CISCO

Cisco UCS Director REST API Getting Started Guide, Release 6.7

First Published: 2020-06-18

Last Modified: 2020-08-13

Getting Started with Cisco UCS Director REST API

New and Changed Information

The following table provides an overview of the significant changes to this guide for the current release. The table does not provide an exhaustive list of all changes, or of all new features in this release.

Table 1: New Features and Changed Behavior in Cisco UCS Director, Release 6.7

Feature	What's New	Where Documented
New APIs for Cisco ACI APIC Connector Pack	Provided newly introduced APIs in Cisco ACI APIC Connector Pack, Release 6.7.4.1	New APIs in Cisco ACI APIC Connector Pack in Release 6.7.4.1, on page 7

Overview

Why Use the 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.

The API accepts and returns HTTP messages that contain JavaScript Object Notation (JSON) or Extensible Markup Language (XML) documents. The JSON or XML payload contained in an HTTP message describes a method or managed object (MO) in Cisco UCS Director. You can use any programming language to generate the messages and the JSON or XML payload.

How the API Works

In RESTful APIs, the HTTP method specifies the action you want to perform and the URI specifies the resource you want to access.

REST API uses the following HTTP methods to perform create, read, update, and delete (CRUD) operations:

HTTP Method	Description
GET	Retrieves the specified resource. GET is a read-only operation that does not change the engine state. • The HTTP GET operation must not have a request body. If information is passed in a GET request, query parameters must be used instead. • Unless specified, the HTTP GET operation returns the configured state. For example, an HTTP GET operation of the global routing table returns the dynamic run-time state.
POST	Submits the data to be processed by the specified resource. The data to be processed is included in the request body. A POST operation can create a new resource. • Every POST request must include a JSON body containing a definition of the new resource. • For a POST operation to create a new resource, the location header in the HTTP response must contain the complete URL to be used for subsequent PUT, GET, and DELETE commands. • The HTTP POST response to a create request must have a 201 return code and a location header containing the URI of the newly created resource in the HTTP header.
PUT	Updates the specified resource with new information. The data that is included in the PUT operation replaces the previous data. • The PUT operation cannot be used to create a new resource. • The request body of a PUT operation must contain the complete representation of the mandatory attributes of the resource in JSON format.
DELETE	Deletes a resource. • If you delete a resource that has already been deleted, a `404 Not Found` response is returned. • The HTTP DELETE operation must not have a request body. If information is passed in a GET request, query parameters must be used instead.

How to Use the REST API

To access the REST API browser through Cisco UCS Director, a valid Cisco UCS Director user account and an API access key are needed. Cisco UCS Director uses the API access key to authenticate API requests. This access key is a unique security access key code that is associated with a specific Cisco UCS Director user account. For more information about how to generate an API access key, see Generating an API Access Key, on page 4.

You must pass the REST API access key as a *name:value* header following standard HTTP syntax and semantic rules. For example, a valid *name:value* header is *X-Cloupia-Request-Key: F90ZZF12345678ZZ90Z12ZZ3456FZ789*. For more information about the API request header, see #unique_ 9 and RFC2616 Header Field Definitions.

The REST API call can be made in one of the following ways:

- Cisco UCS Director GUI—Cisco UCS Director provides a developer menu option to offer the report metadata and REST API Browser for developers. To access these features, enable the developer menu. For more information about how to enable the developer menu, see Enabling the Developer Menu Options, on page 12.

  On enabling the developer menu, you gain access to the following features:

  - Report Metadata—The report metadata enables you to view the REST API URL for every report displayed in Cisco UCS Director. For more information about how to access Report Metadata, see Accessing the Report Metadata, on page 14.

  - REST API Browser—The REST API Browser is accessible from the **Orchestration** menu of Cisco UCS Director. The REST API Browser provides API information and API code generation capabilities that make it easy to see and work with all the available APIs, including both the REST APIs and the Java APIs. For more information about how to access REST API Browser, see Using the REST API Browser, on page 13.

- REST Client—The REST Client is a useful widget for parsing and viewing API requests and responses. In this widget, you can enter a REST URL and apply an HTTP method such as POST, PUT, or DELETE to the URL for data manipulation. The REST Client provides a simple user interface for entering a URL to fetch data from the Cisco UCS Director server.

  - If you are using Mozilla Firefox, download RESTClient from Add-ons for Firefox.

  - If you are using Google Chrome, download Advanced REST Client from the Chrome Web Store.

  Note If you are logged in to Cisco UCS Director, use any supported web browser to send API requests and get responses.

- Code—The **Sample Java Code** tab of the **REST API Browser** tab provides a code snippet that can be used to automate the management services.

  The Cisco UCS Director SDK binary downloaded from the software download area or the DevNet site includes a sample project that can be used as a platform for executing client code.

Prerequisites

Before you start using the Cisco UCS Director SDK, make sure that:

- Cisco UCS Director is installed and running on your system. For more information about how to install Cisco UCS Director, refer the Cisco UCS Director Installation Guide.

- The Cisco UCS Director SDK is installed on your system.

- You have Java Version 1.8 for your Java development environment.

> **Note** You need Java Version 1.6 if you have release 5.3 or earlier of Cisco UCS Director

- You have an API access key. For more information about how to generate an API access key, see Generating an API Access Key, on page 4.

- You have a REST client to execute RESTful web services.

> **Note**
> - If you are using Mozilla Firefox, download RESTClient from Add-ons for Firefox.
>
> - If you are using Google Chrome, download Advanced REST Client from the Chrome Web Store.

Generating an API Access Key

Procedure

Step 1	Log in to Cisco UCS Director.
Step 2	Hover the mouse over the user icon at the top right corner and choose **Edit My Profile** from the drop-down list.
Step 3	On the **Edit My Profile** page, click **Show Advanced Settings**.
Step 4	To copy the value displayed in the **REST API Access Key** area, click **Copy Key Value**.
Step 5	Save the access key in a secure location, and use it in the API request header.
	For more information about the API request header, see #unique_9.
Step 6	If you want to change the API access key, click **Regenerate Key**.
	After you generate a new key, the old key code is retired and you cannot use it.

Downloading and Installing Cisco UCS Director REST API SDK Bundle

The Cisco UCS Director REST API SDK Bundle

The Cisco UCS Director REST API SDK Bundle is part of the Cisco UCS Director REST API. In addition to documentation such as the Cisco UCS Director REST API cookbook, the SDK Bundle provides examples that you can use with the REST API. These examples include test cases and sample code that demonstrates the use of the SDK classes.

The SDK Bundle is delivered in an archive file named `ucsd-rest-api-sdk-v2.zip`.

Installing the Cisco UCS Director SDK

Procedure

Step 1 Download the digitally signed zip file (`CUSD_SDK_BINARIES_xxxxx_SIGNED.zip`) from the software download area or the DevNet site.

Step 2 Unzip the `CUCSD_SDK_BINARIES_xxxxx_SIGNED.zip` file. You will get the `CUCSD_SDK_BINARIES_xxxxx.zip` file and other supporting digital signature files.

Unzip the `CUSD_SDK_BINARIES_xxxxx.zip` file. The following files appear:

- `cucsd_script_code_samples-x.x.x.x`

- `cucsd_cloupia-script-bundle-x.x.x.x`

- `cucsd-open-auto-sdk-bundle-x.x.x.x`

- `cucsd-rest-api-sdk-bundle-x.x.x.x`

- `cucsd-sdk-javadocs-x.x.x.x`

- `UCSDirector_PSC.exe`

Step 3 Unzip the `cucsd-rest-api-sdk-bundle-x.x.0.0.zip` file. This expands and provides the `ucsd-rest-api-sdk-x.x.0.0.zip` and `ucsd-rest-api-sdk-javadocs-x.x.0.0.zip` files.

Step 4 Unzip the `ucsd-rest-api-sdk-x.x.0.0.zip` file.

The `ucsd-rest-api-sdk-x.x.0.0` folder is created on the local drive.

This folder includes the following folders and files to assist you in developing applications with the REST API:

- `lib`—This folder contains the `ucsd-rest-api-sdk-v2.0.jar` file. This file is the archive of all classes and libraries required for managing the Cisco UCS Director.

- `src`—This folder contains the example test cases and sample code. These tests cases and codes demonstrate the use of the SDK classes in the `ucsd-rest-api-sdk-v2.jar` file. The sample code is available in the `src\com\cisco\cuic\api\examples` folder.

- `.classpath`—This file indicates where to look for SDK classes and packages, to the Java Virtual Machine (JVM) or Java compiler.

• `.project`—This file is used to import the SDK Bundle into the Eclipse IDE. This file is required for development purposes. It is available at the top level of the archive.

• `README.txt`—This file contains lists of all components included in the SDK Bundle.

• `rest-server.properties`—This file contains the properties used by the examples in the `src\com\cisco\cuic\api\examples` folder.

Step 5   Unzip the `ucsd-rest-api-sdk-javadocs-x.x.0.0.zip` file.

The `ucsd-rest-api-sdk-javadocs-x.x.0.0` folder is created on the local drive, and it contains:.

• `javadocs`—This folder contains the *Cisco UCS Director REST API Javadocs*. The Javadocs cover all classes provided with the SDK Bundle.

Step 6   Add the `ucsd-rest-api-sdk-v2.jar` file to the Java classpath when compiling and running the client program. Use Java Version 1.8 to run the Cisco UCS Director SDK.

Note   For Cisco UCS Director Releases 5.3 and previous releases, use Java version 1.6 to run the Cisco UCS Director SDK.

Downloading the REST API SDK Bundle

The Cisco UCS Director SDK binaries can be downloaded from the software download area or the DevNet site. Alternatively, an admin user and an end user can download the SDK binaries from Cisco UCS Director.

Procedure

Step 1   To download the REST API SDK bundle as an admin user, complete the following steps:

a) Log in to Cisco UCS Director.

b) Choose **Administration** > **Downloads**.

The following files are displayed:

• REST API SDK—cucsd-rest-api-sdk-bundle.zip

• PowerShell Console—UCSDirector_PSC.exe

• Open Automation SDK—cucsd-open-auto-sdk-bundle.zip

• Custom Tasks Script Samples—cucsd-cloupia-script-bundle.zip

c) Choose the **cucsd-rest-api-sdk-bundle.zip** file.

d) Click **Download** to download the SDK bundle.

Step 2   To download the REST API SDK bundle as an end user, complete the following steps:

a) Log in to the Cisco UCS Director End User Portal.

b) On the menu bar, click **Services**.

c) Click the **Downloads** tab.

The following files are displayed:

> • REST API SDK—cucsd-rest-api-sdk-bundle.zip
>
> • PowerShell Console—console.exe

d) Choose the **cucsd-rest-api-sdk-bundle.zip** file and click **Download**.
The file is downloaded to the default download location.

List of Deprecated APIs

Developers are encouraged to migrate to the new API, instead of using the deprecated API.

Table 2: Deprecated and New APIs

Deprecated API	Supported Till	New API
userAPICheckFunds	Release 4.1	From Cisco UCS Director Release 5.x and later releases, the UserAPICheckFunds API that is available in the **userAPIGlobal** folder is deprecated. To check the available customer funds, use the UserAPICheckFunds API in the **userAPIChargeBack** folder.
userAPISubmitServiceRequest	Release 4.1	userAPIProvisionRequest
userAPISubmitServiceRequestCustom	Release 4.1	userAPIProvisionRequest
UserAPIProvisionRequestWithPortGroup	Release 4.1	userAPIVMWareProvisionRequest
userAPIGetWorkflows	Release 4.1	userAPIGetWorkflowInputs
userAPIImportWorkflows	Release 5.3	
UserAPICreateAPICContainer	Release 5.4	
userAPIResetMyPassword	Release 5.5	userAPIModifyLoginProfilePassword
userAPIResetUserPassword	Release 5.5	userAPIModifyUserPassword
userAPIAddTierToContainerVM	Release 5.5	userAPIAddTierToContainer

New APIs in Cisco ACI APIC Connector Pack in Release 6.7.4.1

Following are the APIs introduced in Cisco ACI APIC Connector Pack, Release 6.7.4.1.

Priority Flow Control Policy

> • CREATE_APIC_PRIORITY_FLOW_CONTROL_POLICY
>
> • UPDATE_APIC_PRIORITY_FLOW_CONTROL_POLICY
>
> • DELETE_APIC_PRIORITY_FLOW_CONTROL_POLICY

MACsec Access Parameters Policy

- CREATE_APIC_MAC_SEC_ACCESS_PARAMETERS_POLICY

- UPDATE_APIC_MAC_SEC_ACCESS_PARAMETERS_POLICY

- DELETE_APIC_MAC_SEC_ACCESS_PARAMETERS_POLICY

Slow Drain Policy

- CREATE_APIC_SLOW_DRAIN_POLICY

- UPDATE_APIC_SLOW_DRAIN_POLICY

- DELETE_APIC_SLOW_DRAIN_POLICY

Port Security Policy

- CREATE_APIC_FABRIC_PORT_SECURITY_POLICY

- UPDATE_APIC_FABRIC_PORT_SECURITY_POLICY

- DELETE_APIC_FABRIC_PORT_SECURITY_POLICY

Fibre Channel Interface Policy

- CREATE_APIC_FABRIC_FIBRE_CHANNEL_INTERFACE_POLICY

- UPDATE_APIC_FABRIC_FIBRE_CHANNEL_INTERFACE_POLICY

- DELETE_APIC_FABRIC_FIBRE_CHANNEL_INTERFACE_POLICY

MACsec Access Interface Policy

- CREATE_APIC_FABRIC_MACSEC_INTERFACE_POLICY

- UPDATE_APIC_FABRIC_MACSEC_INTERFACE_POLICY

- DELETE_APIC_FABRIC_MACSEC_INTERFACE_POLICY

Layer 2 Interface Policy

- CREATE_L2_INTERFACE_POLICY_IN_APIC

- UPDATE_L2_INTERFACE_POLICY_IN_APIC

- DELETE_L2_INTERFACE_POLICY_IN_APIC

Monitoring Policy

- CREATE_APIC_FABRIC_MONITORING_POLICY

- DELETE_APIC_FABRIC_MONITORING_POLICY

VSPAN Destination Group

- CREATE_APIC_FABRIC_VSPAN_DESTINATION_GROUP

- DELETE_APIC_FABRIC_VSPAN_DESTINATION_GROUP

Port Channel Member Policy

- CREATE_APIC_FABRIC_PORT_CHANNEL_MEMBER_POLICY

- UPDATE_APIC_FABRIC_PORT_CHANNEL_MEMBER_POLICY

- DELETE_APIC_FABRIC_PORT_CHANNEL_MEMBER_POLICY

MACsec KeyChain Policy

- CREATE_MACSEC_KEYCHAIN_POLICY_IN_APIC

- UPDATE_MACSEC_KEYCHAIN_POLICY_IN_APIC

- DELETE_MACSEC_KEYCHAIN_POLICY_IN_APIC

Spanning Tree Interface Policy

- APIC_CREATE_STP_INTERFACE_POLICY

- APIC_UPDATE_STP_INTERFACE_POLICY

- APIC_DELETE_STP_INTERFACE_POLICY

MACsec Key Policy

- CREATE_APIC_FABRIC_MACSEC_KEY_POLICY

- UPDATE_APIC_FABRIC_MACSEC_KEY_POLICY

- DELETE_APIC_FABRIC_MACSEC_KEY_POLICY

EIGRP Address Family Context Policy

- CREATE_EIGRP_ADDRESS_FAMILY_CONTEXT_POLICY

- UPDATE_EIGRP_ADDRESS_FAMILY_CONTEXT_POLICY

- DELETE_EIGRP_ADDRESS_FAMILY_CONTEXT_POLICY

Fabric External Collector Reachability

- CREATE_APIC_FABRIC_EXTERNAL_COLLECTOR_REACHABILITY

- UPDATE_APIC_FABRIC_EXTERNAL_COLLECTOR_REACHABILITY

- DELETE_APIC_FABRIC_EXTERNAL_COLLECTOR_REACHABILITY

VSPAN Session

- APIC_CREATE_VSPAN_SESSION

- APIC_UPDATE_VSPAN_SESSION

- APIC_DELETE_VSPAN_SESSION

BGP Route Target Profile

- ADD_APIC_BGP_ROUTE_TARGET_PROFILE_TO_VRF

- REMOVE_APIC_BGP_ROUTE_TARGET_PROFILE_FROM_VRF

NetFlow Monitor Policy

- ASSOCIATE_APIC_FABRIC_FLOW_EXPORTER_TO_NETFLOW_MONITOR_POLICY

- DISASSOCIATE_APIC_FABRIC_FLOW_EXPORTER_FROM_NETFLOW_MONITOR_POLICY

BGP Address Family Context Policy

- CREATE_BGP_ADDRESS_FAMILY_CONTEXT_POLICY

- UPDATE_BGP_ADDRESS_FAMILY_CONTEXT_POLICY

- DELETE_BGP_ADDRESS_FAMILY_CONTEXT_POLICY

Route Target to BGP Route Target Profile

- ADD_ROUTE_TARGET_TO_APIC_BGP_ROUTE_TARGET_PROFILE

- REMOVE_ROUTE_TARGET_FROM_APIC_BGP_ROUTE_TARGET_PROFILE

BGP Context Per Address Family

- ADD_APIC_BGP_CONTEXT_PER_ADDRESS_FAMILY_TO_VRF

- DELETE_APIC_BGP_CONTEXT_PER_ADDRESS_FAMILY_FROM_VRF

Community Profile

- CREATE_APIC_COMMUNITY_PROFILE_TO_SNMP_CONTEXT

- UPDATE_APIC_COMMUNITY_PROFILE_TO_SNMP_CONTEXT

- DELETE_APIC_COMMUNITY_PROFILE_FROM_SNMP_CONTEXT

Flow Record

- CREATE_APIC_FABRIC_FLOW_RECORD

- UPDATE_APIC_FABRIC_FLOW_RECORD

- DELETE_APIC_FABRIC_FLOW_RECORD

IGMP Snoop Policy

- CREATE_IGMP_SNOOP__POLICY

- UPDATE_IGMP_SNOOP__POLICY

- DELETE_IGMP_SNOOP__POLICY

SNMP Context

- CREATE_APIC_SNMP_CONTEXT

- DELETE_APIC_SNMP_CONTEXT

OSPF Context Per Address Family

- ADD_APIC_OSPF_CONTEXT_PER_ADDRESS_FAMILY_TO_VRF

- DELETE_APIC_OSPF_CONTEXT_PER_ADDRESS_FAMILY_FROM_VRF

Tenant Flow Record

- CREATE_APIC_TENANT_FLOW_RECORD

- UPDATE_APIC_TENANT_FLOW_RECORD

- DELETE_APIC_TENANT_FLOW_RECORD

Source Path

- ASSOCIATE_APIC_FABRIC_SOURCE_PATH_TO_VSPAN_SOURCE

- DISASSOCIATE_APIC_FABRIC_SOURCE_PATH_FROM_VSPAN_SOURCE

Source

- ADD_SOURCE_TO_APIC_FABRIC_VSPAN_SESSION

- UPDATE_SOURCE_TO_APIC_FABRIC_VSPAN_SESSION

- REMOVE_SOURCE_FROM_APIC_FABRIC_VSPAN_SESSION

MLD Snoop Policy

- CREATE_MLD_SNOOP__POLICY

- UPDATE_MLD_SNOOP__POLICY

- DELETE_MLD_SNOOP__POLICY

NetFlow Monitor Policy

- ADD_APIC_NETFLOW_MONITOR_POLICY_TO_BRIDGE_DOMAIN

- REMOVE_APIC_NETFLOW_MONITOR_POLICY_FROM_BRIDGE_DOMAIN

BGP Timers Policy

- CREATE_BGP_TIMERS

- UPDATE_BGP_TIMERS

- DELETE_BGP_TIMERS

FHS BD Policy

- CREATE_FHS_BD_POLICY

- UPDATE_FHS_BD_POLICY

- DELETE_FHS_BD_POLICY

DHCP Option Policy

- CREATE_DHCP_OPTION_POLICY

- DELETE_DHCP_OPTION_POLICY

DHCP Option

- ADD_APIC_DHCP_OPTION_TO_DHCP_OPTION_POLICY

- UPDATE_APIC_DHCP_OPTION_TO_DHCP_OPTION_POLICY

- REMOVE_APIC_DHCP_OPTION_FROM_DHCP_OPTION_POLICY

Fabric NetFlow Monitor Policy

- CREATE_APIC_FABRIC_NETFLOW_MONITOR_POLICY

- UPDATE_APIC_FABRIC_NETFLOW_MONITOR_POLICY

- DELETE_APIC_FABRIC_NETFLOW_MONITOR_POLICY

Tenant Monitoring Policy

- CREATE_APIC_TENANT_MONITORING_POLICY

- DELETE_APIC_TENANT_MONITORING_POLICY

Data Plane Policing Policy

- CREATE_APIC_FABRIC_DATE_PLANE_POLICING_POLICY

- UPDATE_APIC_FABRIC_DATE_PLANE_POLICING_POLICY

- DELETE_APIC_FABRIC_DATE_PLANE_POLICING_POLICY

Setting Up the Development Environment

Setting up the Environment for Using the REST API Through the GUI

Enable the developer menu option to access the REST API Browser and Report Metadata information in Cisco UCS Director. The REST API Browser and Report Metadata features provide you with site-specific API data.

The HTTP request code provided by the Report Metadata view yields immediate API service results. You can use these options in every situation where you need API information.

The Cisco UCS Director REST API Browser provides API information and API code generation capabilities that make it easy to see and work with all of the available APIs, including both the REST APIs and the Java APIs.

To use REST API in the GUI, perform the following tasks:

Enabling the Developer Menu Options

Before you begin

Obtain one or more user accounts that provide the same administrative access to data that your application users will have. Your Cisco UCS Director administrator can explain the data access limitations associated with different administrator and end-user roles. You may want multiple user accounts to test the user experiences associated with different data accesses and security controls.

Procedure

Step 1   In Cisco UCS Director, hover the mouse over the user icon at the top right corner and choose **Edit My Profile** from the drop-down list.

Step 2   On the **Edit My Profile** page, click **Show Advanced Settings**.

Step 3   Check **Enable Developer Menu (requires re-login)**.

The **REST API Browser** is activated in the **Orchestration** page, and the **Report Metadata** option becomes available in the report views.

> **Tip**   The **Advanced** area displays the REST API Access Key code for the account.

Step 4   Click **Close**.

Using the REST API Browser

The Cisco UCS Director REST API Browser provides API information and API code generation capabilities that assist and educate developers in the use of all available Cisco UCS Director APIs, including the XML-formatted REST API and the Java API. The primary view lists the Task folders that contain the APIs. The task names supply the categories under which the APIs are listed. For example, all the APIs pertaining to NetApp ONTAP tasks and NetApp OnCommand tasks are available inside the folders with these names.

Note   The access for XML-based version-2 APIs with the /api/v2 context is role-based. Service end users cannot execute the GET Operations with these APIs. Also, service end users can perform the create, update and delete operations using the REST API only if they can perform the same operation using the graphical user interface of Cisco UCS Director. If these users cannot perform a create, update or delete operation in the user interface, then they cannot perform these actions using the REST API either.

Before you begin

- Obtain one or more user accounts that provide the same administrative access to data that your application users will have. Your Cisco UCS Director administrator can explain the data access limitations associated with different administrator and end-user roles. You may want multiple user accounts to test the user experiences associated with different data accesses and security controls.

- Enable the developer menu option for the session.

Procedure

Step 1   Choose **Orchestration**.

Step 2   Click **REST API Browser**.

Click the right scroll arrow, if necessary, to view and navigate to **REST API Browser** menu item.

Step 3   Open the task folder that contains the API you want to view.

Tip   You can use the **Search** field at the top right corner to find a specific API if you do not know which task folder it belongs to. Enter a word(s) that represents the API Resource, Operation, or Description field to narrow your search. You can also use the other options on that menu bar, such as the **Add Advanced Filter**, to find a specific API.

Step 4   Double-click a row that contains an API resource and operation that is required.

The REST API browser screen displays the following:

- **API Examples**—Displays the API data for your selection and enables you to generate a sample XML. Depending on the operation and resource that you have selected, this area might also include data entry boxes that accept parameter values. If the data has to be chosen from a list of data, such as group list available in Cisco UCS Director, the data is listed out in a table. Choose a data and click **Select** to generate sample XML based on the data.

  From the **Server** drop-down list, choose a server on wich the API has to be executed. Click **Execute REST API** to view the response of the sample XML.

- **Details**—Provides more details about the API, including the API definition, input parameters, and output parameters.

- **Sample Java Code**—Provides sample code for the API.

Accessing the Report Metadata

Report Metadata enables you to view the API code used by Cisco UCS Director, including the API request code for every report displayed in Cisco UCS Director. This code includes a complete URL that is ready to paste into a browser to send the URL request to Cisco UCS Director. The immediate API responses provide information for the developer. To see the API request code, navigate to a report and select **Report Metadata**.

Before you begin

- Obtain one or more user accounts that provide the same administrative access to data that your application users will have. Your Cisco UCS Director administrator can explain the data access limitations associated with different administrator and end-user roles. You may want multiple user accounts in order to test the user experiences associated with different data accesses and security controls.

- Enable the Developer Menu option for the session.

Procedure

Step 1   In Cisco UCS Director, navigate to the page for which you want to see the API code.

For example, click **Orchestration**.

Step 2   On the **Workflow** page, click **Report Metadata**.

Step 3   In the **Information** screen, review the report context, report definition, API parameters, and REST API URL.

Setting up the Environment for Using the REST API Through the REST Client

The REST Client provides a standalone client that parses and labels the API data in a useful and informative way. You can also use any supported web browser to send API requests and get responses. Download a supported REST Client in a web browser to execute the REST URLs.

- In Mozilla Firefox, download RESTClient from Add-ons for Firefox.

- In Google Chrome, download Advanced REST Client from the Chrome Web Store.

Using the REST Client, you can execute the JSON APIs and XML APIs.

The sample JSON-based API URL to retrieve all catalogs using the GET method:

```
http://<serverip>/ app/api/rest?formatType=json&opName=userAPIGetAllCatalogs&opData={}
```

The sample XML-based API URL to create a user using the POST method:

```
http://serverip/cloupia/api-v2/user

<cuicOperationRequest>
<payload>
<![CDATA[
<AddUserConfig>
<userType>Admin</userType>
<!-- Accepts value from the list: userGroupByType-->
<userGroup>1</userGroup>
<mspOrganization></mspOrganization>
<loginName>apiuser</loginName>
<!-- Accepts value from the list: password-->
<password>bGthbmRhc2E=</password>
<!-- Accepts value from the list: password-->
<confirmPassword>bGthbmRhc2E=</confirmPassword>
<userContactEmail>apiuser@cisco.com</userContactEmail>
<firstName>API</firstName>
<lastName>User</lastName>
<phone></phone>
<address></address>
</AddUserConfig>

]]>
</payload>
</cuicOperationRequest>
```

The HTTP response of the user creation URL:

```
HTTP Response Code : 200
Response data :
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><cuicOperationResponse>
<Log><Messages>1</Messages><Message><TimeStamp>2015-08-28 06:55:17.240</TimeStamp>
<Severity>INFO</Severity><Text>User Admin Created Successfully</Text>
</Message></Log><operationStatus>0</operationStatus><response><AddUserConfigResponse>
<OUTPUT_USER_NAME>apiuser</OUTPUT_USER_NAME></AddUserConfigResponse></response>
<responseMap><entry><key>OUTPUT_USER_NAME</key><value>apiuser</value></entry>
</responseMap></cuicOperationResponse>
```

Setting up the Environment for Using the REST API Through Code

Import the Cisco UCS Director SDK bundle as a Java project into the Eclipse IDE to execute the code in Eclipse. For more information about how to import the Cisco UCS Director SDK bundle, see Importing the SDK Bundle Project into the Eclipse IDE, on page 16. The code can be taken from the **Sample Java Code**

tab of the **REST API Browser**. For more information about how to access the REST API Browser, see Using the REST API Browser, on page 13.

✎

Note   The instruction for using the Cisco UCS Director SDK bundle in Eclipse is captured in the document. The similar process is applicable for other IDEs but developers must figure out the details for themselves.

Importing the SDK Bundle Project into the Eclipse IDE

Before you begin

Obtain the SDK Bundle archive and extract the contents to an appropriate folder.

Procedure

Step 1   In the Eclipse IDE, choose **File** > **New** > **Java Project**.
The **Create a Java Project** dialog box appears.

Step 2   In the **Project Name** field, enter a name for the project.

Step 3   Right click the project, and select **Import**.
The **Import** dialog box appears.

Step 4   Select **File System** and click **Next**.

Step 5   Click **Browse** and navigate to the folder where you have extracted the SDK Bundle.

Step 6   Click **Finish**.
The Eclipse IDE displays the SDK Bundle project on the **Project Explorer** tab.

How to Make a REST API Request

API clients use an HTTP request to interact with Cisco UCS Director. To pass the REST API access key, each request must be associated with an HTTP header called `X-Cloupia-Request-Key` with its value set to the current REST API access key. For information about how to generate the REST API access key, see Generating an API Access Key, on page 4.

Requests made to the API have the following characteristics:

- Requests are sent over HTTP.

- Request must contain a valid URL in the one of the following formats:

  - **JSON Format**

    ```
    http://SERVER/app/api/rest?formatType=json&opName=operationName&opData=
    operationData
    ```

    where:

      - **SERVER**—The IP address or the hostname of the Cisco UCS Director VM.

- **formatType**—The only supported format that is discussed here is JavaScript Object Notation (JSON). Set this parameter value to json.

- **opName**—The API operation name associated with the request. For example, `userAPIGetMyLoginProfile` or `userAPIGetVMActionStatus`.

- **opData**—Parameters (or arguments) associated with the operation. Cisco UCS Director uses JSON encoding of the parameters. If no arguments are required for the operation, use {} as an empty set. Before you send JSON data in a request, encode the URL by applying escape characters as appropriate. For details about encoding the URL, see the RFC at http://www.ietf.org/rfc/rfc1738.txt. For more information about JSON syntax and data types, see http://en.wikipedia.org/wiki/JSON#Data_types.2C_syntax_and_example. For information about non-JSON formatted API requests, see Using the REST API Browser, on page 13.

> ✎
>
> **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.

- **XML Format**

```
http://server/cloupia/api-v2/group
```

HTTP method: POST

```
<cuicOperationRequest>
<payload>
<![CDATA[
<AddGroupConfig>
<groupName>TestGroup</groupName>
<groupDescription></groupDescription>
<parentGroup>0</parentGroup>
<groupCode></groupCode>
<groupContact>jbesai@cisco.com</groupContact>
<firstName></firstName>
<lastName></lastName>
<phone></phone>
<address></address>
<groupSharePolicyId></groupSharePolicyId>
<allowPrivateUsers>false</allowPrivateUsers>
</AddGroupConfig>
]]>
</payload>
</cuicOperationRequest>
```

For REST API examples, refer the Cisco UCS Director REST API Cookbook.

Note If the variable passed in the REST URL has special characters, the variable must be enclosed within single quotes (') as shown in the following example:

Resource URL Format

```
/cloupia/api-v2/datacenter/{datacenter}/NetworkInventory/{networkInventoryID}/VLAN/{VLANID}
```

Resource URL

```
/cloupia/api-v2/datacenter/Default Pod/NetworkInventory/'172.29.195.215'/VLAN/VLAN0800
```

About Operations Data Parameters or Arguments

As the method and the API resource type are communicated through the opName, the operation parameters must present any arguments that you want to designate a specific instance of the resource to be operated upon.

Operations Data Parameter Syntax

The following table shows examples of operations data parameter syntax in JSON format.

If the operation needs the following parameters (opData)	How to represent in JSON
No parameters	{}
One parameter; integer (for example, 10)	{param0:10}
One parameter: string (for example, cloud)	{param0:"cloud"}
Two parameters: a string and an Integer	{param0:"cloud",param1:10}
Two parameters: a string with null value and an Integer	{param0:null,param1:10}
Three parameters	{param0:"cloud",param1:"cloupia",param2:100}

Operation Data Parameter Examples

```
…&opData={param0:"datacenter",param1:"DataCenter1",param2:"STORAGE-ACCOUNTS-T51"
```

```
…&opData={param0:"Create NFS Datastore",param1:{"list":[{"name":"Volume Size","value":100},
{"name":"Select Group","value":"14"},{"name":"Select vDC","value":18}]},param2:212}
```

- `param0`—Name of the workflow being invoked through the REST API.

- `param1`—Input being passed to the workflow. If there is more than one input, separate the inputs with commas and put two single quotation marks around the input names and values. If there are no inputs, use the keyword null as the parameter value.

- `param2`—If this workflow is being invoked as a child workflow of another service request, use the service request (SR) ID. If this workflow is not invoked as a child workflow, use −1. When −1 is used, a new service request is created.

🔍

Tip Cisco UCS Director provides many complete API requests, formatted as URLs and ready for you to cut and paste into a browser. See Enabling the Developer Menu Options, on page 12.

Context Parameters

In the preceding example, `param0` is used to specify the Cisco UCS Director *context*. The context data value refers to one of the major domains managed by Cisco UCS Director, for example, `global-services`, `datacenter`, or `storage_accounts`. A list of the standard Cisco UCS Director contexts appears in API Request Context Parameters, on page 21.

Report Parameters

The report parameter value is always the *reportId*. Typical reportIds include `"STORAGE-ACCOUNTS-T51"`, `"CPU-S0"`, `"VOLUMES--X1"`, `"NETWORK-USAGE-H0"`, `"PORT-SUMMARY-V50"`, and `"PRIVATE-CLOUD-FREE-STORAGE-S1"`. The reportId is typically the last parameter listed in an API request for a report. So, if the context is specified by two parameters, the report parameter is often the third, `param2`. For an extensive list of report names and reportIds, categorized by context, see List of Available Cisco UCS Director Reports, on page 31.

Handle Special Characters in REST URL

The characters "&", "=", "+", ",", and "$" are reserved special characters by URL standards. Hence, it is better to avoid using them when data is part of the URL (like the JSON requests) instead of being sent as a payload.

In unavoidable circumstances, you can alternatively use their unicode formats:

Special Character	Unicode
&	\u0026
=	\u003D
+	\u002B
,	\u002C
$	\u0024

For example, if the data to be passed in the REST URL is *Business Operations & PMO. WSS*, the special character can be replaced with the unicode format and the data can be passed as *Business Operations \u0026 PMO. WSS*.

Few more examples:

App+App Data = App\u002BApp Data

area,street = area\u002Cstreet

50$ = 50\u0024

if x=y = if x\u003Dy

Sample API Request 1

The operation that requests a Login profile refers to the logged-in user, so there is no need for a parameter. Most other operations need multiple arguments.

```
http://10.10.1.153/app/api/rest?formatType=json&opName=userAPIGetMyLoginProfile&opData={}
```

Response to Sample API Request 1

```
{ "serviceResult":{"userId":"jsmith","firstName":"John","lastName":"Smith","email":
"jsmith@example.com","groupName":"Eng Group","role":"Regular"}, "serviceError":null,
"serviceName":"InfraMgr", "opName":"userAPIGetMyLoginProfile" }
```

Sample API Request 2

In this request for a report about a chassis for a data center, the operation requires three parameters, which are typical for a report request.

```
http://172.17.9.142/app/api/rest?opName=userAPIGetTabularReport&opData=
{param0:"datacenter",param1:"datacenter",param2:"UCS-CHASSIS-T50"}
```

Response to Sample API Request 2

```
{"serviceResult":
    {"rows":[{"ID":"PHY-ACC;sys/chassis-2",
    "Account_Name":"PHY-ACC","DN":"sys/chassis-2",
    "Serial_Number":"1558","Model":"N20-C6508","Power_State":"ok",
    "Operation_State":"accessibility-problem",
    "Configuration_State":"ok","License_State":"license-ok","Servers":5,
    "IO_Modules":2,"PSUs":4,"Fan_Modules":8,
    "Vendor":"Cisco Systems Inc"},{"ID":"UCSCirrus;sys/chassis-1",
    "Account_Name":"UCSCirrus","DN":"sys/chassis-1",
    "Serial_Number":"FOX1352GDX4","Model":"N20-C6508",
    "Power_State":"redundancy-failed","Operation_State":"power-problem",
    "Configuration_State":"ok","License_State":"license-ok","Servers":7,
    "IO_Modules":2,"PSUs":4,"Fan_Modules":8,
    "Vendor":"Cisco Systems Inc"},{"ID":"UCSM237;sys/chassis-1",
    "Account_Name":"UCSM237","DN":"sys/chassis-1",
    "Serial_Number":"1557","Model":"N20-C6508","Power_State":"ok",
    "Operation_State":"operable",
    "Configuration_State":"unsupported-connectivity","License_State":"license-ok",
    "Servers":6,"IO_Modules":2,
    "PSUs":4,"Fan_Modules":8,"Vendor":"Cisco Systems Inc"},
    {"ID":"real108;sys/chassis-1","Account_Name":"real108",
    "DN":"sys/chassis-1","Serial_Number":"FOX1352GDX4","Model":"N20-C6508",
    "Power_State":"redundancy-failed",
    "Operation_State":"power-problem","Configuration_State":"ok",
    "License_State":"license-ok","Servers":7,
    "IO_Modules":2,"PSUs":4,"Fan_Modules":8,"Vendor":"Cisco Systems Inc"}],
    "columnMetaData":null}, "serviceError":null, "serviceName":"InfraMgr",
    "opName":"userAPIGetTabularReport" }}
```

Tip   For advanced Cisco UCS Director API users, the code in the Report Metadata request for the data center UCS-CHASSIS-T50 report gave the first parameter as `param0:"23"`. In the request used in Sample API Request 2, the context "datacenter" was substituted for the value "23" and the request was successful.

API Request Context Parameters

Cisco UCS Director REST API operations that require parameters typically require the context as a parameter. There are a few exceptions, notably the operations that pick the context from login information.

If you want to create an API request, you have two options for finding the correct context parameter value to use:

- Find the correct context name string in the list of standard Cisco UCS Director contexts, and copy it into the JSON parameter specification in your API request.

- In Cisco UCS Director, navigate to a report that represents the same report data that you want to request through the API. Find the REST API URL in Report Metadata and use the context parameter(s) that you find there.

Timesaver If you find the URL code used to send a request in Cisco UCS Director, you can use some or all of that API request data to create your own request.

Note In some scenarios, you might want to use the English (human-readable) name string for the context value rather than the numeral string value that is provided in the Report Metadata listing of the REST API URL.

Context Field Names and Corresponding Parameter Names

In the listing below, the first element is the name of the field; the second, in quotation marks, is the context value that you should assign to the parameter representing the context. In most requests for reports, param0 provides the context. If another context parameter (param1) value is required, you can use the value provided for this parameter in the Report Metadata listing of the REST API URL.

Administrative Contexts

- *CONTEXT_TYPE_GLOBAL*= "global";

- *CONTEXT_TYPE_GLOBAL_ADMIN* = "global_admin";

- *CONTEXT_TYPE_GLOBAL_SERVICES* = "global-services";

- *CONTEXT_TYPE_CLOUD* = "cloud";

- *CONTEXT_TYPE_HOSTNODE* = "hostnode";

- *CONTEXT_TYPE_CLUSTER* = "cluster";

End User Contexts

- *CONTEXT_TYPE_GROUP* = "group";

- *CONTEXT_TYPE_VM* = "vm";

- *CONTEXT_TYPE_VDC* = "vdc";

- *CONTEXT_TYPE_SR* = "servicerequest";

Data Center Contexts

- *CONTEXT_TYPE_PHYSICAL_DATACENTER* = "datacenter";

NetApp Report Contexts

- *CONTEXT_TYPE_STORAGE_ACCOUNTS* = " storage_accounts";

- *CONTEXT_TYPE_STORAGE_FILERS* = " netapp_filer";

- *CONTEXT_TYPE_STORAGE_AGGREGATES* = " storage_aggregates";

- *CONTEXT_TYPE_STORAGE_VOLUMES* = "storage_volumes";

- *CONTEXT_TYPE_STORAGE_LUNS* = "luns";

- *CONTEXT_TYPE_STORAGE_VFLIERS* = "netapp_v_flier";

UCS Report Contexts

- *CONTEXT_TYPE_INFRA_COMPUTE_UCSM_ACCOUNT* = " ucsm";

- *CONTEXT_TYPE_INFRA_COMPUTE_UCS_FABRIC_INTERCONNECT* = " compute_fbi";

- *CONTEXT_TYPE_INFRA_COMPUTE_UCS_CHASSIS* = " compute_chassis";

- *CONTEXT_TYPE_INFRA_COMPUTE_UCS_SERVER* = "compute_server";

- *CONTEXT_TYPE_INFRA_COMPUTE_UCS_SERVICE_PROFILE* = "service_profile";

- *CONTEXT_TYPE_INFRA_COMPUTE_UCS_PORT_CHANNEL* = "ucs_portchannel";

- *CONTEXT_TYPE_INFRA_COMPUTE_UCS_ORGANIZATION* = "ucs_org";

- *CONTEXT_TYPE_INFRA_COMPUTE_UCS_SERVICE_PROFILE_TEMPLATE* = "ucs_service-profile-template";

- *CONTEXT_TYPE_INFRA_COMPUTE_UCS_BOOT_POLICY* = "ucs_boot_policy";

- *CONTEXT_TYPE_INFRA_COMPUTE_UCS_VNIC_TEMPLATE* = "ucs_vnictemplate";

- *CONTEXT_TYPE_INFRA_COMPUTE_UCS_MAC_POOL* = "ucs_mac";

- *CONTEXT_TYPE_INFRA_COMPUTE_UCS_UUID_POOL* = "ucs_uuid";

- *CONTEXT_TYPE_INFRA_COMPUTE_UCS_WWNN_POOL* = "ucs_wwnn";

- *CONTEXT_TYPE_INFRA_COMPUTE_UCS_WWPN_POOL* = "ucs_wwpn";

- *CONTEXT_TYPE_INFRA_COMPUTE_UCS_SERVICE_PROFILE_VHBA* = "ucs_sp_vhba";

- *CONTEXT_TYPE_INFRA_COMPUTE_UCS_SERVICE_PROFILE_VNIC* = "ucs_sp_vnic";

- *CONTEXT_TYPE_INFRA_COMPUTE_UCS_IOMODULE* = "ucs_iomodule";

- *CONTEXT_TYPE_INFRA_COMPUTE_UCS_SERVER_ADAPTER_UNIT* = "compute_server_adapter_unit";

Network Report Contexts

- *CONTEXT_TYPE_INFRA_NETWORK_DEVICE*= "network_device";

- *CONTEXT_TYPE_INFRA_NET_DEVICE_N1K* = " net_device_n1k";

- *CONTEXT_TYPE_INFRA_NET_DEVICE_FAB_IC* = " net_device_fab_ic";

- *CONTEXT_TYPE_INFRA_NET_DEVICE_N5K* = "net_device_n5k";

- *CONTEXT_TYPE_INFRA_NETWORK_DEVICE_VLAN* = "net_device_vlan";

- *CONTEXT_TYPE_INFRA_NETWORK_DEVICE_VSAN* = "net_device_vsan";

- *CONTEXT_TYPE_INFRA_NETWORK_DEVICE_INTERFACE* = "net_device_interface";

- *CONTEXT_TYPE_INFRA_NETWORK_DEVICE_PORT_PROFILE* = "net_device_port_profile";

- *CONTEXT_TYPE_INFRA_NETWORK_DEVICE_ZONE* = "net_device_zone";

- *CONTEXT_TYPE_INFRA_NET_QOS_POLICY* = "net_device_qos_policy";

How to Interpret the HTTP Response

The following HTTP status codes are returned by Cisco UCS Director:

- **401 Unauthorized**—The API key is not a valid key.

- **200 OK**—Cisco UCS Director has processed the request. The actual status of the request is in the body of the response.

The Cisco UCS Director response body is in JSON format as determined by the FormatType parameter specified in the API request.

Code and Status of an REST API Response

Code	Status
0	SUCCESS
1	UNKNOWN_RESOURCE
2	NOT_SUPPORTED
3	INTERNAL_ERROR
4	FAILED
5	ALREADY_EXISTS
6	INVALID_INPUT
7	RESOURCE_NOT_FOUND
8	INVALID_REQUEST

Components of an API Response

API Response Component	Description	Component Example (Success Scenario)
serviceResult	If the request succeeds, this result contains a set of name-value pairs or a JSON object or report.	`"serviceResult": {"userId":"jsmith", "firstName":"John","lastName":"Smith", "email":"jsmith@example.com", groupName":"Eng Group","role":"Regular"}`
serviceError	If the request succeeds, the serviceError is set to null. If the operation fails, the serviceError contains the error message.	`"serviceError":null`
serviceName	Name of the backend service. The backend service is often set to *InfraMgr*.	`"serviceName":"InfraMgr"`
opName	Name of the operation provided in the request.	`"opName":"userAPIGetMyLoginProfile"`

Example: API Response in a Success Scenario

```
{
"serviceResult":{"userId":"jsmith","firstName":"John","lastName":"Smith",
"email":"jsmith@example.
com","groupName":"Eng Group","role":"Regular"}, "serviceError":null,
"serviceName":"InfraMgr",
"opName":"userAPIGetMyLoginProfile"
}
```

Example: API Response in a Failure Scenario

```
{ "serviceResponse":null,
"serviceError":"SERVICE_CALL_EXCEPTION: Service InfraMgr does not support operation test",

"serviceName":"InfraMgr", "opName":"test"
}
```

API Response (Service Result) Data Types

The service result (payload) sent in a response to a Cisco UCS Director REST API request is specified for the operation. The service result can be an operation-specific set of name-value pairs, or it can be formatted as a standard data type for this API, that is, as a report or a JSON object.

How to Use cURL Commands

cURL is a command line tool for getting or sending data using URL syntax. You can use the cURL command to execute a REST API request.

The following sample shows how to execute the `userAPISubmitWorkflowServiceRequest` API and pass the input values:

```
curl  -v -X POST -H 'X-Cloupia-Request-Key:5CF4C115F0034B189616B2B8EBA0F220'
  -g 'http://172.17.32.75/app/api/rest?formatType=json&opName=
```

```
userAPISubmitWorkflowServiceRequest&opData={param0:"TestWorkFlowFromAPI",param1:
{"list":[{"name":"A1","value":"Hello"},{"name":"A2","value":"World"}]},param2:-1}'
```

The following sample shows how to pass the variable value with space in insecure mode:

```
curl --insecure -v -X POST -H 'X-Cloupia-Request-Key:C97B881CE6B94DFB930765F7AC408xxx'
 -g 'https://172.31.234.127/app/api/rest?formatType=json&opName=
userAPISubmitWorkflowServiceRequest&opData={param0:"AddUCSMvlanv1",param1:{"list":
[{"name":"VLAN%20ID","value":"500"},{"name":"VLAN%20DESC","value":"CURLvlantest"}]},param2:-1}'
```

Note To run the command in secure mode, install the CA certificate (cacert) file in your system and use the cert location in curl call using the —cacert curl option.

The following sample shows how to roll back a workflow:

```
curl --insecure -v -X POST -H 'X-Cloupia-Request-Key:C97B881CE6B94DFB930765F7AC408xxx'
 -g 'https://172.31.234.127/app/api/rest?formatType=json&opName=
userAPIRollbackWorkflow&opData={param0:140}'
```

Note The **-g** parameter disables the curl check for nested braces { } [ ]. By default, curl does not allow nested braces in REST calls.

Cisco UCS Director REST API SDK Bundle – Samples

When you download the SDK, you can find the sample Java code in the `ucsd-rest-api-sdk-v2/src/com/cisco/cuic/api/examples` path.

Example: Retrieving VDCs in a Group Based on a Logged-in User

When provisioning VMs, you have to view a list of VDCs available for a particular group and choose the required VDC. To accomplish this, Cisco UCS Director has to identify the group to which the user belongs to and retrieve a list of VDCs available for the group. Then, you can create the VDC on which the VM is provisioned.

Create a workflow with the following REST APIs in sequence:

1. userAPIGetMyLoginProfile—Retrieve the profile of the logged in user to identify the group to which the user belongs to.

2. userAPIGetAllVDCs—Retrieve a list of VDCs available in the user group.

3. userAPICreateVDC—Create a service request to choose a VDC on which the VM is provisioned.

Procedure

Step 1 Identify the group to which the user belongs to using the userAPIGetMyLoginProfile API.

Request:

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

Response:

```
{"serviceResult":{"userId":"sdk","firstName":null,"lastName":null,"email":"xyz@test.com",
"groupName":"Default Group","groupId":1,"role":"Regular"}, "serviceError":null,
"serviceName":"InfraMgr","opName":"userAPIGetMyLoginProfile"}
```

In this example, the user sdk belongs to the DefaultGroup group.

Step 2 Retrieve a list of VDCs in the user group using the userAPIGetAllVDCs API.

Request:

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

Response:

```
{ "serviceResult":{"rows":[{"Cloud":"","vDC_ID":1,"Group":"Default Group","vDC":"Default
vDC","Type":"Standard","Lock_State":"Locked","Total_VMs":82,"Active_VMs":72,"Custom_Categories":0,
"vDC_Description":"","Approvers":"","Status":"Default vDC: Not for new provisioning",
"Tag":null},
{"Cloud":"VCenter","vDC_ID":4,"Group":"Default Group", "vDC":"SDKcont1" ,"Type":"Container",
"Lock_State":"Unlocked","Total_VMs":2,"Active_VMs":2,"Custom_Categories":0,
"vDC_Description":"",
"Approvers":"","Status":"Error(s):\nNo NIC configuration found.",
"Tag":null}],"columnMetaData":
null,"reportParams":{}}, "serviceError":null, "serviceName":"InfraMgr",
"opName":"userAPIGetAllVDCs" }
```

Step 3 Create a service request for choosing the VDC on which the VM is provisioned using the userAPICreateVDC API. An approver approves the service request and the user is notified about the VM provisioning using this VDC. The system policy, computing policy, network policy, storage policy, and the cost model applicable to the VDC are selected. The number of days to wait before deleting an inactive VM is selected. A self-service policy which defines the tasks or actions that can be performed on the VDC is selected.

Request:

```
/app/api/rest?formatType=json&opName=userAPICreateVDC&opData={param0:{"vdcName":"sdk",
"vdcDescription":"sdk vdc", "cloudName":"VCenter","groupName":1, "approver1":"",
"approver2":"",
"vdcSupportEmail":"","vdcCustomerNoticationEmail":"","systemPolicy":"Default System Policy",

"deploymentPolicy":"","slaPolicy":"" , "computingPolicy":"VCenter - Default Computing
Policy",
"storagePolicy":"VCenter - Default Storage Policy", "networkPolicy":"VCenter - Default
Network
 Policy","costModel":"", "isLocked":false,"isDeletable":true,"inactivityPeriodForDeletion":-1,
"selfServiceEndUserPolicy":""}}
```

Response:

```
{ "serviceResult":true, "serviceError":null, "serviceName":"InfraMgr",
"opName":"userAPICreateVDC" }
```

Example: Adding a Cisco UCS Manager UUID pool

The following sample provides the XML-based API URL to add a Cisco UCS Manager UUID pool:

```
<cuicoperationRequest>
<payload>
```

```
<![CDATA[
<ucsUuidPool>
<name>Test_UUID_Pool</name>
<descr>Test_UUID_Pool</descr>
<prefix>other</prefix>
<otherPrefix>00000000-0000-0000<otherPrefix>
<accountName>ucsm-248</accountName>
<org>org-root</org>
<firstMACAddress>0000-000000000001</firstMACAddress>
<size>1</size>
</ucsUuidPool>
]]>
</payload>
```

In this example, ensure that you provide the following details:

- <prefix>—Enter either **derived** or **other**.

- <org>—The org must follow the format: org-*dn*. Where, *dn* is domain name.

Example: Setup PXE Boot Task

The following sample provides the XML-based API URL to execute setup PXE boot task:

```
<cuicOperationRequest>
<operationType>SETUP_PXE_BOOT</operationType>
<payload>
<![CDATA[
<SetupPXEBoot>
<!-- Accepts value from the list: NetworkBootManagerList-->
<netwotkBootManager>PXE</netwotkBootManager>
<osType>ESXi6.0</osType>
<serverMACAddress>dd:cc:ff:ff:cc:cc</serverMACAddress>
<!-- Accepts value from the list: IPConfigType-->
<IPConfigType>Static</IPConfigType>
<serverIPAddress>24.45.45.56</serverIPAddress>
<serverNetMask>255.255.255.0</serverNetMask>
<serverHostname>esxi</serverHostname>
<serverGateway>24.45.45.1</serverGateway>
<serverNameserver></serverNameserver>
<manageMentVlanID>0</manageMentVlanID>
<!-- Accepts value from the list: password-->
<rootPassword>cloupia123</rootPassword>
<timeZone>Africa/Abidjan</timeZone>
<additionalParameterList>
<AdditionalParameter>
<paramName>test</paramName>
<paramValue>123</paramValue>
</AdditionalParameter>

</additionalParameterList>
</SetupPXEBoot>
]]>
</payload>
</cuicOperationRequest>
```

Example: Adding a Storage Policy

The following sample provides the XML-based API URL to add a storage policy with the connectivity type as Simple:

```
<ucsStoragePolicy>
<policyName>Storage_Policy1</policyName>
<policyDescription>Test Storage Policy</policyDescription> <ucsAcctName>UCSM_70</ucsAcctName>
<orgName>org-root</orgName>
<connectivityType>Simple</connectivityType>
<wwnnPool>node-default</wwnnPool>
<noOfVhba>0</noOfVhba>
<vHBA0Name>fc0</vHBA0Name>
<vSAN>VSAN-EST</vSAN>
<vHBA1Name>fc1</vHBA1Name>
<vSAN_FI_B>VSAN-Add</vSAN_FI_B>
<sanConnPolicy></sanConnPolicy>
</ucsStoragePolicy>
```

The following sample provides the XML-based API URL to add a storage policy with the connectivity type as Use SAN Connectivity Policy:

```
<ucsStoragePolicy>
<policyName>Storage_Policy3</policyName>
<policyDescription>Test Storage Policy</policyDescription> <ucsAcctName>UCSM_70</ucsAcctName>
<orgName>org-root</orgName>
<connectivityType>Use SAN Connectivity Policy</connectivityType>
<wwnnPool>node-default</wwnnPool>
<noOfVhba>0</noOfVhba>
<sanConnPolicy>Test</sanConnPolicy>
</ucsStoragePolicy>
```

Define the connectivity type for adding a storage policy.

- connectivityType—Choose one of the following as the connectivity type: Expert, Simple, No vHBAs, Hardware Inherited, and Use SAN Connectivity Policy. Set the <noOfVhba> as zero when you chose the connectivity type as Simple or Use SAN Connectivity Policy.

Example: Fetching Details of a Specific VMware VM

The following sample provides the XML-based API URL to fetch details of a specific VMware VM:

```
Request:
/cloupia/api-v2/datacenter/{dcName}/account/{accountName}/vmwareDC/{datacenterName}/vmwareVM/{configName}
Response:
<cuicOperationResponse>
<cuicOperationStatus>0</cuicOperationStatus>
<response><vmwareVM>
<accountName>VC99</accountName>
<annotation>UCSD=XX.yy.zzz.aaa,SR=56,INST=TestOVF</annotation>
<bootOption>bios</bootOption>
<bootTime>-1</bootTime>
<committedDiskGB>0.02</committedDiskGB>
<configName>CrsDC</configName>
<coresPerSocket>1</coresPerSocket>
<cpuHotAddEnabled>false</cpuHotAddEnabled>
<cpuLimitMhz>-1</cpuLimitMhz>
<cpuOverheadLimitMhz>0</cpuOverheadLimitMhz>
```

```
<cpuReservationMhz>0</cpuReservationMhz>
<cpuShares>1000</cpuShares>
<customAttributes>prs6732_114CustAttrName: ;prs6732CustAttr: ;prs6740VMAnn: ;prsVMName:
;PrsGlobal: ;VMannotationVM: ;VMannotationVMname: ;AutoGlobal: ;WORKGROUP: </customAttributes>
<datastoreList>Automation-VMware-Setup</datastoreList>
<datastores>Automation-VMware-Setup</datastores>
<defaultPowerOffType>soft</defaultPowerOffType>
<dnsDhcp>false</dnsDhcp>
<dnsServerList></dnsServerList>
<dnsSuffixList></dnsSuffixList>
<folder>/Discovered virtual machine</folder>
<guestIPAddress></guestIPAddress>
<guestOS>Other Linux (32-bit)</guestOS>
<guestState>notRunning</guestState>
<hostNode>172.26.225.105</hostNode>
<instId>CrsDC</instId>
<instanceUuid>501998b1-ddaf-3df3-4cfd-2134de5ed0be</instanceUuid>
<memLimitMB>-1</memLimitMB>
<memOverheadLimitMB>0</memOverheadLimitMB>
<memReservationMB>0</memReservationMB>
<memShares>480</memShares>
<memoryHotAddEnabled>false</memoryHotAddEnabled>
<memoryMB>48</memoryMB>
<name>CrsDC</name>
<numCPUs>1</numCPUs>
<numSockets>1</numSockets>
<portGroups>VM Network</portGroups>
<powerState>poweredOff</powerState>
<protectedVM>false</protectedVM>
<remoteDisplayVNCEnabled>false</remoteDisplayVNCEnabled>
<remoteDisplayVNCPort>0</remoteDisplayVNCPort>
<resourcePool>Resources</resourcePool>
<resourcePoolOwner>New Cluster_Others</resourcePoolOwner>
<resourcePoolParent>New Cluster_Others</resourcePoolParent>
<toolsRunningStatus>guestToolsNotRunning</toolsRunningStatus>
<toolsVersion>2147483647</toolsVersion>
<uncommittedDiskGB>0.27</uncommittedDiskGB>
<unsharedDiskGB>0.02</unsharedDiskGB>
<uuid>421995ad-820e-8bd0-aa84-1253285f0539</uuid>
<vcenterVMId>vm-2390</vcenterVMId>
<vmDiskCount>1</vmDiskCount>
<vmIPAddr></vmIPAddr>
<vmId>1</vmId>
<vmMacAddr>00:50:56:99:55:bb</vmMacAddr>
<vmOverallStatus>gray</vmOverallStatus>
<vmPath>[Automation-VMware-Setup] CrsDC/CrsDC.vmx</vmPath>
<vmUniqueId>501998b1-ddaf-3df3-4cfd-2134de5ed0be</vmUniqueId>
<vmVersion>vmx-08</vmVersion>
<vmVnicCount>1</vmVnicCount>
<vmrcEnabled>false</vmrcEnabled>
<vmwareDatacenterName>Others</vmwareDatacenterName>
<vnicDeviceConfigId>4000</vnicDeviceConfigId>
</vmwareVM></response></cuicOperationResponse>
```

Example: Adding a Network Policy

The following sample provides the XML-based API URL to add a network policy with the connectivity type as Use LAN Connectivity Policy:

```
<cuicOperationRequest>
<payload>
<![CDATA[
```

```
<ucsNetworkPolicy>
<vNIC0Label></vNIC0Label>
<vNIC1Label></vNIC1Label>
<policyId></policyId>
<!--Mandatory field. The length of the ID must  be lesser than or equal to 16 characters.
The ID field accepts alpha-numeric value and special characters (_, -, ., ,).-->
<policyName>Net-Test</policyName>
<policyDescription></policyDescription>
<!--Mandatory field.-->
<accountName>UCSM-70</accountName>
<!--Mandatory field.-->
<orgDn>org-root/org-Org-GUReg/org-Org-UGBP2</orgDn>
<dynVNICConnectionPolicy></dynVNICConnectionPolicy>
<!--Mandatory field.-->
<connectivityType>Use LAN Connectivity Policy</connectivityType>
<lanConnPolicy>LAN_CONN_POL</lanConnPolicy>
<!--Mandatory field.-->
<expertNoOfVnics>0</expertNoOfVnics>
</ucsNetworkPolicy>
]]>
</payload>
</cuicOperationRequest>
```

Define the connectivity type for adding a network policy.

- connectivityType—Choose one of the following as the connectivity type: Expert, Simple, No vNICs, Hardware Inherited, and Use LAN Connectivity Policy. Set the <expertNoOfVnics> as zero when you chose the connectivity type as Use LAN Connectivity Policy.

Example: Modifying a LAN Port Channel

The UcsLanPortChannel (UPDATE) API enables you to perform the following actions:

- edit—To edit the flowCtrlPolicy, adminSpeed, and name.

- addports—To add a new port to an existing lanport channel.

- enableport—To enable all the ports attached to the respective lanport channel.

- disableport—To disable all the ports attached to the respective lanport channel.

The following samples provides the XML-based API URL to execute the above mentioned actions:

edit

```
<cuicOperationRequest>
<payload>
<![CDATA[
<ucsLanPortChannel>
<accountName>ucsm-70</accountName>
<adminSpeed>1gbps</adminSpeed>
<flowCtrlPolicy>Flow_Contrl</flowCtrlPolicy>
<switchId>A</switchId>
<name>PC01</name>
<portId>1</portId>
<apiAction>edit</apiAction>
</ucsLanPortChannel>
]]>
</payload>
</cuicOperationRequest>
```

addports

```
<cuicOperationRequest>
<payload>
<![CDATA[
<ucsLanPortChannel>
<accountName>ucsm-70</accountName>
<switchId>A</switchId>
<name>PC01</name>
<portId>001</portId>
<apiAction>addPorts</apiAction>
<portList>
<accountName>ucsm-70</accountName>
<slotId>1</slotId>
<portId>18</portId>
</portList>
</ucsLanPortChannel>
]]>
</payload>
</cuicOperationRequest>
```

enableport

```
<cuicOperationRequest>
<payload>
<![CDATA[
<ucsLanPortChannel>
<accountName>ucsm-70</accountName>
<switchId>A</switchId>
<name>PC01</name>
<portId>1</portId>
<apiAction>enablePort</apiAction>
</ucsLanPortChannel>
]]>
</payload>
</cuicOperationRequest>
```

disableport

```
<cuicOperationRequest>
<payload>
<![CDATA[
<ucsLanPortChannel>
<accountName>ucsm-70</accountName>
<switchId>A</switchId>
<name>PC01</name>
<portId>1</portId>
<apiAction>disablePort</apiAction>
</ucsLanPortChannel>
]]>
</payload>
</cuicOperationRequest>
```

List of Available Cisco UCS Director Reports

In the following table, reports are grouped by context. The same report (with the same reportID) may be used for different contexts, and so may appear in multiple locations in this list.

Note Report IDs are always single strings without character spaces. Because some of these strings are very long, they are broken across multiple lines within individual table cells below. If you perform an electronic search for a full, exact reportId string, it may not return a result because the break inserts a character space; search instead for a short unique string contained within the full reportId character string.

Context	Report Name	Report ID	Report Type
AWS-EC2 cloud	Summary	SUMMARY-V0	Summary
AWS-EC2 cloud	VMs	VMS-T0	Tabular
AWS-EC2 cloud	Images	IMAGES-T0	Tabular
AWS-EC2 cloud	Deleted VMs	DELETED-VMS-T0	Tabular
AWS-EC2 cloud	Memory	MEMORY-S0	Bar Chart
AWS-EC2 cloud	VPU	CPU-S0	Bar Chart
AWS-EC2 cloud	Disk	DISK-S0	Bar Chart
AWS-EC2 cloud	Trend: Memory	TREND-MEMORY-H0	Trend
AWS-EC2 cloud	Trend: CPU	TREND-CPU-H0	Trend
AWS-EC2 cloud	Trend: CPU Usage	TREND-CPU-USAGE-H0	Trend
AWS-EC2 cloud	Trend: Disk Reads & Writes (bytes)	TREND-DISK-READS-& -WRITES-(BYTES)-H0	Trend
AWS-EC2 cloud	Trend: Disk Reads & Writes (ops)	TREND-DISK-READS-& -WRITES-(OPS)-H0	Trend
AWS-EC2 cloud	Trend: Disk Reads & Writes (ops)	TREND-DISK-READS-& -WRITES-(OPS)-H0	Trend
AWS-EC2 cloud	Trend: Network Usage	TREND-NETWORK-USAGE-H0	Trend
AWS-EC2 cloud	Groups with Most CPU Usage	GROUPS-WITH-MOST -CPU-USAGE-T0	Tabular
AWS-EC2 cloud	VMs With Most Trend:CPU Usage	VMS-WITH-MOST -CPU-USAGE-T0	Tabular
AWS-EC2 cloud	VMs With Over-Utilized CPU Usage	VMS-WITH-OVER- UTILIZED-CPU-USAGE-T0	Tabular
AWS-EC2 cloud	VMs With Under-Utilized CPU Usage	VMS-WITH-UNDER -UTILIZED-CPU-USAGE-T0	Tabular
AWS-EC2 cloud	Volumes	VOLUMES-X1	Tabular with Actions

Context	Report Name	Report ID	Report Type
AWS-EC2 cloud	Snapshots	SNAPSHOTS-X1	Tabular with Actions
AWS-EC2 cloud	Summary	SUMMARY-V1	Summary
AWS-EC2 cloud	SP Status	SP-STATUS-T0	Tabular
AWS-EC2 vm	CPU Usage (percent)	CPU-USAGE-(PERCENT)-S0	Bar Chart
AWS-EC2 vm	Network In	NETWORK-IN-S0	Bar Chart
AWS-EC2 vm	Network Out	NETWORK-OUT-S0	Bar Chart
AWS-EC2 vm	Trend: CPU Usage	TREND-CPU-USAGE-H0	Trend
AWS-EC2 vm	Trend: Disk Reads & Writes (bytes)	TREND-DISK-READS -&-WRITES-(BYTES)-H0	Trend
AWS-EC2 vm	Trend: Disk Reads & Writes (ops)	TREND-DISK-READS -&-WRITES-(OPS)-H0	Trend
AWS-EC2 vm	Network Usage	NETWORK-USAGE-H0	Trend
catalog	Deployability Assessment	DEPLOYABILITY-ASSESSMENT-T45	Tabular
cloud	vDCs	VDCS-T0	Tabular
cloud	Events	EVENTS-T0	Tabular
cloud	Number of Events by Severity	NUMBER-OF-EVENTS -BY-SEVERITY-S0	Bar Chart
cloud	Groups WIth Most Number of VMs	GROUPS-WITH-MOST -NUMBER-OF-VMS-T0	Tabular
cloud	vDCs With Most Number of VMs	VDCS-WITH-MOST -NUMBER-OF-VMS-T0	Tabular
cloud	vDCs With Most CPU Usage	VDCS-WITH-MOST -CPU-USAGE-T0	Tabular
cloud	vDCs With Most Memory Usage	VDCS-WITH-MOST -MEMORY-USAGE-T0	Tabular
cloud	vDCs With Most Disk Usage	VDCS-WITH-MOST -DISK-USAGE-T0	Tabular
cloud, vm	VM Action Requests	VM-ACTION-REQUESTS-X0	Tabular with Actions
compute_chassis	Summary	SUMMARY-V50	Summary
compute_chassis	Servers	SERVERS-T50	Tabular

Context	Report Name	Report ID	Report Type
compute_chassis	Fan Modules	FAN-MODULES-T50	Tabular
compute_chassis	IO Modules	IO-MODULES-T50	Tabular
compute_chassis	Power Supply Units	POWER-SUPPLY-UNITS-T50	Tabular
compute_chassis	Events	EVENTS-T50	Tabular
compute_fbi	Summary	SUMMARY-V50	Summary
compute_fbi	Power Supply Units	POWER-SUPPLY-UNITS-T50	Tabular
compute_fbi	Fans	FANS-T50	Tabular
compute_fbi	Ethernet Ports	ETHERNET-PORTS-X50	Tabular with Actions
compute_fbi	Fibre Channel Ports	FIBRE-CHANNEL-PORTS-X50	Tabular with Actions
compute_fbi	Trend: Ethernet Ports Total Kilo Bytes Transferred/Received	TREND-ETHERNET-PORTS -TOTAL-KILO-BYTES -TRANSFERRED/RECEIVED-H50	Trend
compute_fbi	Trend: Fibre Channel Ports Total Kilo Bytes Transferred/Received	TREND-FIBRE-CHANNEL -PORTS-TOTAL-KILO-BYTES -TRANSFERRED/RECEIVED-H50	Trend
compute_fbi	Trend: CPU Utilization	TREND-CPU-UTILIZATION-H50	Trend
compute_fbi	Trend: Memory	TREND-MEMORY-H50	Trend
compute_fbi	Events	EVENTS-T50	Tabular
compute_fbi_port	Port Summary	PORT-SUMMARY-V50	Summary
compute_server	Local Disks	LOCAL-DISKS-T50	Tabular
compute_server	Memory Units	MEMORY-UNITS-T50	Tabular
compute_server	Processor Units	PROCESSOR-UNITS-T50	Tabular
compute_server	Interface Cards	INTERFACE-CARDS-X50	Tabular with Actions
compute_server	Service Request Details	SERVICE-REQUEST-DETAILS-T50	Tabular
compute_server	Trend: MotherBoard Input Current Received	TREND-MOTHERBOARD -INPUT-CURRENT-RECEIVED-H50	Trend
compute_server	Trend: MotherBoard Input Voltage Received	TREND-MOTHERBOARD -INPUT-VOLTAGE-RECEIVED-H50	Trend

Context	Report Name	Report ID	Report Type
compute_server	Trend: MotherBoard Consumed Power	TREND-MOTHERBOARD -CONSUMED-POWER-H50	Trend
compute_server	Trend: MotherBoard Sens IO Temperature	TREND-MOTHERBOARD -SENS-IO-TEMPERATURE-H50	Trend
compute_server	Trend: MotherBoard Sens Rear Temperature	TREND-MOTHERBOARD -SENS-REAR-TEMPERATURE-H50	Trend
compute_server	Trend: Memory Unit Temperature	TREND-MEMORY-UNIT -TEMPERATURE-H50	Trend
compute_server	Trend: Processor Unit Current	TREND-PROCESSOR-UNIT -CURRENT-H50	Trend
compute_server	Trend: Processor Unit Temperature	TREND-PROCESSOR-UNIT -TEMPERATURE-H50	Trend
compute_server	Summary	SUMMARY-V50	Summary
compute_server	Events	EVENTS-T50	Tabular
compute_server _adapter_unit	DCE Interfaces	DCE-INTERFACES-T50	Tabular
compute_server _adapter_unit	HBAs	HBAS-T50	Tabular
compute_server _adapter_unit	NICs	NICS-T50	Tabular
custom_actions	Add/Edit Tasks	ADD/EDIT-TASKS-X46	Tabular with Actions
datacenter	Storage Accounts	STORAGE-ACCOUNTS-T51	Tabular
datacenter	Top 5 Volume Total-Used Storage	TOP-5-VOLUME-TOTAL -USED-STORAGE-T9999	Tabular
datacenter	Top 5 Lun Total-Used Storage	TOP-5-LUN-TOTAL-USED -STORAGE-T9999	Tabular
datacenter	Summary	SUMMARY-V51	Summary
datacenter	Volumes: Total vs Used	VOLUMES-TOTAL-VS-USED-S51	Bar Chart
datacenter	LUNs: Total vs Used	LUNS-TOTAL-VS-USED-S51	Bar Chart
datacenter	Aggregates: Free vs Used	AGGREGATES-FREE-VS -USED-S51	Pie Chart
datacenter	UCSM Accounts	UCSM-ACCOUNTS-X50	Tabular with Actions

Context	Report Name	Report ID	Report Type
datacenter	Chassis	CHASSIS-T50	Tabular
datacenter	Fabric Interconnects	FABRIC-INTERCONNECTS-T9999	Tabular
datacenter	Server Pools	SERVER-POOLS-T9999	Tabular
datacenter	Summary	SUMMARY-V50	Summary
datacenter	Service Profiles	SERVICE-PROFILES-T9999	Tabular
datacenter	Processor Units	PROCESSOR-UNITS-T9999	Tabular
datacenter	Memory Units	MEMORY-UNITS-T9999	Tabular
datacenter	Local Disks	LOCAL-DISKS-T9999	Tabular
datacenter	IO Modules	IO-MODULES-T9999	Tabular
datacenter	Managed Network Elements	MANAGED-NETWORK-ELEMENTS-X52	Tabular with Actions
datacenter	VTP Status	VTP-STATUS-T52	Tabular
datacenter	Private VLANs	PRIVATE-VLANS-T52	Tabular
datacenter	L2 Neighbors	L2-NEIGHBORS-T52	Tabular
datacenter	Port Profiles	PORT-PROFILES-T52	Tabular
datacenter	VM Network Details	VM-NETWORK-DETAILS-T52	Tabular
datacenter	Host Network Details	HOST-NETWORK-DETAILS-T52	Tabular
datacenter	VSANs	VSANS-T52	Tabular
datacenter	VLANs	VLANS-X52	Tabular with Actions
datacenter	HP Accounts	HP-ACCOUNTS-X50	Tabular with Actions
datacenter	DHCP Log	DHCP-LOG-T50	Tabular
datacenter	PXE Boot Requests	PXE-BOOT-REQUESTS-X50	Tabular with Actions
global	Summary	SUMMARY-V0	Summary
global	Clouds	CLOUDS-T0	Tabular
global	vDCs	VDCS-T0	Tabular
global	Clusters	CLUSTERS-T0	Tabular

Context	Report Name	Report ID	Report Type
global	VMs	VMS-T0	Tabular
global	Host Node Status	HOST-NODE-STATUS-T0	Tabular
global	Host Node Inventory	HOST-NODE-INVENTORY-T0	Tabular
global	Resource Pools	RESOURCE-POOLS-T0	Tabular
global	Events	EVENTS-T0	Tabular
global	Images	IMAGES-T0	Tabular
global	Deleted VMs	DELETED-VMS-T0	Tabular
global	System Health	SYSTEM-HEALTH-T30	Tabular
global	Active VM Distribution	ACTIVE-VM-DISTRIBUTION-BY-CLOUD-S0	Pie Chart
global	Active VMs Public vs Private By Clouds	ACTIVE-VMS-PUBLIC-VS-PRIVATE-CLOUDS-S0	Pie Chart
global	Memory	MEMORY-S0	Bar Chart
global	CPU	CPU-S0	Bar Chart
global	Disk	DISK-S0	Bar Chart
global	Private Cloud Storage Capacity	PRIVATE-CLOUD-STORAGE-CAPACITY-S1	Pie Chart
global	Private Cloud Free Storage	PRIVATE-CLOUD-FREE-STORAGE-S1	Pie Chart
global	Private Cloud Used Storage	PRIVATE-CLOUD-USED-STORAGE-S1	Pie Chart
global	Storage Capacity Per Storage Type	STORAGE-CAPACITY-PER-STORAGE-TYPE-S1	Bar Chart
global	Used Storage Per Storage Type	USED-STORAGE-PER-STORAGE-TYPE-S1	Bar Chart
global	Free Storage Per Storage Type	FREE-STORAGE-PER-STORAGE-TYPE-S1	Bar Chart
global	Top 5 Datastores Most Used	TOP-5-DATASTORES-MOST-USED-S1	Bar Chart
global	Top 5 Datastores Least Used	TOP-5-DATASTORES-LEAST-USED-S1	Bar Chart
global	Number of Events	NUMBER-OF-EVENTS-S0	Bar Chart
global	Trend: Number of Host Nodes	TREND-NUMBER-OF-HOST-NODES-H0	Trend

Context	Report Name	Report ID	Report Type
global	Trend: Memory	TREND-MEMORY-H0	Trend
global	Trend: Storage Capacity, Used & Free	TREND-STORAGE-CAPACITY, -USED-&-FREE-H1	Trend
global	Trend: CPU	TREND-CPU-H0	Trend
global	All VMware Activity	ALL-VMWARE-ACTIVITY-T0	Tabular
global	Summary	SUMMARY-V1	Summary
global	SP Status	SP-STATUS-T0	Tabular
global, cloud	VMs Active vs Inactive	VMS-ACTIVE-VS-INACTIVE-S0	Bar Chart
global, cloud	Trend: Number of VMs	TREND-NUMBER-OF-VMS-H0	Trend
global, cloud	Trend: VM Additions & Deletions	TREND-VM-ADDITIONS-& -DELETIONS-H0	Trend
global_admin	NetAppVolumesTable	NETAPPVOLUMESTABLE-T9999	Tabular
global_admin	NetAppVFilerVolumes Table	NETAPPVFILERVOLUMES TABLE-T9999	Tabular
global_admin	NetAppLunsTable	NETAPPLUNSTABLE-T9999	Tabular
global_admin	NetAppVFilerLunsTable	NETAPPVFILERLUNSTABLE-T9999	Tabular
global_admin	NetAppFilersTable	NETAPPFILERSTABLE-T9999	Tabular
global_admin	NetAppInitiatorGroups Table	NETAPPINITIATORGROUPS TABLE-T9999	Tabular
global_admin	NetAppVFilerInitiator Groups Table	NETAPPVFILERINITIATOR GROUPSTABLE-T9999	Tabular
global_admin	NetAppIPSpacesTable	NETAPPIPSPACESTABLE-T9999	Tabular
global_admin	NetAppVFilersTable	NETAPPVFILERSTABLE-T9999	Tabular
global_admin	NetAppAggregatesTable	NETAPPAGGREGATESTABLE-T9999	Tabular
global_admin	NetAppONTAPAccounts Table	NETAPPONTAPACCOUNTS TABLE-T9999	Tabular
global_admin	NetAppDFMAccounts Table	NETAPPDFMACCOUNTS TABLE-T9999	Tabular
global_admin	NetAppDFMFiler Table	NETAPPDFMFILERTABLE-T9999	Tabular
global_admin	NetAppONTAPFiler Table	NETAPPONTAPFILERTABLE-T9999	Tabular

Context	Report Name	Report ID	Report Type
global_admin	NetAppONTAPvFilers Table	NETAPPONTAPVFILERS TABLE-T9999	Tabular
global_admin	NetAppUnAssignedIP SpacesTable	NETAPPUNASSIGNEDIPSPACES TABLE-T9999	Tabular
global_admin	NetAppInterfacesTable	NETAPPINTERFACESTABLE-T9999	Tabular
global_admin	NetAppVLANPhysical InterfacesTable	NETAPPVLANPHYSICALINTERFACES TABLE-T9999	Tabular
global_admin	NetAppInterfacesvFilers Assigned Table	NETAPPINTERFACESVFILERS ASSIGNEDTABLE-T9999	Tabular
global_admin	NetAppDFMvFilersTable	NETAPPDFMVFILERSTABLE-T9999	Tabular
global_admin	NetAppDfmVolume DatasetTable	NETAPPDFMVOLUMEDATASET TABLE-T9999	Tabular
global_admin	NetAppDfmLUN DatasetTable	NETAPPDFMLUNDATASET TABLE-T9999	Tabular
global_admin	NetAppDfmGroupTable	NETAPPDFMGROUPTABLE-T9999	Tabular
global_admin	NetAppDfmProvision PolicyTable	NETAPPDFMPROVISIONPOLICY TABLE-T9999	Tabular
global_admin	NetAppDfmStorage ServiceTable	NETAPPDFMSTORAGESERVICE TABLE-T9999	Tabular
global_admin	NetAppUnAssigned DatasetTable	NETAPPUNASSIGNEDDATASET TABLE-T9999	Tabular
global_admin	NetAppAssignedDatasetTable	NETAPPASSIGNEDDATASET TABLE-T9999	Tabular
global_admin	NetAppDatasetTable	NETAPPDATASETTABLE-T9999	Tabular
global_admin	NetAppDatasetMember LUN Table	NETAPPDATASETMEMBER LUNTABLE-T9999	Tabular
global_admin	NetAppReourcePoolTable	NETAPPREOURCEPOOLTABLE-T9999	Tabular
global_admin	NetAppGroupAssigned VFilersTable	NETAPPGROUPASSIGNED VFILERSTABLE-T9999	Tabular
global_admin	Provisioning Policies	PROVISIONING-POLICIES-T48	Tabular
global_admin	Protection Policies	PROTECTION-POLICIES-T48	Tabular
global_admin	Storage Services	STORAGE-SERVICES-T48	Tabular
global_admin	vFiler Templates	VFILER-TEMPLATES-T48	Tabular

Context	Report Name	Report ID	Report Type
global_admin	OnCommand Datasets	ONCOMMAND-DATASETS-T48	Tabular
global_admin	OnCommand Groups	ONCOMMAND-GROUPS-T48	Tabular
global_admin	Resource Pool	RESOURCE-POOL-T48	Tabular
global_admin	UIMenuItemTable	UIMENUITEMTABLE-T9999	Tabular
global_admin	UIOperationTable	UIOPERATIONTABLE-T9999	Tabular
global_admin	Customer Organizations	CUSTOMER-ORGANIZATIONS-X23	Tabular with Actions
global_admin	Customer Organizations	CUSTOMER-ORGANIZATIONS-T23	Tabular
global_admin	Customer Organizations	CUSTOMER-ORGANIZATIONS-T23	Tabular
global_admin	MSP Organizations	MSP-ORGANIZATIONS-X23	Tabular with Actions
global_admin	MSP Organizations	MSP-ORGANIZATIONS-T23	Tabular
global_admin	Login Page Branding	LOGIN-PAGE-BRANDING-X23	Tabular with Actions
global_admin	Login Page Branding	LOGIN-PAGE-BRANDING-T23	Tabular
global_admin	Login Users	LOGIN-USERS-X23	Tabular with Actions
global_admin	Login Users	LOGIN-USERS-T23	Tabular
global_admin	Current Online Users	CURRENT-ONLINE-USERS-X23	Other
global_admin	Virtual Accounts	VIRTUAL-ACCOUNTS-X22	Tabular with Actions
global_admin	Virtual Accounts	VIRTUAL-ACCOUNTS-T22	Tabular
global_admin	Physical Accounts	PHYSICAL-ACCOUNTS-X24	Tabular with Actions
global_admin	Physical Accounts	PHYSICAL-ACCOUNTS-T24	Tabular
global_admin	Catalog	CATALOG-X40	Tabular with Actions
global_admin	UcsOrganizationTable	UCSORGANIZATION TABLE-T9999	Tabular
global_admin	UcsServerTable	UCSSERVERTABLE-T9999	Tabular
global_admin	UcsServiceProfileTable	UCSSERVICEPROFILE TABLE-T9999	Tabular
global_admin	UcsServerPoolTable	UCSSERVERPOOLTABLE-T9999	Tabular

Context	Report Name	Report ID	Report Type
global_admin	UcsBootPolicyTable	UCSBOOTPOLICYTABLE-T9999	Tabular
global_admin	Data Centers	DATA-CENTERS-X24	Tabular with Actions
global_admin	Data Centers	DATA-CENTERS-T24	Tabular
global_admin	Storage Policy	STORAGE-POLICY-X47	Tabular with Actions
global_admin	Network Policy	NETWORK-POLICY-X47	Tabular with Actions
global_admin	vHBA	VHBA-X47	Tabular with Actions
global_admin	vNIC	VNIC-X47	Tabular with Actions
global_admin	Placement Policy	PLACEMENT-POLICY-X47	Tabular with Actions
global_admin	deviceZonesTable	DEVICEZONESTABLE-T9999	Tabular
global_admin	Vendor Validated Designs	VENDOR-VALIDATED-DESIGNS-X24	Tabular with Actions
global_admin	HpServerTable	HPSERVERTABLE-T9999	Tabular
global_admin	Reports Customization	REPORTS-CUSTOMIZATION-X20	Tabular with Actions
global_admin	Amazon Deployment Policy	AMAZON-DEPLOYMENT-POLICY-X41	Tabular with Actions
global_admin	Amazon Deployment Policy	AMAZON-DEPLOYMENT-POLICY-T41	Tabular
global_admin	VMware Deployment Policy	VMWARE-DEPLOYMENT-POLICY-X21	Tabular with Actions
global_admin	VMware Deployment Policy	VMWARE-DEPLOYMENT-POLICY-T21	Tabular
global_admin	Resource Allocation Policy	RESOURCE-ALLOCATION-POLICY-X21	Tabular with Actions
global_admin	Resource Allocation Policy	RESOURCE-ALLOCATION-POLICY-T21	Tabular
global_admin	Rackspace Deployment Policy	RACKSPACE-DEPLOYMENT-POLICY-X41	Tabular with Actions
global_admin	Rackspace Deployment Policy	RACKSPACE-DEPLOYMENT-POLICY-T41	Tabular

Context	Report Name	Report ID	Report Type
global_admin	KVM Deployment Policy	KVM-DEPLOYMENT-POLICY-X41	Tabular with Actions
global_admin	KVM Deployment Policy	KVM-DEPLOYMENT-POLICY-T41	Tabular
global_admin	VMware Network Policy	VMWARE-NETWORK- POLICY-X44	Tabular with Actions
global_admin	VMware Network Policy	VMWARE-NETWORK- POLICY-T44	Tabular
global_admin	Network Provisioning Policy	NETWORK-PROVISIONING-POLICY-X44	Tabular with Actions
global_admin	Network Provisioning Policy	NETWORK-PROVISIONING-POLICY-T44	Tabular
global_admin	VLAN Pool Policy	VLAN-POOL-POLICY-X44	Tabular with Actions
global_admin	VLAN Pool Policy	VLAN-POOL-POLICY-T44	Tabular
global_admin	Virtual Storage Catalog	VIRTUAL-STORAGE- CATALOG-X43	Tabular with Actions
global_admin	Virtual Storage Catalog	VIRTUAL-STORAGE- CATALOG-T43	Tabular
global_admin	VMware Storage Policy	VMWARE-STORAGE- POLICY-X43	Tabular with Actions
global_admin	VMware Storage Policy	VMWARE-STORAGE- POLICY-T43	Tabular
global_admin	VMware Storage Policy	VMWARE-SYSTEM-POLICY-X41	Tabular with Actions
global_admin	VMware Storage Policy	VMWARE-SYSTEM-POLICY-T41	Tabular
global_admin	VMware Computing Policy	VMWARE-COMPUTING-POLICY-X42	Tabular with Actions
global_admin	VMware Computing Policy	VMWARE-COMPUTING-POLICY-T42	Tabular
global_admin	KVM Computing Policy	KVM-COMPUTING-POLICY-X42	Tabular with Actions
global_admin	KVM Computing Policy	KVM-COMPUTING-POLICY-T42	Tabular
global_admin	Cost Model	COST-MODEL-X41	Tabular with Actions
global_admin	Cost Model	COST-MODEL-T41	Tabular

Context	Report Name	Report ID	Report Type
global_admin	Storage Tier Cost Model	STORAGE-TIER-COST-MODEL-X41	Tabular with Actions
global_admin	Storage Tier Cost Model	STORAGE-TIER-COST- MODEL-T41	Tabular
global_admin	OS License	OS-LICENSE-X41	Tabular with Actions
global_admin	OS License	OS-LICENSE-T41	Tabular
global_admin	Network Services Agents	NETWORK-SERVICES- AGENTS-X24	Tabular with Actions
global_admin	Discovered Devices	DISCOVERED-DEVICES-X24	Tabular with Actions
global_admin	Active Modules	ACTIVE-MODULES-X26	Tabular with Actions
global_admin	Active Modules	ACTIVE-MODULES-T26	Tabular
global_admin	Modules	MODULES-X26	Tabular with Actions
global_admin	Modules	MODULES-T26	Tabular
global_admin	Module Snapshots	MODULE-SNAPSHOTS-X26	Tabular with Actions
global_admin	Module Snapshots	MODULE-SNAPSHOTS-T26	Tabular
global_admin	vmwareVSwitchTable	VMWAREVSWITCH TABLE-T9999	Tabular
global_admin	portGroups	PORTGROUPS-T9999	Tabular
global_admin	Workflows	WORKFLOWS-X46	Tabular with Actions
global_admin	User VM Action Policy	USER-VM-ACTION-POLICY-X46	Tabular with Actions
global_admin	Context Workflow Mapping	CONTEXT-WORKFLOW- MAPPING-X46	Tabular with Actions
global_admin	Templates	TEMPLATES-X46	Tabular with Actions
global_admin	Workflow Schedules	WORKFLOW-SCHEDULES-X46	Tabular with Actions
global_admin	Custom Approval Tasks	CUSTOM-APPROVAL-TASKS-X46	Tabular with Actions

Context	Report Name	Report ID	Report Type
global_admin	License	LICENSE-X20	Tabular with Actions
global_admin	tabularlov.system.advanced. property.report	TABULARLOV.SYSTEM.ADVANCED. PROPERTY.REPORT-T9999	Tabular
global_admin	Change Records	CHANGE-RECORDS-T25	Tabular
global_admin	Application Categories	APPLICATION-CATEGORIES-X20	Tabular with Actions
global_admin	System Tasks	SYSTEM-TASKS-X20	Tabular with Actions
global_admin	System Tasks	SYSTEM-TASKS-T20	Tabular
global_admin	LDAP Integration	LDAP-INTEGRATION-X23	Tabular with Actions
global_admin	LDAP Integration	LDAP-INTEGRATION-T23	Tabular
global_admin	Virtual Console Servers	VIRTUAL-CONSOLE-SERVERS-X24	Tabular with Actions
global_admin	Triggers	TRIGGERS-X46	Tabular with Actions
global_admin, group	vDC	VDC-X45	Tabular with Actions
global_admin, group	vDC Service Profiles	VDC-SERVICE-PROFILES-X45	Tabular with Actions
global_admin, group	Catalog	CATALOG-T40	Tabular
group	vDCs	VDCS-T14	Tabular
group	Port Groups	PORT-GROUPS-X14	Tabular with Actions
group	vFilers	VFILERS-X15	Tabular with Actions
group	Servers	SERVERS-X15	Tabular with Actions
group	Service Profiles	SERVICE-PROFILES-X15	Tabular with Actions
group	Service Requests	SERVICE-REQUESTS-X10	Tabular with Actions

Context	Report Name	Report ID	Report Type
group	Archived Service Requests	ARCHIVED-SERVICE-REQUESTS-X10	Tabular with Actions
group	Users	USERS-X13	Tabular with Actions
group	VMs	VMS-T14	Tabular
group	Resource Limits	RESOURCE-LIMITS-T13	Tabular
group	Summary	SUMMARY-V13	Summary
group	Top 5 Failure Reasons	TOP-5-FAILURE-REASONS-T10	Tabular
group	Trend: Network Usage	TREND-NETWORK-USAGE-H14	Trend
group	Trend: CPU Usage	TREND-CPU-USAGE-H14	Trend
group	Trend: Disk Usage	TREND-DISK-USAGE-H14	Trend
group	Trend: Consolidated Resource Usage	TREND-CONSOLIDATED-RESOURCE-USAGE-H14	Trend
group	VMs Active vs Inactive	VMS-ACTIVE-VS-INACTIVE-S13	Bar Chart
group	VLANs	VLANS-X15	Tabular with Actions
group	Resource Accounting	RESOURCE-ACCOUNTING-T12	Tabular
group	Resource Accounting Details	RESOURCE-ACCOUNTING-DETAILS-T12	Tabular
group	Chargeback	CHARGEBACK-T12	Tabular
group	Resource Accounting Details	RESOURCE-ACCOUNTING-DETAILS-T85	Tabular
group	Chargeback	CHARGEBACK-T85	Tabular
group	Current Month Summary	CURRENT-MONTH-SUMMARY-V12	Summary
group	Previous Month Summary	PREVIOUS-MONTH-SUMMARY-V12	Summary
group	Current Month Cost Summary	CURRENT-MONTH-COST-SUMMARY-S12	Pie Chart
group	Previous Month Cost Summary	PREVIOUS-MONTH-COST-SUMMARY-S12	Pie Chart
group	Current Month Top 5 Applications	CURRENT-MONTH-TOP-5-APPLICATIONS-S12	Pie Chart

Context	Report Name	Report ID	Report Type
group	Previous Month Top 5 Applications	PREVIOUS-MONTH-TOP-5 -APPLICATIONS-S12	Pie Chart
group	Trend: Budget Spending	TREND-BUDGET-SPENDING-H12	Trend
group	Trend: Total Cost	TREND-TOTAL-COST-H12	Trend
group	Trend: VM Cost	TREND-VM-COST-H12	Trend
group	Trend: CPU Cost	TREND-CPU-COST-H12	Trend
group	Trend: Memory Cost	TREND-MEMORY-COST-H12	Trend
group	Trend: Network Cost	TREND-NETWORK-COST-H12	Trend
group	Payment Information	PAYMENT-INFORMATION-X10	Tabular with Actions
group	Customer Funds	CUSTOMER-FUNDS-X10	Tabular with Actions
group	Resource Accounting Details	RESOURCE-ACCOUNTING- DETAILS-T10	Tabular
hostnode	Summary	SUMMARY-V0	Summary
hostnode	VMs	VMS-T0	Tabular
hostnode	Events	EVENTS-T0	Tabular
hostnode	Deleted VMs	DELETED-VMS-T0	Tabular
hp, datacenter	ILO Servers	ILO-SERVERS-X50	Tabular with Actions
hp_server	Server NICs	SERVER-NICS-T50	Tabular
hp_server	Server Memory	SERVER-MEMORY-T50	Tabular
hp_server	Server Processor	SERVER-PROCESSOR-T50	Tabular
hp_server	Server Slots	SERVER-SLOTS-T50	Tabular
HyperV cloud	Trend: Number of Host Nodes	TREND-NUMBER-OF- HOST -NODES-H0	Trend
HyperV cloud	Trend: Memory	TREND-MEMORY-H0	Trend
HyperV cloud	Trend: CPU	TREND-CPU-H0	Trend
HyperV cloud	Summary	SUMMARY-V0	Summary
HyperV cloud	Clusters	CLUSTERS-T0	Tabular

Context	Report Name	Report ID	Report Type
HyperV cloud	Host Node Status	HOST-NODE-STATUS-T0	Tabular
HyperV cloud	Host Node Inventory	HOST-NODE-INVENTORY-T0	Tabular
HyperV cloud	VMs	VMS-T0	Tabular
HyperV cloud	Deleted VMs	DELETED-VMS-T0	Tabular
HyperV cloud	Data Stores	DATA-STORES-T0	Tabular
HyperV cloud	Images	IMAGES-T9999	Tabular
HyperV cloud	Memory	MEMORY-S0	Bar Chart
HyperV cloud	CPU	CPU-S0	Bar Chart
HyperV cloud	Disk	DISK-S0	Bar Chart
HyperV cluster	Summary	SUMMARY-V0	Summary
HyperV cluster	Host Node Status	HOST-NODE-STATUS-T0	Tabular
HyperV cluster	Host Node Inventory	HOST-NODE-INVENTORY-T0	Tabular
HyperV cluster	VMs	VMS-T0	Tabular
HyperV cluster	Events	EVENTS-T0	Tabular
HyperV hostnode	CPU Usage	CPU-USAGE-S0	Bar Chart
HyperV hostnode	CPU Usage(Mhz)	CPU-USAGE(MHZ)-S0	Bar Chart
HyperV hostnode	Memory Usage	MEMORY-USAGE-S0	Bar Chart