"completedTime":-1,"validTill":-1,"startAfter":-1},{"stepId":
"GetMSPAdminEmailAddresses","executionStatus":0,"statusMessage":null,"handlerId":12,"startedTime":-1,"completedTime":-1,"validTill":-1,
"startAfter":-1},{"stepId":"Send Container
Email","executionStatus":0,"statusMessage":null,"handlerId":12,"startedTime":-1,"completedTime":
-1,"validTill":-1,"startAfter":-1},{"stepId":"Complete","executionStatus":0,"statusMessage":null,"handlerId":13,"startedTime":-1,
"completedTime":-1,"validTill":-1,"startAfter":-1}]}, "serviceError":null, "serviceName":"InfraMgr",
 "opName":
"userAPIGetServiceRequestWorkFlow" }
```


In the response, the stepId represents the task executed by the workflow. On successful completion of the workflow execution, the stepId is represented as **Complete**. The service container called SCN_Name is created.

Step 4 Execute the service request using the userAPISubmitServiceRequest API to provision a VM.

Request

```
/app/api/rest?formatType=json&opName=userAPISubmitServiceRequest&opData={param0:"cat82",param1:"vdc82",
param2:1,param3:-1,param4:1,param5:"vm provisioning"}
```

Where,

- param0—The name of the catalog that is used for provisioning a VM.
- param1—The name of the virtual datacenter (VDC) on which the VM needs to be provisioned.
- param2—The duration of VM provisioning in hours. After the set duration, VM will be automatically deprovisioned. Use -1 to set the duration as indefinite.
- param3—This is an optional parameter. Schedule the time at which you want to start provisioning a VM. For example, January 1, 2014, 00:00:00 GMT. Use 0 or -1 to start the VM provisioning immediately.
- param4—The number of VM to be provisioned.
- param5—This is an optional parameter. Any comments on provisioning a VM.

Note When passing parameters in the REST API URL request, you must pass the parameters within the two single quotes (for example, param0: "catalogName"). If the parameter value includes any punctuations, your session will get hanged after validation.

Response

```
{ "serviceResult":456, "serviceError":null, "serviceName":"InfraMgr",
"opName":"userAPISubmitServiceRequest" }
```

The service request ID for provisioning a VM is 456.

Example: Rollback a Provisioned VM

When a provisioned VM is no longer required, you can use the rollback workflow to release and reallocate the resources allotted to that VM. A system administrator or an end user with Write - Group Service Request user permissions can roll back a workflow.

The roll back workflow must include a task to pass the ID of the service request that was used to provision the VM in the userAPIRollbackWorkflow API. The ID of the service request that is in progress is available in Cisco UCS Director (**Organizations** > **Service Requests**).

When you roll back a VM that was provisioned by another user, a rollback workflow approval is triggered to get approval from that user. The rollback workflow is completed after approval is received.

Request

The following REST URL rolls back the service request with the ID 456.

```
/app/api/rest?formatType=json&opName=userAPIRollbackWorkflow&opData={param0:456}
```

Response

```
{ "serviceResult":458, "serviceError":null, "serviceName":"InfraMgr",
"opName":"userAPIRollbackWorkflow" }
```

Check the status of the rollback workflow using the userAPIGetServiceRequestWorkFlow API as follows:

Request

```
/app/api/rest?formatType=json&opName=userAPIGetServiceRequestWorkFlow&opData={param0:458}
```

On successful completion of the rollback workflow execution, the stepId is represented as **Complete**. The resources allotted for the VM are released and made available for reallocation.

Exceptions

The userAPIRollbackWorkflow API throws exceptions on unsuccessful roll back of a workflow.

If you try to rollback a service request ID that is still in progress, the userAPIRollbackWorkflow API throws the following exception:

```
REMOTE_SERVICE_EXCEPTION: Cannot rollback work-flow for SR ID 332, when the work-flow
execution is in progress
```

If you try to rollback a service request ID that is rolled back already, the userAPIRollbackWorkflow API throws the following exception:

```
REMOTE_SERVICE_EXCEPTION: Cannot Rollback SR:332 as it is already rolled back.
```

Requesting JSON API using HTTP/HTTPS POST

In general, the JSON API request is sent using the HTTP GET method. You can also pass the JSON API request using the HTTP POST method. For example, while handling sensitive data, you can use the HTTP POST method.

The following example explains the format followed for passing JSON API request using the HTTP GET method and HTTP POST method:

HTTP GET method

Header:

X-Cloupia-Request-Key: {REST API Access Key}

URL:

```
https://{UCSD_IP}/app/api/rest?formatType=json&opName=userAPISubmitWorkflowServiceRequest&opData=
{"param0":"Post_Example","param1":{"list":[{"name":"Input1","value":"Russ1"},{"name":"Input2","value":"Russ2"}]},"param2":-1}
```

HTTP POST method

Header: For the POST method, the header must include both the API access key and content type.

X-Cloupia-Request-Key: {REST API Access Key}

Content-Type: application/x-www-form-urlencoded

URL

```
https://{UCSD_IP}/app/api/rest
```

You can pass the parameters as the request parameters or as a body text.

Table 1: Request Parameters:

Key	Value
formatType	json
opName	userAPISubmitWorkflowServiceRequest
opData	{"param0": "Post_Example", "param1": {"list": [{"name": "Input1","value": "Russ1"},{"name": "Input2","value": "Russ2"}]},"param2": -1}

Body Text

```
formatType=json&opName=userAPISubmitWorkflowServiceRequest&opData=
{"param0":"Post_Example","param1":{"list":[{"name":"Input1","value":"Russ1"},{"name":"Input2","value":"Russ2"}]},"param2":-1}
```

How to use Global Variables in REST API

The **REST API Browser** provides the CREATE, READ, UPDATE, and DELETE operations for global variables. Click each operation and enter the required details as follows to execute the operations:

- CREATE**—**In the **SAMPLE XML** field of the **API Examples** tab, enter values in the <varName>, <description>, and <value> tags. The <varName> and <value> tags are mandatory, whereas the <description> tag is optional. Leave the <defaultVariable> tag empty. By defaullt, the <defaultVariable> tag is set as false.

  Click **Execute REST API** to get the response of the create operation in the **Response** field.

- READ**—**In the **Resource URL** field of the **API Examples** tab, append the */GlobalvariableName* to the URL. For example, /cloupia/api-v2/GlobalVariables/TEST_MACRO.

  Click **Execute REST API** to get the response of the read operation in the **Response** field.

- UPDATE**—**In the **Resource URL** field of the **API Examples** tab, replace {varName} with the user defined global variable name and provide the values that need to be updated in the **SAMPLE XML** field. Ensure that you provide same variable name in the **Resource URL** field and <varName> tag of the **SAMPLE XML** field.

  Click **Execute REST API** to get the response of the update operation in the **Response** field.

- DELETE**—**In the **Resource URL** field of the **API Examples** tab, replace {varName} with the user defined global variable name to delete the record of the global variable name. Click **Execute REST API** to get the response of the delete operation in the **Response** field.