

Active System Manager Release 8.2 API Reference Guide



Notes, cautions, and warnings

 **NOTE:** A NOTE indicates important information that helps you make better use of your computer.

 **CAUTION:** A CAUTION indicates either potential damage to hardware or loss of data and tells you how to avoid the problem.

 **WARNING:** A WARNING indicates a potential for property damage, personal injury, or death.

© 2016 Dell Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. Dell and the Dell logo are trademarks of Dell Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

2016 - 04

Rev. A00

Contents

1 Overview.....	10
Other Documents You May Need.....	10
2 Resource URI Endpoints.....	12
3 Authentication Headers.....	13
4 Generic Query Parameters.....	15
Dates in query strings.....	15
5 HTTP Message Bodies.....	16
Validations.....	16
6 Usage Examples.....	18
Timezone.....	18
Get the current timezone.....	18
Set the current timezone.....	19
Credential.....	19
Get all defined Credentials.....	19
Define new Credential.....	19
DiscoveryRequest.....	20
Discover Chassis and contained Blade Servers.....	20
ManagedDevice.....	21
Get all Managed Devices.....	21
ServiceTemplate.....	21
Get all ServiceTemplates.....	21
Get a ServiceTemplate by name.....	22
Publish a ServiceTemplate.....	22
Deployment.....	23
Get all deployed Services.....	23
Deploy a new Service.....	23
Tear down a Service.....	24
Firmware.....	24
Check Firmware compliance of all Resources.....	24
7 Exception Handling.....	26
8 Appendix A - API Reference.....	27

/Authenticate/.....	27
/Chassis/.....	28
/Chassis/{refId}.....	29
/Chassis/{refId}.....	29
/Chassis/{refId}.....	30
/Chassis/iom/ip/{ip}.....	31
/Chassis/{refId}/sensorlog.....	32
/Chassis/{refId}/sensorlog.....	32
/Chassis/{serviceTag}/{type}.....	33
/Chassis/{serviceTag}/{type}.....	34
/Credential/.....	35
/Credential/.....	36
/Credential/{id}.....	37
/Credential/{id}.....	37
/Credential/{id}.....	38
/DHCP/.....	39
/DHCP/.....	39
/Deployment/.....	40
/Deployment/.....	41
/Deployment/{id}.....	42
/Deployment/{id}.....	43
/Deployment/{id}.....	44
/Deployment/export/csv.....	44
/Deployment/defineService.....	45
/Deployment/defineServiceDiff.....	46
/Deployment/device/{deviceId}.....	46
/Deployment/filter/{numOfDeployments}.....	47
/Deployment/migrate/{serviceId}/{serverId}.....	48
/Deployment/migrateServer/{serviceId}/{componentId}.....	49
/Deployment/serverNetworking/{serviceId}/{serverId}.....	49
/DeviceGroup/.....	50
/DeviceGroup/.....	51
/DeviceGroup/{refId}.....	52
/DeviceGroup/{refId}.....	53
/DeviceGroup/{refId}.....	54
/DiscoveryRequest/.....	54
/DiscoveryRequest/.....	55
/DiscoveryRequest/{id}.....	56
/DiscoveryRequest/{id}.....	57
/DiscoveryRequest/chassislist.....	57
/DiscoveryRequest/discoveryresult/{id}.....	58
/FirmwareRepository/.....	59

/FirmwareRepository/.....	60
/FirmwareRepository/{id}.....	60
/FirmwareRepository/{id}.....	61
/FirmwareRepository/connection.....	62
/FirmwareRepository/compliance.....	63
/FirmwareRepository/compliance/{id}.....	63
/FirmwareRepository/softwarecomponent.....	64
/FirmwareRepository/softwarebundle/{id}.....	65
/ManagedDevice/.....	66
/ManagedDevice/.....	68
/ManagedDevice/count.....	68
/ManagedDevice/{refId}.....	69
/ManagedDevice/{refId}.....	70
/ManagedDevice/{refId}.....	71
/ManagedDevice/firmware.....	71
/ManagedDevice/export/csv.....	72
/ManagedDevice/withcompliance.....	73
/ManagedDevice/puppet/{certName}.....	73
/ManagedDevice/puppet/{certName}.....	74
/ManagedDevice/withcompliance/{refId}.....	75
/NTP/.....	75
/NTP/.....	76
/NTP/.....	77
/Network/.....	77
/Network/.....	78
/Network/export/csv.....	79
/Network/{networkId}.....	80
/Network/{networkId}.....	81
/Network/{networkId}.....	81
/Network/ipAddress/assign.....	82
/Network/ipAddress/release.....	83
/Network/{networkId}/usageids.....	84
/Network/{networkId}/ipaddresses.....	84
/OSRepository/.....	85
/OSRepository/.....	86
/OSRepository/{id}.....	86
/OSRepository/{id}.....	87
/OSRepository/{id}.....	88
/OSRepository/{id}.....	88
/OSRepository/sync/{id}.....	89
/OSRepository/connection.....	89
/Proxy/.....	90
/Proxy/.....	90

/Proxy/test.....	91
/Server/.....	92
/Server/count.....	93
/Server/{refId}.....	94
/Server/{refId}.....	95
/Server/serviceTag/{serviceTag}.....	96
/ServiceTemplate/.....	97
/ServiceTemplate/.....	98
/ServiceTemplate/{id}.....	99
/ServiceTemplate/{id}.....	99
/ServiceTemplate/{id}.....	100
/ServiceTemplate/upload.....	101
/ServiceTemplate/export.....	102
/ServiceTemplate/{id}/copy.....	103
/ServiceTemplate/export/csv.....	103
/ServiceTemplate/uploadConfig.....	104
/ServiceTemplate/updateParameters.....	105
/ServiceTemplate/device/{deviceId}.....	105
/ServiceTemplate/{id}/mapToPhysicalResources.....	106
/ServiceTemplate/components/service/{serviceId}.....	107
/ServiceTemplate/components/template/{templateId}.....	108
/ServiceTemplate/service/{serviceId}/{componentType}.....	108
/ServiceTemplate/template/{templateId}/{componentType}.....	109
/Timezone/.....	110
/Timezone/.....	111
/Timezone/all.....	111
/User/.....	112
/User/.....	113
/User/{userId}.....	114
/User/{userId}.....	115
/User/{userId}.....	116
/WizardStatus/.....	116
/WizardStatus/.....	117

9 Appendix B - Model Reference..... 118

BrownfieldStatus.....	118
ComparatorValue	118
CompliantState	118
DeploymentHealthStatusType	119
DeploymentStatusType	119
DeviceHealth	119
DeviceState	119

DeviceType	120
DiscoverDeviceType	120
DiscoveryStatus	120
DiskMediaType	120
Health	121
HotSpareStatus	121
IOMLocationType	121
IdentityType	121
LogSeverity	121
MediaType	122
NetworkMode	122
PhysicalType	122
RepositoryState	122
RepositoryStatus	122
ServerSlotType	123
ServiceTemplateComponentSubType	123
ServiceTemplateComponentType	123
ServiceTemplateSettingType	123
UIRaidLevel	123
UpdateType	124
AsmDetailedMessage	124
AuthenticateRequest	124
AuthenticateResponse	125
ChassisController	125
ChassisPowerSupply	125
Enclosure	126
FirmwareInventory	126
GroupUser	126
IKVM	127
IpAddress	127
IpRange	128
Link	128
List	128
MemoryInventory	128
OSRepository	129
ProcessorInventory	129
RejectedServer	130
SelectedNIC	130
ServerJobStatus	130
ServiceTemplateOption	131
ServiceTemplateUploadRequest	131
SoftwareComponent	132

timeZone	132
UsageIdList	133
VM	133
connected_switch	133
dhcpSettings	133
fc_interface	134
logEntry	134
ntpSettings	135
proxySettings	135
references	135
testProxyResponse	135
timeZone	136
AsmDetailedMessageList	136
DeploymentDevice	136
DeviceDiscoveryRequest	137
FirmwareComplianceComponents	137
FirmwareUpdateRequest	138
GregorianCalendar	138
GroupUserList	139
IOM	139
LogicalNetworkIdentityInventory	139
PhysicalDisk	140
Server	140
ServiceTemplateValid	140
SoftwareBundle	141
StaticNetworkConfiguration	141
VirtualDiskConfiguration	142
availableTimeZones	142
credential	142
wizardStatus	143
Chassis	143
ChassisLogEntry	145
DeviceDiscoveryRequests	145
LogicalNetworkInterface	145
Network	146
User	146
VirtualDisk	147
asmCredential	147
Controller	147
Partition	148
RAIDConfiguration	148
credentialList	149

Interface	149
SelectedServer	149
DeploymentFilterResponse	150
Fabric	150
NetworkConfiguration	151
ServerNetworkObjects	151
ServiceTemplateSetting	151
ServiceTemplateCategory	152
component	153
FirmwareRepository	154
ServiceTemplate	154
Deployment	155
DiscoveredDevices	157
DiscoveryResult	158
FirmwareDeviceInventory	159
DiscoveryRequest	160
ManagedDevice	160
ManagedDeviceList	161
DeviceGroup	161
DeviceGroupList	162
10 Appendix C - Ruby Code Usage.....	163

Overview

This document is intended for a technical audience. It is assumed that the reader is familiar with REST, and programmatic interaction with REST APIs. In reality, any programming languages can be used with these APIs, however, the code examples contained in this guide are written in Ruby. XML is also used extensively for these examples.

ASM is intended to automate the processes involved in provisioning, configuring, and managing bare metal hardware: chassis, servers, switches, and network storage devices. ASM supports processes such as automated discovery of hardware, automated installation of operating systems, creation of virtual networks, and firmware management.

Two essential concepts in ASM are, *ServiceTemplate* and *Deployment process*. A *ServiceTemplate* describes a blueprint for provisioning and configuring a collection of devices as a named unit, also known as a *Service*. Deployment is the process of utilizing a *ServiceTemplate* to physically realize what service the *ServiceTemplate* describes. While the names *Deployment* and *Service* are synonymous, ASM-UI uses the term *Service* while the REST API uses the term *Deployment*.

ASM is distributed as a virtual appliance running a collection of interacting web applications that communicate through the REST interfaces. An additional web application, ASM-UI, also residing on the appliance, provides full browser-based GUI access to ASM. This ASM-UI interacts with ASM through the same REST interfaces described in this guide.

It is possible to effectively integrate ASM with any custom UI or access it through programs for better automation experiences.

In this API guide, we have described how to use these APIs with appropriate examples.

A full reference to all of the available REST methods is explained in [Appendix A](#).

 **NOTE:** Most of these REST methods exchange complex data models within the HTTP requests and responses, which can be represented as either XML or JSON depending on the preference. The structure of these models is contained in [Appendix B](#) for further reference.

Other Documents You May Need

Go to <http://www.dell.com/asmdocs> for additional supporting documents such as:

- *Active System Manager Release 8.2 User's Guide*
- *Active System Manager Release 8.2 Installation Guide*
- *Active System Manager Release 8.2 Compatibility Matrix*
- *Active System Manager Release 8.2 Release Notes*
- *Active System Manager Release 8.2 SDK Reference Guide*

- *Active System Manager Integration for VMware vRealize Orchestrator User's Guide*

For more information about ASM, including how-to videos, white papers, and blogs, see the Active System Manager page on Dell TechCenter:

<http://www.dell.com/asmtechcenter>

Resource URI Endpoints

An example of resource endpoint is: **https://Api/V1/ServiceTemplate**

From this example, API endpoints are accessible over https (at port 443).

All URI paths are prefixed with /Api/V1/ where V1 is the current API version.

Following the prefix above the remainder of the URI path (endpoint) will have the form /<resource>/<action>?<query>.

Where action and query are optional, and resource is one of the following resources:

- Authenticate
- Chassis
- Credential
- Deployment
- DeviceGroup
- DHCP
- DiscoveryRequest
- FirmwareRepository
- ManagedDevice
- Network
- NTP
- Proxy
- Server
- ServiceTemplate
- Timezone
- User
- WizardStatus

Each of these resources is fully detailed in Appendix A, showing the applicable HTTP methods for each, optional actions and query parameters, and relevant payloads for POST and PUT methods.

Authentication Headers

All API requests require 3 custom HTTP headers for authentication.

- X-dell-auth-key
- X-dell-auth-signature
- X-dell-auth-timestamp

In order to compute the authentication headers, first a REST call must be made to the /Authenticate endpoint. The Full URI is **/Api/V1/Authenticate**.

This call must be a POST request supplying either an XML or JSON payload. For illustration, the XML payload will follow the structure below providing the user name and password:

```
<AuthenticateRequest>
  <userName>admin</userName>
  <domain>ASMLLOCAL</domain>
  <password>abc123</password>
</AuthenticateRequest>
```

This call must be a POST request supplying either an XML or JSON payload. For illustration, the XML payload will follow the structure below providing the user name and password:

Depending on the Accept header, the response will contain either an XML or JSON payload. For illustration we show the XML version:

```
<AuthenticateResponse>
  <userName>admin</userName>
  <domain>ASMLLOCAL</domain>
  <role>Administrator</role>
  <apiKey>34b3577f7c3c03174a9a506b</apiKey>
  <apiSecret>9a6d9692ba64142e6a1934f9be994f3b0ae63959a6132c8b</apiSecret>
</AuthenticateResponse>
```

The returned apiKey and apiSecret from the XML response above will then be used to generate the security headers. The method is to concatenate 5 values and then compute a digest using the apiSecret from above. For illustration, a Ruby implementation of the generated headers is shown. This relies on the base64 and openssl gems to compute the signature and is extracted from the supplied Ruby ASM module discussed further in [Appendix C](#).

```
apiKey = '34b3577f7c3c03174a9a506b'
apiSecret = '9a6d9692ba64142e6a1934f9be994f3b0ae63959a6132c8b'
timestamp = Time.now.to_i.to_s

# Concatenate the following values
requestString = "%s:%s:%s:%s:%s" %
[apiKey,httpMethod,uriPath,userAgent,timestamp]

# Compute a digest on concatenated string using apiSecret
hash_str = OpenSSL::HMAC.digest('sha256',apiSecret,requestString)
signature = Base64.strict_encode64(hash_str)
```

```
headers['x-dell-auth-key'] = apiKey
headers['x-dell-auth-signature'] = signature
headers['x-dell-auth-timestamp'] = timestamp
```



NOTE: Due to the timestamp component, authentication of future requests may fail if the 2 systems communicating with each other have system clocks that are not well synchronized. So, this is essential before anything else, to synchronize the system clocks of 2 different systems.

Generic Query Parameters

Many GET methods which return a list of objects support generic query parameters for filtering, sorting, and pagination. These methods are indicated in the API reference section but the detailed syntax is described here.

- **filter**=<operator>,<valid filter column>,<value>,<value2>...
- operator is one of (eq, co)
- where valid filter columns are listed for each supporting method in the API reference
- Examples:
 - "eq,columnName,columnValue"
To include resources where columnName is columnValue.
 - "eq,columnName,columnValue,columnValue2[,columnName3...]"
To include resources where columnName is columnValue OR columnValue2.
 - "eq,columnName,"
To include resources where columnName is empty.
 - "eq,columnName,,columnValue[,columnName2...]"
To include resources where columnName is empty OR columnValue.
 - "co,columnName,columnValue"

To include resources where columnName *contains* columnValue:

- **sort**=[-]<valid sort column>, where valid sort columns are listed for each supporting method in the API reference and a minus sign preceding the column name indicates descending order
- **offset**= <integer> default is 0
- **limit**= <integer> default is 50

Dates in query strings

Some of the columns that are sortable or filterable are dates such as *createdDate*. When using the filter parameter with date strings, only the *eq* operator is valid. The format for these date strings is:

dd-MM-yyyy

example: *30-07-2015*

HTTP Message Bodies

All HTTP Requests and Responses support both XML and JSON content types. The examples used throughout this document show HTTP body content in XML form. The represented data models accepted and returned by the API calls can be quite complex. The structure of these data models is detailed in Appendix B in a language independent representation. While it is possible to craft say an XML request body from this reference documentation, a much more convenient method is to GET an existing object from ASM in the desired representation (XML or JSON) and save it to a file for use as a skeleton for future requests involving that model type.

Since a fresh system will not have any existing Deployments. A Deployment model is shown below for illustration. It is actually only a skeleton because it is lacking the essential ServiceTemplate content that would make it useful. For more information see the Usage Example called *Deploy a Service*.

```
<?xml version="1.0" encoding="UTF-8"?>
  <Deployment>
    <deploymentName>Ottomatic</deploymentName>
    <numberOfDeployments>1</numberOfDeployments>
    <deploymentDescription>test REST deployment</deploymentDescription>
    <allUsersAllowed>>false</allUsersAllowed>
    <canCancel>>false</canCancel>
    <canDelete>>false</canDelete>
    <canDeleteResources>>false</canDeleteResources>
    <canEdit>>false</canEdit>
    <canMigrate>>false</canMigrate>
    <canRetry>>false</canRetry>
    <canScaleupApplication>>false</canScaleupApplication>
    <canScaleupCluster>>false</canScaleupCluster>
    <canScaleupServer>>false</canScaleupServer>
    <canScaleupStorage>>false</canScaleupStorage>
    <canScaleupVM>>false</canScaleupVM>
    <updateServerFirmware>>false</updateServerFirmware>
    <individualTeardown>>false</individualTeardown>
    <teardown>>false</teardown>
    <retry>>false</retry>
    <serviceTemplate>
    </serviceTemplate>
  </Deployment>
```

Validations

Message bodies are subject to a wide range of validations. Capturing every validation rule is currently beyond the scope of this document but a few general rules are worth noting.

- In general validation failures generate exceptions and structured messages to isolate the violation. See the section on Exception Handling for more details. Most validation exceptions return a 400 HTTP response, but some API methods generate specific HTTP response codes to indicate particular exceptions. These are listed in the API reference in the Appendix.

- Uniqueness validations. Many models require unique names. Notable models include ServiceTemplate, Deployment, Network, User. This violation returns a 409 HTTP response in these cases.
- Relationship constraints. In the case of deletion, an exception will occur when trying to delete an entity which is being used by another entity.
- In a ServiceTemplate containing a Server component, if a hostname is supplied in addition to a hostname_template (hostname generator), this will not generate a validation exception and the behavior is to use the hostname_template over the supplied hostname.

Usage Examples

The following examples use a simple Ruby module called ASM which utilizes the publicly available RestClient gem to perform the HTTP requests. This module contains utilities to generate the security headers so we can focus on the API calls themselves. See Appendix C for detailed information on using and configuring the ASM module.

In addition to the API requests, there is sometimes significant XML processing required to prepare the request bodies or parse the responses. A utility module ASM::Payload is provided for simple get/set operations on XML document elements and simple load/save operations on XML files for use as starting templates and intermediate data storage between API calls. This module is not required but is useful for simplifying the example code that follows.

Timezone

Get the current timezone

```
require 'ASMConfig'

# Get timezone settings
url = ASM::API::URI("/Timezone")

response = ASM::API::sign {
  RestClient.get url
}
# Save it for future use as a payload
payload = ASM::Payload.from_xml(response)
payload.save_xml('timezone.xml')
```

Here we utilize a block construct to sign our request. All RestClient invocations inside the sign block will automatically be signed. The url is specified simply as '/Timezone' because ASM has already been configured with the base path prefix /Asm/V1. The response object as XML is:

```
<timeZone>
  <timeZone>(UTC-06:00) Central Time (US & Canada)</timeZone>
  <timeZoneId>11</timeZoneId>
</timeZone>
```

which we saved in the file 'timezone.xml' as a payload used in the next example to set the timezone.

Set the current timezone

Here we utilize the simple XML utility class called ASM::Payload to do simple manipulations of our XML.

```
require 'ASMConfig'

payload = ASM::Payload.load_xml('timezone.xml')

payload.set('timeZoneId','15')

# Set Timezone
response = ASM::API::sign {
  url = ASM::API::URI("/Timezone")
  RestClient.put url, payload.to_xml, :content_type => :xml
}
```

Credential

Credential objects can be created and referenced in ASM. These named credentials are used to gain access to the various Resources during the Discovery process. Credentials are also typed according to their target resource such as, Server, Storage, VCenter, IOM, Chassis, and SCVMM.

Get all defined Credentials

Default credentials are defined for several hardware types. These can be listed for the purpose of obtaining their reference id's needed for the Discovery process and to utilize as XML skeleton.

```
require 'ASMConfig'

url = ASM::API::URI('/Credential')
begin
  response = ASM::API::sign {
    RestClient.get url, :content_type => :xml
  }
  payload = ASM::Payload.from_xml(response)
  payload.save_xml('credentials.xml')
rescue RestClient::Exception => e
  print "Got exception with status: %d\n" % e.response.code
  print "%s\n" % e.response
end
```

Define new Credential

Define a new Server Credential using a skeleton file previously saved to a file.

```
require 'ASMConfig'

url = ASM::API::URI('/Credential')

cred = ASM::Payload::load_xml('credential_skeleton.xml')

# Set Credential parameters
cred.set('label', 'ServerX')
cred.set('username', 'admin')
cred.set('password', 'abc123')
cred.set('protocol', '')
cred.set('snmpCommunityString', '')
```

```

response = ASM::API::sign {
  RestClient.post url, cred.to_xml, :content_type => :xml
}

payload = ASM::Payload.from_xml(response)
credId = payload.get('id')

# Save it for future use as a payload
payload.save_xml("credential_%s.xml" % credId)

```

DiscoveryRequest

A DiscoveryRequest is a set of credential references and a set of IP ranges. A sweep of the given IP ranges will attempt to determine if Resources of the given types, specified by the credential types referenced in the request, exist at those IP addresses.

Discover Chassis and contained Blade Servers

Using the predefined chassis and server credentials, initiate a discovery request against an IP address that is known to belong to a Chassis.

```

require 'ASMConfig'

url = ASM::API::URI('/DiscoveryRequest')

# Get credential ids
creds = ASM::Payload::load_xml('credentials.xml')
server_cred = creds.doc.xpath('//serverCredential').select{|c|
c.at_xpath('label').text == 'Dell PowerEdge Default'}.first.xpath('id').text
chassis_cred = creds.doc.xpath('//chassisCredential').select{|c|
c.at_xpath('label').text == 'Dell chassis default'}.first.xpath('id').text

disc = ASM::Payload::load_xml('discovery_skeleton.xml')
disc.set('deviceStartIp', '172.17.4.17')
disc.set('deviceEndIp', '172.17.4.17')
disc.set('unmanaged', 'false')

disc.appendChild('DiscoverIPRangeDeviceRequest',
  'deviceChassisCredRef' => chassis_cred,
  'deviceServerCredRef' => server_cred,
  'bDiscoverDevicesInChassis' => 'true')

begin
  response = ASM::API::sign {
    RestClient.post url, disc.to_xml, :content_type => :xml
  }

  payload = ASM::Payload.from_xml(response)

  # Get the job status
  jobStatus = payload.get('DiscoveryRequest/status')
  print "<status>%s</status>\n" % jobStatus

rescue RestClient::Exception => e
  print "Got exception with status: %d\n" % e.response.code
  print "%s\n" % e.response
end

```

ManagedDevice

ManagedDevice is a generic interface to any Resources on the network which is being managed by ASM. A successful discovery request results in the creation of instances of ManagedDevice.

Get all Managed Devices

Assuming the Discovery process succeeded, ASM will now know about some or all of the devices available on the network.

```
require 'ASMConfig'

url = ASM::API::URI('/ManagedDevice')

response = ASM::API::sign {
  RestClient.get url
}

doc = Nokogiri::XML(response)

# Print selected info for each device
print "%30s%15s%20s%20s%15s\n" % %w(ServiceTag State IP Model Type)
print
"-----\n"
-----\n"

doc.xpath('//ManagedDevice').sort_by{|d| d.at_xpath('deviceType').content}.each
do |device|

  svc_tag = device.at_xpath('serviceTag').content
  state = device.at_xpath('state').content
  ip_addr = device.at_xpath('ipAddress').content
  model = device.at_xpath('model').content
  dev_type = device.at_xpath('deviceType').content

  print "%30s%15s%20s%20s%15s\n" % [svc_tag, state, ip_addr, model, dev_type]

end
```

ServiceTemplate

A ServiceTemplate is a reusable structure that defines the network topology and components of a future service deployment. A trivial example template could be a single server component setup to run Linux. Deploying this template would result in a single server being chosen from the available discovered resources and installing Linux on it.

Get all ServiceTemplates

Here we retrieve all ServiceTemplates and output the names of those not in draft mode (published). The draft property determines if a template is in the published state. Only published templates can be deployed.

```
require 'ASMConfig'

url = ASM::API::URI('/ServiceTemplate')
```

```

result = ASM::API::sign {
  RestClient.get url
}
# Do some XML processing to find all published templates
doc = Nokogiri::XML(result)
templates = doc.xpath('//ServiceTemplate')
published = templates.select{|t| t.at_xpath('draft').content == 'false'}
template_names = published.collect{|t| t.at_xpath('templateName').content}

print "Published Templates:\n"
template_names.each{|name| print "%s\n" % name}

```

Get a ServiceTemplate by name

Here we retrieve an existing ServiceTemplate from ASM named 'TwoServers' by passing the 'filter' query parameter. See the detail for ServiceTemplate in Appendix B for the full list of names that can be sorted and filtered on. Note that even if the template name is unique this method will always return a container around the result(s). For example, in the case of XML the root element is <ServiceTemplates>.

```

require 'ASMConfig'

url = ASM::API::URI('/ServiceTemplate')

template_name = 'TwoServers'
template_filter = "eq,name,#{template_name}"

response = ASM::API::sign {
  RestClient.get url, {:params=>{:filter=>template_filter}}
}

```

The response object is not shown for this example because it is too verbose. However, within the ServiceTemplate XML are parameters designated with the element <requiredAtDeploy>. If this value is true, then this parameter will be required when this ServiceTemplate is deployed. Otherwise the parameter is optional. For example, the server host name is required in this extracted snippet of the ServiceTemplate. Save the result above to a file inspect the full XML for a ServiceTemplate.

```

<parameters>
  <id>os_host_name</id>
  <value/>
  <type>STRING</type>
  <displayName>Host Name</displayName>
  <required>true</required>
  <requiredAtDeployment>true</requiredAtDeployment>
  <hideFromTemplate>true</hideFromTemplate>
  <min>0</min>
  <max>0</max>
  <dependencyTarget>generate_host_name</dependencyTarget>
  <dependencyValue>>false</dependencyValue>
</parameters>

```

Publish a ServiceTemplate

In order to use a ServiceTemplate to deploy a Service, the ServiceTemplate must be first *published*. This is accomplished by setting the ServiceTemplate element <draft> to false. i.e. it is no longer in the draft state.

```

require 'ASMConfig'

templateId = "ff8080814be03f17014bea250f100b7d"
url = ASM::API::URI("/ServiceTemplate/%s"%templateId)

```

```

response = ASM::API::sign {
  RestClient.get url
}
# Set draft to false
payload = ASM::Payload.from_xml(response)
payload.set('/draft', 'false')

# Send it back as a PUT
response = ASM::API::sign {
  RestClient.put url, payload.to_xml, :content_type => :xml
}

```

Deployment

While the ASM UI refers to a Service, the REST API resource is not Service, but rather Deployment. The two terms are synonymous. Deployments require an instance of a ServiceTemplate with its required parameters supplied.

Get all deployed Services

```

require 'ASMConfig'

response = ASM::API::sign {
  url = ASM::API::URI("/Deployment")
  RestClient.get url
}

doc = Nokogiri::parse(response)

doc.xpath('//Deployment').each do |deploy|
  id = deploy.at_xpath('id').content
  name = deploy.at_xpath('deploymentName').content
  status = deploy.at_xpath('jobStatus').content
  print "%s : %s : %s\n" % [id,name,status]
end

```

After getting all Services we print the id, name, and status of each. Services that are still in the act of being deployed will have a status of IN_PROGRESS. So calling this method with the Deployment Id as a path parameter can also be used to periodically poll a Deployment to find out when it has finished.

Deploy a new Service

Here, after having previously gotten an existing ServiceTemplate by name, and saved it to a file, we then set the required parameters in the ServiceTemplate file, and finally embed the ServiceTemplate into a prefabricated Deployment structure that is read from a file. As mentioned previously we can monitor the status of this Deployment by performing a GET with the deployment id as a path parameter.

```

require 'ASMConfig'

url = ASM::API::URI('/Deployment')

template_name = 'TwoServers'
template = ASM::Payload::load_xml("%s.xml"%template_name)

deployment = ASM::Payload::load_xml('deployment_skeleton.xml')

# Set Deployment parameters
deployment.set("deploymentName", 'ManualAPIDeploymentDeux')
deployment.set("deploymentDescription", 'Manual API Deployment')

```

```

deployment.set("numberOfDeployments", '1')

# Inject the template into the deployment
deployment.replaceInner('serviceTemplate',
template.doc.at_xpath('ServiceTemplate'))
deployment.save_xml('deploy_payload.xml')

response = ASM::API::sign {
  RestClient.post url, deployment.to_xml, :content_type => :xml
}
doc = Nokogiri::parse(response)
deploymentId = doc.xpath('//Deployment/id').text

print "<id>%s</id>\n" % deploymentId

```

Tear down a Service

In order to completely tear down a Service we set the teardown element to true for all components in the Deployment plus the Deployment itself, and then we PUT this Deployment back to ASM.

```

require 'ASMConfig'

deploymentId = "ff8080814aba74ce014ac455e9b80951"
url = ASM::API::URI("/Deployment/%s"%deploymentId)

response = ASM::API::sign {
  RestClient.get url
}
payload = ASM::Payload.from_xml(response)
payload.set('Deployment/teardown', 'true')
payload.set_all('components/teardown', 'true')
payload.save_xml('teardown.xml')

response = ASM::API::sign {
  RestClient.put url, payload.to_xml, :content_type => :xml
}
doc = Nokogiri::parse(response)
deploymentId = doc.xpath('//Deployment/id').text
jobStatus = doc.xpath('//Deployment/jobStatus').text

print "<jobStatus>%s</jobStatus>\n" % jobStatus

```

Firmware

Firmware versions reported by individual resource components are compared against a firmware repository to determine if component firmware is up-to-date with the latest required versions.

Check Firmware compliance of all Resources

Here we check the firmware compliance state of all resources. Note this is done by the POST method and supplying a ManagedDevice as the request body. The response is the same ManagedDevice with the addition of a compliance field.

```

require 'ASMConfig'

# Get all managed devices
url = ASM::API::URI('/ManagedDevice')
response = ASM::API::sign {
  RestClient.get url
}

```



```

}
doc = Nokogiri::XML(response)
doc.xpath('//ManagedDevice').each do |device|

  url = ASM::API::URI('/FirmwareRepository/compliance')

  begin
    response = ASM::API::sign {
      RestClient.post url, device.to_xml, :content_type => :xml
    }

    doc2 = Nokogiri::XML(response)

    serviceTag = doc2.at_xpath('//serviceTag').content
    compliance = doc2.at_xpath('//compliance').content

    print "Device %s firmware is %s\n" % [serviceTag,compliance]

  rescue RestClient::Exception => e
    print "Got exception with status: %d\n" % e.response.code
    print "%s\n" % e.response
  end
end
end

```

Exception Handling

When an HTTP error code is returned the response payload will contain information about the error that occurred. For the purpose of illustration the following exception response was intentionally generated by trying to get a ServiceTemplate with an id that doesn't exist. In this case the HTTP response was 404 and the XML response is below. See the API reference section for the relevant HTTP responses that may be returned by each REST endpoint.

The example code used to catch this exception and report the error is below.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<AsmDetailedMessageList>
  <Messages>
    <messages>
      <messageBundle>AsmManagerMessages</messageBundle>
      <messageCode>ASM0042</messageCode>
      <severity>ERROR</severity>
      <category>USER_FACING</category>
      <displayMessage>Unable to find the Template Id: ABC.</displayMessage>
      <responseAction>Make sure that a correct Template Id is entered and retry
the operation. Template Ids are case sensitive. Verify the list of devices
under the Summary tab to make sure whether or not the Template already exists.</
responseAction>
      <detailedMessage>The Template Id cannot be found.</detailedMessage>
      <agentId>ASM Manager</agentId>
      <timeStamp>2015-03-10T14:34:24.866Z</timeStamp>
      <sequenceNumber>0</sequenceNumber>
    </messages>
  </Messages>
</AsmDetailedMessageList>

require 'ASMConfig'

templateId = "ABC" // bogus id
url = ASM::API::URI("/ServiceTemplate/%s"%templateId)

begin
  response = ASM::API::sign {
    RestClient.get url
  }
rescue RestClient::Exception => e
  print "Got exception with status: %d\n" % e.response.code
  print "%s\n" % e.response
end
```

Appendix A - API Reference

/Authenticate/

Description

Confirm User access credentials

Method

post

Response Class

AuthenticateResponse

Response Content-Type: application/json, application/xml

Parameters

Table 1. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	(required)	Description of user to be authenticated	N/A	body	AuthenticateRequest

Response Status Codes

Table 2. Response Status Codes

HTTP Status Code	Reason
201	Authentication successful
401	User not found, domain not found or password does not match

/Chassis/

Description

Retrieve a list a Chassis Devices

Method

get

Response Class

[Chassis]

Response Content-Type: application/xml, application/json

Parameters

Table 3. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
sort	N/A	Valid sort columns: health,managementIP,serviceTag,credentialRefId	N/A	query	string
filter	N/A	Valid filter columns: health,managementIP,serviceTag,credentialRefId	N/A	query	array
offset	N/A	Specify pagination offset	0	query	integer
limit	N/A	Specify page limit	50	query	integer

Response Status Codes

Table 4. Response Status Codes

HTTP Status Code	Reason
200	Retrieved
400	Bad Request, verify input parameters are correct

HTTP Status Code	Reason
401	No login information specified in the request
403	User doesn't have privileges to access this operation
500	Server Internal Error, contact your system administrator

/Chassis/{refId}

Description

Retrieve an individual Chassis Device

Method

get

Response Class

Chassis

Response Content-Type: application/xml, application/json

Parameters

Table 5. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
refId	(required)	Device Id	N/A	path	string

Response Status Codes

Table 6. Response Status Codes

HTTP Status Code	Reason
200	retrieved
400	Bad Request, verify Device id is correct
404	Bad Request, verify Device id is correct

/Chassis/{refId}

Description

Update The Chasis configuration

Method

put

Response Class

Chassis

Response Content-Type: application/xml, application/json

Parameters

Table 7. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	(required)	Chassis to update	N/A	body	Chassis
refId	(required)	Id of Chassis to update	N/A	path	string

Response Status Codes

Table 8. Response Status Codes

HTTP Status Code	Reason
204	Update completed successfully
400	Problem with the update data, check error response for details
404	Chasis not found

/Chassis/{refId}

Description

Deletes an existing Chassis Device

Method

delete

Response Class

void

Response Content-Type: application/xml, application/json

Parameters

Table 9. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
refId	(required)	Device Id	N/A	path	string

Response Status Codes

Table 10. Response Status Codes

HTTP Status Code	Reason
204	deleted
400	Bad Request, verify Device id is correct.

/Chassis/iom/ip/{ip}

Description

Retrieve an individual IOM Device

Method

get

Response Class

IOM

Response Content-Type: application/xml, application/json

Parameters

Table 11. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
ip	(required)	Device IP	N/A	path	string

Response Status Codes

Table 12. Response Status Codes

HTTP Status Code	Reason
200	retrieved
400	Bad Request, verify Device IP is correct
404	Bad Request, verify Device IP is correct

/Chassis/{refId}/sensorlog

Description

Retrieve an individual Chassis Device Sensor Event Logs

Method

get

Response Class

[ChassisLogEntry]

Response Content-Type: application/xml, application/json

Parameters

Table 13. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
refId	(required)	Device Id	N/A	path	string

Response Status Codes

Table 14. Response Status Codes

HTTP Status Code	Reason
200	retrieved
400	Bad Request, verify Device id is correct
404	Bad Request, verify Device id is correct

/Chassis/{refId}/sensorlog

Description

Clear an individual Chassis Device Sensor Event Logs

Method

delete

Response Class

void

Response Content-Type: application/xml, application/json

Parameters

Table 15. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
refId	(required)	Device Id	N/A	path	string

Response Status Codes

Table 16. Response Status Codes

HTTP Status Code	Reason
204	updated
400	Bad Request, verify Device object is correct
404	Bad Request, verify Device Id is correct

/Chassis/{serviceTag}/{type}

Description

Retrieve a Chassis Device associated to a specific Server or IOM

Method

get

Response Class

Chassis

Response Content-Type: application/xml, application/json

Parameters

Table 17. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
serviceTag	(required)	Service Tag	N/A	path	string
type	(required)	Type	N/A	path	string

Response Status Codes

Table 18. Response Status Codes

HTTP Status Code	Reason
200	retrieved
400	Bad Request, verify Service tag and Type are correct
404	Bad Request, verify Service tag and Type are correct

/Chassis/{serviceTag}/{type}

Description

Deletes an existing server or IOA in chassis

Method

delete

Response Class

void

Response Content-Type: application/xml, application/json

Parameters

Table 19. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
serviceTag	(required)	Service Tag	N/A	path	string
type	(required)	Type	N/A	path	string

Response Status Codes

Table 20. Response Status Codes

HTTP Status Code	Reason
204	deleted
400	Bad Request, verify Device id is correct.

/Credential/

Description

Retrieve a list of all ASM credentials

Method

get

Response Class

credentialList

Response Content-Type: application/xml, application/json

Parameters

Table 21. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
type	N/A	filter on credential type	N/A	query	com.dell.asm.encryptedmgr.client.CredentialType
sort	N/A	Specify sort columns in a comma separated list of column names to sort. Default order is ascending. Column name can be prefixed with a minus sign to indicate descending for that column.	N/A	query	string
filter	N/A	Specify filter criteria, Example co,label,default	N/A	query	array
offset	N/A	Specify pagination offset	0	query	integer

Parameter	Value	Description	Default Value	Parameter Type	Data Type
limit	N/A	Specify page limit, can not exceed the system maximum limit.	50	query	integer

Response Status Codes

Table 22. Response Status Codes

HTTP Status Code	Reason
200	OK

/Credential/

Description

Add a new ASM credential

Method

post

Response Class

asmCredential

Response Content-Type: application/xml, application/json

Parameters

Table 23. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	(required)	ASM credential to create	N/A	body	asmCredential

Response Status Codes

Table 24. Response Status Codes

HTTP Status Code	Reason
201	Credential created
400	Bad Request, verify credential data object is correct

/Credential/{id}

Description

Find an ASM credential by type and id

Method

get

Response Class

asmCredential

Response Content-Type: application/xml, application/json

Parameters

Table 25. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
id	(required)	Id of credential to retrieve	N/A	path	string

Response Status Codes

Table 26. Response Status Codes

HTTP Status Code	Reason
200	OK
404	Credential not found

/Credential/{id}

Description

Updates an existing ASM credential

Method

put

Response Class

asmCredential

Response Content-Type: application/xml, application/json

Parameters

Table 27. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	(required)	ASM credential to update	N/A	body	asmCredential
id	(required)	Id of credential to update	N/A	path	string

Response Status Codes

Table 28. Response Status Codes

HTTP Status Code	Reason
200	Update completed successfully
400	Bad Request, verify credential data object is correct
404	Credential to be updated was not found.

/Credential/{id}

Description

Deletes an existing ASM credential

Method

delete

Response Class

void

Response Content-Type: application/xml, application/json

Parameters

Table 29. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
id	(required)	Id of credential to delete	N/A	path	string

Response Status Codes

Table 30. Response Status Codes

HTTP Status Code	Reason
204	Delete completed successfully
400	Bad Request, verify credential type and id is correct

/DHCP/

Description

Retrieve DHCP settings from ASM appliance

Method

get

Response Class

dhcpSettings

Response Content-Type: application/json, application/xml

Response Status Codes

Table 31. Response Status Codes

HTTP Status Code	Reason
200	Retrieved DHCP settings from ASM appliance successfully.
500	Unable to get DHCP settings from the ASM appliance.

/DHCP/

Description

Apply DHCP settings on ASM appliance

Method

put

Response Class

void

Response Content-Type: application/json, application/xml

Parameters

Table 32. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	(required)	DHCPEntity object that needs to be set on ASM appliance	N/A	body	dhcpSettings

Response Status Codes

Table 33. Response Status Codes

HTTP Status Code	Reason
204	Successfully changed the DHCP settings.
500	Unable to change DHCP settings on the ASM appliance.

/Deployment/

Description

Retrieve all Deployments with filter, sort, paginate which returns Array of Deployment.class

Method

get

Response Class

[Deployment]

Response Content-Type: application/xml, application/json

Parameters

Table 34. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
sort	N/A	Valid sort columns: name,createdBy,createdDate,updatedBy,updateDate	N/A	query	string

Parameter	Value	Description	Default Value	Parameter Type	Data Type
filter	N/A	Valid filter columns: name, createdBy, createdDate, updatedBy, updatedDate, expirationDate, deploymentDesc, marshalledTemplateData, health	N/A	query	array
offset	N/A	Pagination Offset	0	query	integer
limit	N/A	Page Limit	50	query	integer
full	N/A	Use full templates including resources in response	false	query	boolean

Response Status Codes

Table 35. Response Status Codes

HTTP Status Code	Reason
200	All Deployment Retrieved on filter, sort, paginate

/Deployment/

Description

Create new Deployment

Method

post

Response Class

Deployment

Response Content-Type: application/xml, application/json

Parameters

Table 36. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	N/A	Deployment to be created	N/A	body	Deployment

Response Status Codes

Table 37. Response Status Codes

HTTP Status Code	Reason
201	Deployment created
400	Invalid Parameters to create Deployment or problems with the Resource Adapters
409	Deployment already exists

/Deployment/{id}

Description

Retrieve Deployment based on deployment ID

Method

get

Response Class

Deployment

Response Content-Type: application/xml, application/json

Parameters

Table 38. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
id	(required)	Deployment Id (String)	N/A	path	string

Response Status Codes

Table 39. Response Status Codes

HTTP Status Code	Reason
200	Deployment retrived
404	Deployment not found

/Deployment/{id}

Description

Update Deployment

Method

put

Response Class

Deployment

Response Content-Type: application/xml, application/json

Parameters

Table 40. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	N/A	Deployment to be updated to	N/A	body	Deployment
id	(required)	Deployment Id (String)	N/A	path	string

Response Status Codes

Table 41. Response Status Codes

HTTP Status Code	Reason
204	Deployment updated
400	Invalid Parameters to Update Deployment or problem with the Resource Adapters
404	Deployment not found

/Deployment/{id}

Description

Delete Deployment -- this operation is idempotent

Method

delete

Response Class

void

Response Content-Type: application/xml, application/json

Parameters

Table 42. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
id	(required)	Deployment Id (String)	N/A	path	string

Response Status Codes

Table 43. Response Status Codes

HTTP Status Code	Reason
204	Deployment deleted
500	Unable to delete Deployment.

/Deployment/export/csv

Description

Exports all Deployments in csv format

Method

get

Response Class

void

Response Content-Type: application/octet-stream

Response Status Codes

Table 44. Response Status Codes

HTTP Status Code	Reason
401	No login information specified in the request
403	User doesn't have privileges to access this operation
500	User Log Internal Error, contact your system administrator

/Deployment/defineService

Description

Defines a brownfield service that may be a Deployment

Method

post

Response Class

Deployment

Response Content-Type: application/xml, application/json

Parameters

Table 45. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	(required)	Service to define	N/A	body	com.dell.asm.asmcore.asmmanager.client.deployment.ServiceDefinition

Response Status Codes

Table 46. Response Status Codes

HTTP Status Code	Reason
201	Brownfield service defined.
400	Invalid parameters to define a brownfield service that may be used for a brownfield deployment.

/Deployment/defineServiceDiff

Description

Identifies the differences from the existing service and defines an updated version of the service that that may used as a deployment

Method

post

Response Class

Deployment

Response Content-Type: application/xml, application/json

Parameters

Table 47. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
serviceId	N/A		N/A	query	string

Response Status Codes

Table 48. Response Status Codes

HTTP Status Code	Reason
201	ServiceTemplate Defined
400	Invalid service. Either service does not exist or it is not a brownfield service.

/Deployment/device/{deviceId}

Description

Retrieve all Deployments for device

Method

get

Response Class

[Deployment]

Response Content-Type: application/xml, application/json

Parameters

Table 49. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
deviceId	(required)		N/A	path	string

Response Status Codes

Table 50. Response Status Codes

HTTP Status Code	Reason
200	Deployments retrived
404	Deployments not found

/Deployment/filter/{numOfDeployments}

Description

Find available servers for template components

Method

post

Response Class

DeploymentFilterResponse

Response Content-Type: application/xml, application/json

Parameters

Table 51. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	N/A	Service template	N/A	body	ServiceTemplate
numOfDeployments	N/A	Number of deployments	N/A	path	integer
unique	N/A	If true (the default), only assign a server to one component per deployment.	true	query	boolean

Parameter	Value	Description	Default Value	Parameter Type	Data Type
		Otherwise the same server may be assigned to multiple components.			

Response Status Codes

Table 52. Response Status Codes

HTTP Status Code	Reason
200	DeploymentFilterResponse retrieved
400	Invalid Parameters to getAvailableServers
500	Unable to find available servers

/Deployment/migrate/{serviceld}/{serverId}

Description

Migrate Deployment based on deployment ID

Method

put

Response Class

Deployment

Response Content-Type: application/xml, application/json

Parameters

Table 53. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	N/A	Target server pool ID	N/A	body	string
serviceld	(required)	Deployment ID	N/A	path	string
serverId	(required)	Template server component ID	N/A	path	string

Response Status Codes

Table 54. Response Status Codes

HTTP Status Code	Reason
200	Deployment migrated
404	Deployment not found

/Deployment/migrateServer/{serviceld}/{componentId}

Description

Mark server as failed, find another available to retry deployment ID

Method

put

Response Class

Deployment

Response Content-Type: application/xml, application/json

Parameters

Table 55. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
serviceld	(required)		N/A	path	string
componentId	(required)		N/A	path	string

Response Status Codes

Table 56. Response Status Codes

HTTP Status Code	Reason
200	Server found
404	Server not found

/Deployment/serverNetworking/{serviceld}/{serverId}

Description

Returns the networks that a server in a deployment is using.

Method

get

Response Class

ServerNetworkObjects

Response Content-Type: application/xml, application/json

Parameters

Table 57. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
serviceId	(required)		N/A	path	string
serverId	(required)		N/A	path	string

Response Status Codes

Table 58. Response Status Codes

HTTP Status Code	Reason
401	No login information specified in the request
403	User doesn't have privileges to access this operation
500	Internal Error, contact your system administrator

/DeviceGroup/

Description

Retrieve all Device Group from Inventory with filter, sort, paginate, return Device Group which contains list of Managed Devices and User if exist

Method

get

Response Class

[DeviceGroup]

Response Content-Type: application/xml, application/json

Parameters

Table 59. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
sort	N/A	Valid sort columns: name,description,createdBy,createdAtDate,updatedAt,updatedDate	N/A	query	string
filter	N/A	Valid filter columns: name,description,users,devices,createdBy,updatedAt,createdAtDate,updatedDate	N/A	query	array
offset	N/A	Pagination Offset	0	query	integer
limit	N/A	Page Limit	50	query	integer

Response Status Codes

Table 60. Response Status Codes

HTTP Status Code	Reason
200	Device Group retrieved from inventory on filter, sort, paginate

/DeviceGroup/

Description

Create Device Group in Inventory

Method

post

Response Class

DeviceGroup

Response Content-Type: application/xml, application/json

Parameters

Table 61. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	N/A		N/A	body	DeviceGroup

Response Status Codes

Table 62. Response Status Codes

HTTP Status Code	Reason
201	Device Group created in inventory
400	Invalid Device Group object to create in inventory, first error encountered error causes an error response to return. Call will not return a list of errors.
409	Group already exists in inventory, first error encountered error causes an error response to return. Call will not return a list of errors.

/DeviceGroup/{refId}

Description

Retrieve Device Group from Inventory based on refId

Method

get

Response Class

DeviceGroup

Response Content-Type: application/xml, application/json

Parameters

Table 63. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
refId	(required)	Reference Id	N/A	path	string

Response Status Codes

Table 64. Response Status Codes

HTTP Status Code	Reason
200	Device Group retrieved from inventory
404	Device Group not found in inventory

/DeviceGroup/{refId}

Description

Update Device Group in Inventory

Method

put

Response Class

DeviceGroup

Response Content-Type: application/xml, application/json

Parameters

Table 65. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	N/A	Device Group to be updated	N/A	body	DeviceGroup
refId	(required)	Reference Id	N/A	path	string

Response Status Codes

Table 66. Response Status Codes

HTTP Status Code	Reason
201	Device Group updated in inventory
400	Invalid Device group id
404	Device Group not found in inventory

/DeviceGroup/{refId}

Description

Delete Device Group from Inventory

Method

delete

Response Class

Response

Response Content-Type: application/xml, application/json

Parameters

Table 67. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
refId	(required)	Reference Id	N/A	path	string

Response Status Codes

Table 68. Response Status Codes

HTTP Status Code	Reason
204	Device Group deleted from inventory
400	Unable to delete Device Group from inventory

/DiscoveryRequest/

Description

Retrieve a Array of DiscoveryRequest

Method

get

Response Class

[DiscoveryRequest]

Response Content-Type: application/xml, application/json

Parameters

Table 69. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
sort	N/A	Valid sort columns: id,status,status Message	N/A	query	string
filter	N/A	Valid filter columns: id,status,status Message	N/A	query	array
offset	N/A	pagination offset	N/A	query	integer
limit	N/A	page limit	N/A	query	integer

Response Status Codes

Table 70. Response Status Codes

HTTP Status Code	Reason
400	Problem with a query parameter, check response for details

/DiscoveryRequest/

Description

discover devices of Ip range

Method

post

Response Class

DiscoveryRequest

Response Content-Type: application/xml, application/json

Parameters

Table 71. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	N/A		N/A	body	DiscoveryRequest

Response Status Codes

Table 72. Response Status Codes

HTTP Status Code	Reason
202	Create the discovery resource
400	Bad Request, verify Discovery request object is correct

/DiscoveryRequest/{id}

Description

Retrieve Device from Inventory based on Id

Method

get

Response Class

DiscoveryRequest

Response Content-Type: application/xml, application/json

Parameters

Table 73. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
id	(required)	Discovery ID	N/A	path	string

Response Status Codes

Table 74. Response Status Codes

HTTP Status Code	Reason
200	Discovery result retrieved for the job
400	Bad Request, verify sort, filter, and pagination are valid
404	Job not found in the discovery

/DiscoveryRequest/{id}

Description

Delete Device from Discover result -- this operation is idempotent

Method

delete

Response Class

void

Response Content-Type: application/xml, application/json

Parameters

Table 75. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
id	(required)	Discovery ID	N/A	path	string

Response Status Codes

Table 76. Response Status Codes

HTTP Status Code	Reason
204	Job deleted from inventory

/DiscoveryRequest/chassislist

Description

discover devices of Ip range

Method

post

Response Class

DiscoveryRequest

Response Content-Type: application/xml, application/json

Parameters

Table 77. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	N/A		N/A	body	DiscoveryRequest

Response Status Codes

Table 78. Response Status Codes

HTTP Status Code	Reason
202	Create the discovery resource
400	Bad Request, verify Discovery request object is correct

/DiscoveryRequest/discoveryresult/{id}

Description

Retrieve a Array of DiscoveryResults

Method

get

Response Class

[DiscoveryResult]

Response Content-Type: application/xml, application/json

Parameters

Table 79. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
id	(required)	ref_id	N/A	path	string

Response Status Codes

Table 80. Response Status Codes

HTTP Status Code	Reason
400	Problem with a query parameter, check response for details

/FirmwareRepository/

Description

Retrieve all FirmwareRepository with filter, sort, paginate which returns Array of firmwarerepository.class

Method

get

Response Class

[FirmwareRepository]

Response Content-Type: application/xml, application/json

Parameters

Table 81. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
sort	N/A	Sort Column	N/A	query	string
filter	N/A	Filter Criteria	N/A	query	array
offset	N/A	Pagination Offset	0	query	integer
limit	N/A	Page Limit	50	query	integer
related	N/A	Hydrate related objects	N/A	query	boolean

Response Status Codes

Table 82. Response Status Codes

HTTP Status Code	Reason
200	All FirmwareRepository Retrived on filter, sort, paginate

/FirmwareRepository/

Description

Add a new

Method

post

Response Class

FirmwareInventory

Response Content-Type: application/xml, application/json

Parameters

Table 83. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	(required)	ASM firmware repository to create	N/A	body	FirmwareRepository

Response Status Codes

Table 84. Response Status Codes

HTTP Status Code	Reason
201	Firmware repository created
400	Bad Request, verify firmware repository data object is correct

/FirmwareRepository/{id}

Description

Retrieve an individual Chassis Device

Method

get

Response Class

FirmwareRepository

Response Content-Type: application/xml, application/json

Parameters

Table 85. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
id	(required)	Repo Id	N/A	path	string
related	N/A	Hydrate related objects	N/A	query	boolean

Response Status Codes

Table 86. Response Status Codes

HTTP Status Code	Reason
200	retrieved
400	Bad Request, verify id is correct
404	Bad Request, verify id is correct

/FirmwareRepository/{id}

Description

Updates an existing ASM FirmwareRepository

Method

put

Response Class

FirmwareRepository

Response Content-Type: application/xml, application/json

Parameters

Table 87. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	(required)	ASM FirmwareRepository to update	N/A	body	FirmwareRepository
id	(required)	Id of FirmwareRepository to update	N/A	path	string

Response Status Codes

Table 88. Response Status Codes

HTTP Status Code	Reason
200	Update completed successfully
400	Bad Request, verify credential data object is correct
404	Credential to be updated was not found.

/FirmwareRepository/connection

Description

Tests the connection to a remote path

Method

post

Response Class

FirmwareRepository

Response Content-Type: application/xml, application/json

Parameters

Table 89. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	N/A		N/A	body	FirmwareRepository

Response Status Codes

Table 90. Response Status Codes

HTTP Status Code	Reason
201	Connection to remote path successful

/FirmwareRepository/compliance

Description

Get the compliance level for the specified device

Method

post

Response Class

ManagedDevice

Response Content-Type: application/xml, application/json

Parameters

Table 91. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	(required)	Managed device to get compliance for	N/A	body	ManagedDevice

Response Status Codes

Table 92. Response Status Codes

HTTP Status Code	Reason
200	The compliance level

/FirmwareRepository/compliance/{id}

Description

Get the compliance level for the specified device

Method

post

Response Class

ManagedDevice

Response Content-Type: application/xml, application/json

Parameters

Table 93. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	(required)	Managed device to get compliance for	N/A	body	ManagedDevice
id	(required)	Id of the firmware to compare against instead of default	N/A	path	string

Response Status Codes

Table 94. Response Status Codes

HTTP Status Code	Reason
200	The compliance level

/FirmwareRepository/softwarecomponent

Description

Retrieve all FirmwareRepository with filter, sort, paginate which returns Array of firmwarerepository.class

Method

get

Response Class

[FirmwareRepository]

Response Content-Type: application/xml, application/json

Parameters

Table 95. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
vendorId	N/A	Vendor ID	N/A	query	string
deviceId	N/A	Device ID	N/A	query	string
subDeviceId	N/A	Sub Device ID	N/A	query	string
componentId	N/A	Component ID	N/A	query	string
subVendorId	N/A	Sub Vendor ID	N/A	query	string
systemId	N/A	System ID	N/A	query	string
type	N/A	Type	N/A	query	string

Response Status Codes

Table 96. Response Status Codes

HTTP Status Code	Reason
200	All FirmwareRepository Retrived on filter, sort, paginate

/FirmwareRepository/softwarebundle/{id}

Description

Retrieve an individual Software Bundle

Method

get

Response Class

SoftwareBundle

Response Content-Type: application/xml, application/json

Parameters

Table 97. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
id	(required)	Id	N/A	path	string

Response Status Codes

Table 98. Response Status Codes

HTTP Status Code	Reason
200	retrieved
400	Bad Request, verify id is correct
404	Bad Request, verify id is correct

/ManagedDevice/

Description

Retrieve all Devices from Inventory with filter.

Method

get

Response Class

[ManagedDevice]

Response Content-Type: application/xml, application/json

Parameters

Table 99. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
sort	N/A	Valid sort columns: displayName,serviceTag,refId,health,refType,deviceType,ipAddress,state,model,statusMessage,createdDate,createdBy,updatedDate,updatedBy,healthMessage,compliant,infraTemplateDate,infraTemplateId,serverTemplateDate,serverTem	N/A	query	string

Parameter	Value	Description	Default Value	Parameter Type	Data Type
filter	N/A	Valid filter columns: platelId,inventoryDate,complianceCheckDate,discoveredDate,identityRef,vendor	N/A	query	array
offset	N/A	Pagination Offset	0	query	integer
limit	N/A	Page Limit	50	query	integer

Response Status Codes

Table 100. Response Status Codes

HTTP Status Code	Reason
200	Device(s) retrieved from inventory on filter, sort, paginate

/ManagedDevice/

Description

Create Device in Inventory, return array of Managed Devices created

Method

post

Response Class

[ManagedDevice]

Response Content-Type: application/xml, application/json

Parameters

Table 101. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	N/A		N/A	body	[Lcom.dell.asm.asmcore.asmmanager.client.deviceinventory.ManagedDevice;

Response Status Codes

Table 102. Response Status Codes

HTTP Status Code	Reason
201	Devices created in inventory
400	Invalid Device to create in inventory, first error encountered error causes an error response to return. Call will not return a list of errors.
409	Device already exists in inventory, first error encountered error causes an error response to return. Call will not return a list of errors.

/ManagedDevice/count

Description

Retrieve Device total count

Method

get

Response Class

ManagedDevice

Response Content-Type: application/xml, application/json

Parameters

Table 103. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
filter	N/A	Filter Criteria	N/A	query	array

Response Status Codes

Table 104. Response Status Codes

HTTP Status Code	Reason
200	Device Count retrieved from inventory

/ManagedDevice/{refId}

Description

Retrieve Device from Inventory based on refId

Method

get

Response Class

ManagedDevice

Response Content-Type: application/xml, application/json

Parameters

Table 105. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
refId	(required)	Reference Id	N/A	path	string

Response Status Codes

Table 106. Response Status Codes

HTTP Status Code	Reason
200	Device retrieved from inventory
404	Device not found in inventory

/ManagedDevice/{refId}

Description

Update Device in Inventory

Method

put

Response Class

void

Response Content-Type: application/xml, application/json

Parameters

Table 107. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	N/A	Device to be updated to	N/A	body	ManagedDevice
refId	(required)	Reference Id	N/A	path	string

Response Status Codes

Table 108. Response Status Codes

HTTP Status Code	Reason
201	Device updated in inventory
400	Invalid Device
404	Device not found in inventory

/ManagedDevice/{refId}

Description

Delete Device from Inventory -- this operation is idempotent

Method

delete

Response Class

ManagedDevice

Response Content-Type: application/xml, application/json

Parameters

Table 109. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
refId	(required)	Reference Id	N/A	path	string
forceDelete	N/A	Force Delete	false	query	boolean

Response Status Codes

Table 110. Response Status Codes

HTTP Status Code	Reason
204	Device deleted from inventory
400	Unable to delete from RA's inventory

/ManagedDevice/firmware

Description

Update Device Firmware

Method

put

Response Class

void

Response Content-Type: application/xml, application/json

Parameters

Table 111. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	N/A		N/A	body	FirmwareUpdateRequest

Response Status Codes

Table 112. Response Status Codes

HTTP Status Code	Reason
201	Device updated Firmware
400	Invalid Device
404	Device not found in inventory

/ManagedDevice/export/csv

Description

Exports all Devices in csv format

Method

get

Response Class

void

Response Content-Type: application/octet-stream

Response Status Codes

Table 113. Response Status Codes

HTTP Status Code	Reason
400	Bad Request, verify input parameters are correct
401	No login information specified in the request
403	User doesn't have privileges to access this operation
500	User Log Internal Error, contact your system administrator

/ManagedDevice/withcompliance

Description

Retrieve all Devices from Inventory with filter.

Method

get

Response Class

[ManagedDevice]

Response Content-Type: application/xml, application/json

Parameters

Table 114. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
sort	N/A	Sort Column	N/A	query	string
filter	N/A	Filter Criteria	N/A	query	array
offset	N/A	Pagination Offset	0	query	integer
limit	N/A	Page Limit	50	query	integer

Response Status Codes

Table 115. Response Status Codes

HTTP Status Code	Reason
200	Device(s) retrieved from inventory on filter, sort, paginate

/ManagedDevice/puppet/{certName}

Description

Retrieve Device from Inventory based on certName

Method

get

Response Class

ManagedDevice

Response Content-Type: application/xml, application/json

Parameters

Table 116. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
certName	(required)	Certificate Name	N/A	path	string

Response Status Codes

Table 117. Response Status Codes

HTTP Status Code	Reason
200	Device retrieved from inventory
404	Device not found in inventory

/ManagedDevice/puppet/{certName}

Description

Update Device in Inventory

Method

put

Response Class

void

Response Content-Type: application/xml, application/json

Parameters

Table 118. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	N/A	Device to be updated to	N/A	body	ManagedDevice
certName	(required)	Certificate Name	N/A	path	string

Response Status Codes

Table 119. Response Status Codes

HTTP Status Code	Reason
201	Device updated in inventory
400	Invalid Device
404	Device not found in inventory

/ManagedDevice/withcompliance/{refId}

Description

Retrieve Device from Inventory based on refId

Method

get

Response Class

ManagedDevice

Response Content-Type: application/xml, application/json

Parameters

Table 120. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
refId	(required)	Reference Id	N/A	path	string

Response Status Codes

Table 121. Response Status Codes

HTTP Status Code	Reason
200	Device retrieved from inventory
404	Device not found in inventory

/NTP/

Description

Retrieve NTP settings from ASM appliance

Method

get

Response Class

ntpSettings

Response Content-Type: application/json, application/xml

Response Status Codes

Table 122. Response Status Codes

HTTP Status Code	Reason
200	Retrieved NTP settings from ASM appliance successfully.
500	Unable to get NTP settings from the ASM appliance.

/NTP/

Description

Apply NTP settings on ASM appliance

Method

put

Response Class

void

Response Content-Type: application/json, application/xml

Parameters

Table 123. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	(required)	NTPSetting object that needs to be set on ASM appliance	N/A	body	ntpSettings

Response Status Codes

Table 124. Response Status Codes

HTTP Status Code	Reason
204	Successfully set the NTP settings to on.
500	Unable to turn on NTP settings on the ASM appliance.

/NTP/

Description

Turn off NTP settings on ASM appliance

Method

delete

Response Class

void

Response Content-Type: application/json, application/xml

Response Status Codes

Table 125. Response Status Codes

HTTP Status Code	Reason
204	Successfully set the NTP settings to off.
500	Unable to turn off NTP settings on the ASM appliance.

/Network/

Description

Retrieve networks

Method

get

Response Class

[Network]

Response Content-Type: application/json, application/xml

Parameters

Table 126. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
sort	N/A	Valid sort columns are: name,description,type,vlanId,createdBy,createDate,updatedBy,updateDate	N/A	query	string
filter	N/A	Valid filter columns are: name,description,type,vlanId,createdBy,updateBy,createDate,updateDate	N/A	query	array
offset	N/A	Specify pagination offset	0	query	integer
limit	N/A	Specify page limit	50	query	integer

Response Status Codes

Table 127. Response Status Codes

HTTP Status Code	Reason
200	Networks is retrieved
400	Problem with a query parameter, check response for details
404	Networks not found

/Network/

Description

Add a new network

Method

post

Response Class

Network

Response Content-Type: application/json, application/xml

Parameters

Table 128. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	(required)	Network object that needs to be added to ASM	N/A	body	Network

Response Status Codes

Table 129. Response Status Codes

HTTP Status Code	Reason
201	Network created
400	Invalid input or network supplied or verify other input data is correct
409	Network already exists

/Network/export/csv

Description

Exports all Networks in csv format

Method

get

Response Class

void

Response Content-Type: application/octet-stream

Response Status Codes

Table 130. Response Status Codes

HTTP Status Code	Reason
400	Bad Request, verify input parameters are correct
401	No login information specified in the request
403	User doesn't have privileges to access this operation
500	User Log Internal Error, contact your system administrator

/Network/{networkId}

Description

Find a network by id

Method

get

Response Class

Network

Response Content-Type: application/json, application/xml

Parameters

Table 131. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
networkId	(required)	Id of network to retrieve	N/A	path	string

Response Status Codes

Table 132. Response Status Codes

HTTP Status Code	Reason
200	Network retrieved successfully
404	Network not found

/Network/{networkId}

Description

Update an existing network

Method

put

Response Class

void

Response Content-Type: application/xml, application/json

Parameters

Table 133. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	(required)	Network object that contains update fields	N/A	body	Network
networkId	(required)	Id of network to update	N/A	path	string

Response Status Codes

Table 134. Response Status Codes

HTTP Status Code	Reason
204	Network updated successfully
400	Invalid input for id or network supplied or verify other input data is correct
404	Network not found
409	Network with similar name already exists

/Network/{networkId}

Description

Delete an existing network

Method

delete

Response Class

void

Response Content-Type: application/xml, application/json

Parameters

Table 135. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
networkId	(required)	Id of network to delete	N/A	path	string

Response Status Codes

Table 136. Response Status Codes

HTTP Status Code	Reason
204	Network deleted successfully
400	Invalid network id supplied

/Network/ipAddress/assign

Description

Assigns a list of ip addresses to a usage Guid

Method

put

Response Class

string

Response Content-Type: application/json, application/xml

Parameters

Table 137. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
networkId	(required)	Id of network to update	N/A	query	string
numberToReserve	(required)	Number of ip addresses to reserve	N/A	query	integer
usageGUID	(required)	Usage GUID to associate with ipAddress	N/A	query	string

Response Status Codes

Table 138. Response Status Codes

HTTP Status Code	Reason
204	IP Addresses assigned successfully
400	Invalid network id supplied

/Network/ipAddress/release

Description

Releases ip addresses assigned to a usage guid

Method

put

Response Class

void

Response Content-Type: application/xml, application/json

Parameters

Table 139. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
usageGUID	(required)	Usage GUID associated with ipAddress(es)	N/A	query	string

Response Status Codes

Table 140. Response Status Codes

HTTP Status Code	Reason
204	Released IP addresses successfully
400	Invalid input for usageGUID
404	Reservation associated with usageGUID not found

/Network/{networkId}/usageids

Description

Find Usage Ids of network by id

Method

get

Response Class

UsageIdList

Response Content-Type: application/json, application/xml

Parameters

Table 141. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
networkId	(required)	Id of network to retrieve from	N/A	path	string

Response Status Codes

Table 142. Response Status Codes

HTTP Status Code	Reason
200	Network Usage Ids retrieved successfully
404	Network Usage Ids not found

/Network/{networkId}/ipaddresses

Description

Find IPs of network by id

Method

get

Response Class

[IpAddress]

Response Content-Type: application/json, application/xml

Parameters

Table 143. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
networkId	(required)	Id of network to retrieve from	N/A	path	string

Response Status Codes

Table 144. Response Status Codes

HTTP Status Code	Reason
200	Network IPs retrieved successfully
404	Network IPs not found

/OSRepository/

Description

Gets a list of existing ASM OS Repositories

Method

get

Response Class

OSRepository

Response Content-Type: application/xml, application/json

Response Status Codes

Table 145. Response Status Codes

HTTP Status Code	Reason
200	All OSRepository Retrieved

/OSRepository/

Description

Creates a new ASM OS Repository

Method

post

Response Class

OSRepository

Response Content-Type: application/xml, application/json

Parameters

Table 146. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	N/A		N/A	body	OSRepository

Response Status Codes

Table 147. Response Status Codes

HTTP Status Code	Reason
201	OS repository created
400	Bad Request, verify OS repository data object is correct

/OSRepository/{id}

Description

Gets the ASM OS Repository by its ID

Method

get

Response Class

OSRepository

Response Content-Type: application/xml, application/json

Parameters

Table 148. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
id	(required)		N/A	path	string

Response Status Codes

Table 149. Response Status Codes

HTTP Status Code	Reason
200	OSRepository Retrieved

/OSRepository/{id}

Description

Update OSRepository

Method

put

Response Class

OSRepository

Response Content-Type: application/xml, application/json

Parameters

Table 150. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	N/A		N/A	body	OSRepository
id	(required)		N/A	path	string
sync	N/A	Sync ISO	false	query	boolean

Response Status Codes

Table 151. Response Status Codes

HTTP Status Code	Reason
204	OSRepository updated
400	Invalid Parameters to Update OSRepository or problem with the Resource Adapters
404	OSRepository not found

/OSRepository/{id}

Description

Deletes an existing ASM OS Repository

Method

delete

Response Class

OSRepository

Response Content-Type: application/xml, application/json

Parameters

Table 152. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
id	(required)		N/A	path	string

Response Status Codes

Table 153. Response Status Codes

HTTP Status Code	Reason
200	Successfully deleted OS repo
400	Bad Request, verify id is correct
404	Bad Request, verify id is correct

/OSRepository/sync/{id}

Description

Syncs the ASM OS Repository by its ID

Method

put

Response Class

OSRepository

Response Content-Type: application/xml, application/json

Parameters

Table 154. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	N/A	osRepo	N/A	body	OSRepository
id	(required)		N/A	path	string

Response Status Codes

Table 155. Response Status Codes

HTTP Status Code	Reason
200	OSRepository Sync

/OSRepository/connection

Description

Tests the connection to a remote path

Method

post

Response Class

OSRepository

Response Content-Type: application/xml, application/json

Parameters

Table 156. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	N/A		N/A	body	OSRepository

Response Status Codes

Table 157. Response Status Codes

HTTP Status Code	Reason
201	Connection to remote path successful

/Proxy/

Description

Retrieve proxy settings from ASM appliance

Method

get

Response Class

proxySettings

Response Content-Type: application/json, application/xml

Response Status Codes

Table 158. Response Status Codes

HTTP Status Code	Reason
200	ASM appliance proxy settings retrieved successfully
500	Unable to retrieve proxy settings from ASM appliance

/Proxy/

Description

Set the proxy settings on ASM appliance

Method

put

Response Class

proxySettings

Response Content-Type: application/json, application/xml

Parameters

Table 159. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	(required)	ProxySetting object that needs to be set on ASM appliance	N/A	body	proxySettings

Response Status Codes

Table 160. Response Status Codes

HTTP Status Code	Reason
200	Proxy settings set successfully in ASM appliance
500	Unable to set the proxy settings in ASM appliance

/Proxy/test

Description

Retrieve proxy settings from ASM appliance

Method

post

Response Class

testProxyResponse

Response Content-Type: application/json, application/xml

Parameters

Table 161. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	(required)	ProxySetting object that needs to be tested on ASM appliance	N/A	body	proxySettings

Response Status Codes

Table 162. Response Status Codes

HTTP Status Code	Reason
200	Passed test for proxy settings defined in ASM appliance
500	Unable to test proxy settings defined in ASM appliance

/Server/

Description

Gets a list of servers

Method

get

Response Class

[Server]

Response Content-Type: application/xml, application/json

Parameters

Table 163. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
sort	N/A	Valid sort columns: health,managementIP,serviceT	N/A	query	string

Parameter	Value	Description	Default Value	Parameter Type	Data Type
filter	N/A	Valid filter columns: health,managementIP,serviceTag,powerState,credentialId	N/A	query	array
offset	N/A	Specify pagination offset	0	query	integer
limit	N/A	Specify page limit	50	query	integer

Response Status Codes

Table 164. Response Status Codes

HTTP Status Code	Reason
200	Retrieved
400	Bad Request, verify input parameters are correct
401	No login information specified in the request
403	User doesn't have privileges to access this operation
500	Server Internal Error, contact your system administrator

/Server/count

Description

Gets a count of servers

Method

get

Response Class

integer

Response Content-Type: application/xml, application/json

Parameters

Table 165. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
filter	N/A	Valid filter columns: health,managementIP,serviceTag,powerState,credentialId	N/A	query	array

Response Status Codes

Table 166. Response Status Codes

HTTP Status Code	Reason
200	Retrieved
400	Bad Request, verify input parameters are correct
401	No login information specified in the request
403	User doesn't have privileges to access this operation
500	Server Internal Error, contact your system administrator

/Server/{refId}

Description

Retrieves an individual Server

Method

get

Response Class

Server

Response Content-Type: application/xml, application/json

Parameters

Table 167. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
refId	(required)	Reference id of server	N/A	path	string

Response Status Codes

Table 168. Response Status Codes

HTTP Status Code	Reason
200	Retrieved
400	Bad Request, verify refId is correct
401	No login information specified in the request
403	User doesn't have privileges to access this operation
404	Server not found
500	Server Internal Error, contact your system administrator

/Server/{refId}

Description

Removes a Server

Method

delete

Response Class

void

Response Content-Type: application/xml, application/json

Parameters

Table 169. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
refId	(required)	Reference id of server	N/A	path	string

Response Status Codes

Table 170. Response Status Codes

HTTP Status Code	Reason
204	Deleted
400	Bad Request, verify refId is correct
401	No login information specified in the request
403	User doesn't have privileges to access this operation
500	Server Internal Error, contact your system administrator

/Server/serviceTag/{serviceTag}

Description

Retrieves an individual Server with serviceTag

Method

get

Response Class

Server

Response Content-Type: application/xml, application/json

Parameters

Table 171. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
serviceTag	(required)	serviceTag of the server	N/A	path	string

Response Status Codes

Table 172. Response Status Codes

HTTP Status Code	Reason
200	Retrieved
400	Bad Request, verify refId is correct
401	No login information specified in the request

HTTP Status Code	Reason
403	User doesn't have privileges to access this operation
404	Server not found
500	Server Internal Error, contact your system administrator

/ServiceTemplate/

Description

Retrieve all ServiceTemplates with filter, sort, paginate which returns Array of ServiceTemplate.class

Method

get

Response Class

[ServiceTemplate]

Response Content-Type: application/xml, application/json

Parameters

Table 173. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
sort	N/A	Supported sort columns are: name,createdBy,createdDate,updatedBy,updatedDate	N/A	query	string
filter	N/A	Supported filter columns are: name,draft,createdBy,updatedBy,createdDate,updatedDate	N/A	query	array
offset	N/A	Pagination Offset	0	query	integer
limit	N/A	Page Limit	50	query	integer
full	N/A	Full or Brief	false	query	boolean

Response Status Codes

Table 174. Response Status Codes

HTTP Status Code	Reason
200	All ServiceTemplates Retrived on filter, sort, paginate

/ServiceTemplate/

Description

Create new ServiceTemplate

Method

post

Response Class

ServiceTemplate

Response Content-Type: application/xml, application/json

Parameters

Table 175. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	N/A	Infrastructure Template to be created	N/A	body	ServiceTemplate

Response Status Codes

Table 176. Response Status Codes

HTTP Status Code	Reason
201	ServiceTemplate created
400	Invalid Parameters to create ServiceTemplate or problems with the Resource Adapters
409	Template already exists

/ServiceTemplate/{id}

Description

Retrieve ServiceTemplate based on ServiceTemplate id

Method

get

Response Class

ServiceTemplate

Response Content-Type: application/xml, application/json

Parameters

Table 177. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
id	(required)	Infrastructure Template Id (String)	N/A	path	string
includeBrownfieldVmMangers	N/A	Will return Managed and Reserved VM Managers if set to true, otherwise only returns Discovered.	false	query	boolean

Response Status Codes

Table 178. Response Status Codes

HTTP Status Code	Reason
200	ServiceTemplate retrieved
404	ServiceTemplate not found

/ServiceTemplate/{id}

Description

Update Template

Method

put

Response Class

ServiceTemplate

Response Content-Type: application/xml, application/json

Parameters

Table 179. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	N/A	Infrastructure Template to be updated to	N/A	body	ServiceTemplate
id	(required)	Infrastructure Template Id (String)	N/A	path	string

Response Status Codes

Table 180. Response Status Codes

HTTP Status Code	Reason
204	InfrastructureTemplate updated
400	Invalid Parameters to Update ServiceTemplate
404	Template not found

/ServiceTemplate/{id}

Description

Delete ServiceTemplate -- this operation is idempotent

Method

delete

Response Class

void

Response Content-Type: application/xml, application/json

Parameters

Table 181. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
id	(required)	Infrastructure Template Id (String)	N/A	path	string

Response Status Codes

Table 182. Response Status Codes

HTTP Status Code	Reason
204	ServiceTemplate deleted
500	Unable to delete ServiceTemplate.

/ServiceTemplate/upload

Description

Upload a ServiceTemplate

Method

post

Response Class

ServiceTemplate

Response Content-Type: application/xml, application/json

Parameters

Table 183. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	N/A	Service Template upload request	N/A	body	ServiceTemplateUploadRequest

Response Status Codes

Table 184. Response Status Codes

HTTP Status Code	Reason
201	ServiceTemplate uploaded successfully
400	Invalid Parameters to upload template

/ServiceTemplate/export

Description

Export a ServiceTemplate

Method

post

Response Class

ServiceTemplate

Response Content-Type: application/xml, application/json

Parameters

Table 185. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	N/A	Service Template	N/A	body	ServiceTemplate
encPassword	N/A	Encryption password	N/A	query	string
useEncPwdFromBackup	N/A	use password from backup	N/A	query	boolean

Response Status Codes

Table 186. Response Status Codes

HTTP Status Code	Reason
201	ServiceTemplate exported successfully
400	Invalid Parameters to copy ServiceTemplate or problems with the Resource Adapters
404	Template not found

/ServiceTemplate/{id}/copy

Description

Copy a ServiceTemplate

Method

post

Response Class

ServiceTemplate

Response Content-Type: application/xml, application/json

Parameters

Table 187. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	N/A	Infrastructure Template settings	N/A	body	ServiceTemplate
id	(required)	Infrastructure Template Id (String)	N/A	path	string

Response Status Codes

Table 188. Response Status Codes

HTTP Status Code	Reason
201	ServiceTemplate copied successfully
400	Invalid Parameters to copy ServiceTemplate or problems with the Resource Adapters
409	Template already exists

/ServiceTemplate/export/csv

Description

Exports all Templates in csv format

Method

get

Response Class

void

Response Content-Type: application/octet-stream

Response Status Codes

Table 189. Response Status Codes

HTTP Status Code	Reason
400	Bad Request, verify input parameters are correct
401	No login information specified in the request
403	User doesn't have privileges to access this operation
500	User Log Internal Error, contact your system administrator

/ServiceTemplate/uploadConfig

Description

Retrieve Default Template customized with uploaded config

Method

get

Response Class

ServiceTemplate

Response Content-Type: application/xml, application/json

Parameters

Table 190. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
configPath	N/A	Uploaded Config Path importConfig(String)	N/A	query	string

Response Status Codes

Table 191. Response Status Codes

HTTP Status Code	Reason
200	ServiceTemplate retrieved
404	ServiceTemplate not found

/ServiceTemplate/updateParameters

Description

Create new ServiceTemplate

Method

post

Response Class

ServiceTemplate

Response Content-Type: application/xml, application/json

Parameters

Table 192. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	N/A	Fills in any missing parameters required to deploy the template	N/A	body	ServiceTemplate

Response Status Codes

Table 193. Response Status Codes

HTTP Status Code	Reason
200	Parameters updated

/ServiceTemplate/device/{deviceId}

Description

Retrieve Default Template customized for specified device

Method

get

Response Class

ServiceTemplate

Response Content-Type: application/xml, application/json

Parameters

Table 194. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
deviceId	(required)	Device Id (String)	N/A	path	string

Response Status Codes

Table 195. Response Status Codes

HTTP Status Code	Reason
200	ServiceTemplate retrieved
404	ServiceTemplate not found

/ServiceTemplate/{id}/mapToPhysicalResources

Description

Do Physical Resource allocation based on ServiceTemplate

Method

post

Response Class

ServiceTemplate

Response Content-Type: application/xml, application/json

Parameters

Table 196. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
id	(required)	Infrastructure Template Id (String)	N/A	path	string

Response Status Codes

Table 197. Response Status Codes

HTTP Status Code	Reason
201	Map Service Template to physical resources
400	Invalid Parameters to copy ServiceTemplate or problems with the Resource Adapters
409	Template already exists

/ServiceTemplate/components/service/{serviceId}

Description

Update Template by analysing related components.

Method

put

Response Class

ServiceTemplate

Response Content-Type: application/xml, application/json

Parameters

Table 198. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	N/A	Template to update	N/A	body	ServiceTemplate
serviceId	(required)	Service Id	N/A	path	string

Response Status Codes

Table 199. Response Status Codes

HTTP Status Code	Reason
200	ServiceTemplate updated
404	ServiceTemplate not found

/ServiceTemplate/components/template/{templateId}

Description

Update Template by analysing related components.

Method

put

Response Class

ServiceTemplate

Response Content-Type: application/xml, application/json

Parameters

Table 200. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	N/A	Template to update	N/A	body	ServiceTemplate
templateId	(required)	Template Id	N/A	path	string

Response Status Codes

Table 201. Response Status Codes

HTTP Status Code	Reason
200	ServiceTemplate updated
404	ServiceTemplate not found

/ServiceTemplate/service/{serviceId}/{componentType}

Description

Retrieve Default Template with components refined for specified template ID

Method

get

Response Class

ServiceTemplate

Response Content-Type: application/xml, application/json

Parameters

Table 202. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
serviceId	(required)	Service Id (String)	N/A	path	string
componentType	(required)	Template Component Type (String)	N/A	path	string

Response Status Codes

Table 203. Response Status Codes

HTTP Status Code	Reason
200	ServiceTemplate retrieved
404	ServiceTemplate not found

/ServiceTemplate/template/{templateId}/ {componentType}

Description

Retrieve Default Template with components refined for specified template ID

Method

get

Response Class

ServiceTemplate

Response Content-Type: application/xml, application/json

Parameters

Table 204. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
templateId	(required)	Template Id (String)	N/A	path	string
componentType	(required)	Template Component Type (String)	N/A	path	string

Response Status Codes

Table 205. Response Status Codes

HTTP Status Code	Reason
200	ServiceTemplate retrieved
404	ServiceTemplate not found

/Timezone/

Description

Retrieve ASM appliance timezone

Method

get

Response Class

timeZone

Response Content-Type: application/json, application/xml

Response Status Codes

Table 206. Response Status Codes

HTTP Status Code	Reason
200	ASM appliance timezone retrieved successfully.
500	Unable to retrieve ASM appliance Time Zone

/Timezone/

Description

Sets the appliance timezone

Method

put

Response Class

Response

Response Content-Type: application/json, application/xml

Parameters

Table 207. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	(required)	TimeZoneInfo that needs to be set on ASM appliance	N/A	body	timeZone

Response Status Codes

Table 208. Response Status Codes

HTTP Status Code	Reason
200	ASM appliance timezone set successfully.
500	Unable to set the ASM appliance Time Zone

/Timezone/all

Description

Retrieve list of available timezones

Method

get

Response Class

availableTimeZones

Response Content-Type: application/json, application/xml

Parameters

Table 209. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
offset	N/A	Specify pagination offset	N/A	query	integer
limit	N/A	Specify page limit, can not exceed system maximum limit.	N/A	query	integer

Response Status Codes

Table 210. Response Status Codes

HTTP Status Code	Reason
200	Available timezones list retrieved successfully.
500	Unable to retrieve list of available timezones

/User/

Description

Retrieve a list of Users

Method

get

Response Class

[User]

Response Content-Type: application/json, application/xml

Parameters

Table 211. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
filter	N/A	Valid filter columns: userName,domainName,firstName	N/A	query	array

Parameter	Value	Description	Default Value	Parameter Type	Data Type
		me,lastName,enabled,systemUser,email,phoneNumber,role,createdDate,createdBy,updatedDate,updatedBy			
offset	N/A	Specify pagination offset	0	query	integer
limit	N/A	Specify page limit	50	query	integer
sort	N/A	Valid sort columns: userName,domainName,firstName,lastName,enabled,email,phoneNumber,role,createdDate,createdBy,updatedDate,updatedBy	N/A	query	string

Response Status Codes

Table 212. Response Status Codes

HTTP Status Code	Reason
200	OK

/User/

Description

Add a new user

Method

post

Response Class

User

Response Content-Type: application/json, application/xml

Parameters

Table 213. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	(required)	User object that needs to be added to ASM	N/A	body	User

Response Status Codes

Table 214. Response Status Codes

HTTP Status Code	Reason
201	User created
400	Invalid input
409	User already exists

/User/{userId}

Description

Find a user by Id

Method

get

Response Class

User

Response Content-Type: application/json, application/xml

Parameters

Table 215. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
userId	(required)	Id of user to retrieve	N/A	path	integer

Response Status Codes

Table 216. Response Status Codes

HTTP Status Code	Reason
200	OK
404	User not found

/User/{userId}

Description

Update an existing user

Method

put

Response Class

User

Response Content-Type: application/json, application/xml

Parameters

Table 217. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	(required)	User object that contains update fields	N/A	body	User
userId	(required)	ID of user to update	N/A	path	integer

Response Status Codes

Table 218. Response Status Codes

HTTP Status Code	Reason
204	Update completed successfully
400	Problem with the update data, check error response for details
404	User not found

/User/{userId}

Description

Delete an existing user

Method

delete

Response Class

void

Response Content-Type: application/xml, application/json

Parameters

Table 219. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
userId	(required)	Id of user to delete	N/A	path	integer

Response Status Codes

Table 220. Response Status Codes

HTTP Status Code	Reason
204	Delete completed successfully
404	User not found

/WizardStatus/

Description

Retrieve ASM Setup Status

Method

get

Response Class

wizardStatus

Response Content-Type: application/json, application/xml

Response Status Codes

Table 221. Response Status Codes

HTTP Status Code	Reason
204	Initial Wizard Status retrieved successfully.
500	Get Initial Wizard Status operation failed

/WizardStatus/

Description

Set the ASM Setup Status

Method

put

Response Class

wizardStatus

Response Content-Type: application/json, application/xml

Parameters

Table 222. Parameters

Parameter	Value	Description	Default Value	Parameter Type	Data Type
body	(required)	Initial Wizard status that needs to be set in ALCM	N/A	body	wizardStatus

Response Status Codes

Table 223. Response Status Codes

HTTP Status Code	Reason
204	Initial Wizard Status updated successfully.
400	Request parameter is invalid.
500	Update Initial Wizard Status operation failed

Appendix B - Model Reference

BrownfieldStatus

Enum: BrownfieldStatus

Table 224. BrownfieldStatus

Property Name	Property Type
values	["NEWLY_AVAILABLE", "NOT_APPLICABLE", "UNAVAILABLE_RELATED_SERVER_NOT_IN_INVENTORY", "UNAVAILABLE_RELATED_SERVER_IN_EXISTING_SERVICE", "UNAVAILABLE_RELATED_SERVER_NOT_MANAGED_OR_RESERVED", "UNAVAILABLE_NOT_IN_INVENTORY", "UNAVAILABLE_NOT_MANAGED_OR_RESERVED", "UNAVAILABLE_IN_EXISTING_SERVICE", "UNAVAILABLE_THIS_DEVICE_AND_RELATED_SERVER_NOT_IN_INVENTORY", "UNAVAILABLE_THIS_DEVICE_AND_RELATED_SERVER_NOT_MANAGED_OR_RESERVED", "UNAVAILABLE_THIS_DEVICE_NOT_MANAGED_OR_RESERVED_AND_RELATED_SERVER_NOT_IN_INVENTORY", "UNAVAILABLE_THIS_DEVICE_NOT_MANAGED_OR_RESERVED_AND_RELATED_SERVER_IN_EXISTING_SERVICE", "UNAVAILABLE_THIS_DEVICE_NOT_IN_INVENTORY_AND_RELATED_SERVER_NOT_MANAGED_OR_RESERVED", "UNAVAILABLE_THIS_DEVICE_NOT_IN_INVENTORY_AND_RELATED_SERVER_IN_EXISTING_SERVICE", "AVAILABLE", "CURRENTLY_DEPLOYED_IN_BROWNFIELD"]

ComparatorValue

Enum: ComparatorValue

Table 225. ComparatorValue

Property Name	Property Type
values	["minimum", "maximum", "exact"]

CompliantState

Enum: CompliantState

Table 226. CompliantState

Property Name	Property Type
values	["NA", "compliant", "noncompliant", "unknown", "updaterequired"]

DeploymentHealthStatusType

Enum: DeploymentHealthStatusType

Table 227. DeploymentHealthStatusType

Property Name	Property Type
values	["green", "yellow", "red"]

DeploymentStatusType

Enum: DeploymentStatusType

Table 228. DeploymentStatusType

Property Name	Property Type
values	["Pending", "In Progress", "Deployed", "Error", "Cancelled", "Firmware Updating"]

DeviceHealth

Enum: DeviceHealth

Table 229. DeviceHealth

Property Name	Property Type
values	["green", "yellow", "red", "unknown"]

DeviceState

Enum: DeviceState

Table 230. DeviceState

Property Name	Property Type
values	["Discovered", "Updating", "Configuration Error", "Pending Configuration Template", "Pending Delete", "Pending", "Ready", "Discovery Failed", "Deployment Error", "Delete Failed", "Deleted", "Deployed", "Deploying", "unmanaged", "reserved"]

DeviceType

Enum: DeviceType

Table 231. DeviceType

Property Name	Property Type
values	["Chassis", "Server", "Switch", "Chassis", "Chassis", "Rack Server", "Blade Server", "Switch", "Switch", "Storage Group", "Storage Group", "Storage Group", "Switch", "Switch", "VM Manager", "Virtual Machine", "Storage Group", "VM Manager", "Unknown", "Server", "Chassis", "FX Server", "EM", "Tower Server"]

DiscoverDeviceType

Enum: DiscoverDeviceType

Table 232. DiscoverDeviceType

Property Name	Property Type
values	["CMC", "CMC_FX2", "IDRAC7", "IDRAC8", "UNKNOWN", "SERVER", "CSERVER", "VRTX", "VCENTER", "EQUALLOGIC", "COMPELLENT", "FORCE10", "FORCE10_S4810", "FORCE10_S5000", "FORCE10_S6000", "FORCE10_S4048", "FORCE10IOM", "FX2_IOM", "DELL_IOM_84", "BROCADE", "POWERCONNECT", "POWERCONNECT_N3000", "POWERCONNECT_N4000", "NETAPP", "CISCONEXUS", "EQUALLOGIC_NODISCOVER", "SCVMM", "BMC", "EM"]

DiscoveryStatus

Enum: DiscoveryStatus

Table 233. DiscoveryStatus

Property Name	Property Type
values	["Pending", "Connected", "Unsupported", "Inprogress", "Success", "Failed to discover", "Error discovering", "Ignore device at this time"]

DiskMediaType

Enum: DiskMediaType

Table 234. DiskMediaType

Property Name	Property Type
values	["requiresd", "requirehdd", "any"]

Health

Enum: Health

Table 235. Health

Property Name	Property Type
values	["GREEN", "YELLOW", "RED", "UNKNOWN"]

HotSpareStatus

Enum: HotSpareStatus

Table 236. HotSpareStatus

Property Name	Property Type
values	["No", "Dedicated", "Global"]

IOMLocationType

Enum: IOMLocationType

Table 237. IOMLocationType

Property Name	Property Type
values	["A1", "A2", "B1", "B2", "C1", "C2", "UNKNOWN"]

IdentityType

Enum: IdentityType

Table 238. IdentityType

Property Name	Property Type
values	["UNKNOWN", "WWPN", "WWNN", "ISCSI_MAC", "FIPS_MAC", "LAN_MAC", "IQN_INITIATOR", "IP_INITIATOR", "BOOTLUN", "IQN_TARGET", "IP_TARGET", "WWPN_TARGET"]

LogSeverity

Enum: LogSeverity

Table 239. LogSeverity

Property Name	Property Type
values	["Normal", "Warning", "Critical", "Unknown"]

MediaType

Enum: MediaType

Table 240. MediaType

Property Name	Property Type
values	["SSD", "HDD", "ANY"]

NetworkMode

Enum: NetworkMode

Table 241. NetworkMode

Property Name	Property Type
values	["UNKNOWN", "LAN", "ISCSI", "FCOE"]

PhysicalType

Enum: PhysicalType

Table 242. PhysicalType

Property Name	Property Type
values	["BLADE", "RACK", "SLED"]

RepositoryState

Enum: RepositoryState

Table 243. RepositoryState

Property Name	Property Type
values	["copying", "errors", "available"]

RepositoryStatus

Enum: RepositoryStatus

Table 244. RepositoryStatus

Property Name	Property Type
values	["pending", "copying", "error", "available"]

ServerSlotType

Enum: ServerSlotType

Table 245. ServerSlotType

Property Name	Property Type
values	["HALF", "FULL", "QUARTER", "UNKNOWN"]

ServiceTemplateComponentSubType

Enum: ServiceTemplateComponentSubType

Table 246. ServiceTemplateComponentSubType

Property Name	Property Type
values	["CLASS", "TYPE", "HYPERVISOR"]

ServiceTemplateComponentType

Enum: ServiceTemplateComponentType

Table 247. ServiceTemplateComponentType

Property Name	Property Type
values	["CONFIGURATION", "TOR", "STORAGE", "SERVER", "CLUSTER", "VIRTUALMACHINE", "SERVICE", "TEST"]

ServiceTemplateSettingType

Enum: ServiceTemplateSettingType

Table 248. ServiceTemplateSettingType

Property Name	Property Type
values	["BOOLEAN", "STRING", "PASSWORD", "INTEGER", "LIST", "TEXT", "NETWORKCONFIGURATION", "ENUMERATED", "RAIDCONFIGURATION", "BIOSCONFIGURATION", "RADIO"]

UIRaidLevel

Enum: UIRaidLevel

Table 249. UIRaidLevel

Property Name	Property Type
values	["raid0", "raid1", "raid5", "raid6", "raid10", "raid50", "raid60"]

UpdateType

Enum: UpdateType

Table 250. UpdateType

Property Name	Property Type
values	["SERVICE", "DEVICE"]

AsmDetailedMessage

Name: AsmDetailedMessage

Table 251. AsmDetailedMessage

Property Name	Property Type	Required
timeStamp	string	false
messageBundle	string	false
severity	string	false
sequenceNumber	integer	false
category	string	false
displayMessage	string	false
messageCode	string	false
correlationId	string	false
agentId	string	false
responseAction	string	false
detailedMessage	string	false

AuthenticateRequest

Name: AuthenticateRequest

Table 252. AuthenticateRequest

Property Name	Property Type	Required
userName	string	false
domain	string	false

Property Name	Property Type	Required
password	string	false

AuthenticateResponse

Name: AuthenticateResponse

Table 253. AuthenticateResponse

Property Name	Property Type	Required
userName	string	true
apiSecret	string	true
domain	string	true
role	string	true
apiKey	string	true

ChassisController

Name: ChassisController

Table 254. ChassisController

Property Name	Property Type	Required
controllerPresent	boolean	false
controllerPrimary	boolean	false
controllerName	string	false
controllerFWVersion	string	false
id	string	false

ChassisPowerSupply

Name: ChassisPowerSupply

Table 255. ChassisPowerSupply

Property Name	Property Type	Required
present	boolean	false
capacity	string	false
powerStatus	string	false
id	string	false
slot	string	false

Enclosure

Name: Enclosure

Table 256. Enclosure

Property Name	Property Type	Required
fqdd	string	false
productName	string	false
id	string	false

FirmwareInventory

Name: FirmwareInventory

Table 257. FirmwareInventory

Property Name	Property Type	Required
deviceRef	string	false
firmwareId	string	false
fqdd	string	false
componentID	string	false
deviceId	string	false
vendorID	string	false
subdeviceId	string	false
subvendorID	string	false
bPresent	boolean	false
updateable	boolean	false
version	string	false
lastUpdateTime	string	false
componentType	string	false
name	string	false

GroupUser

Name: GroupUser

Table 258. GroupUser

Property Name	Property Type	Required
userName	string	false
userSeqId	integer	true
firstName	string	false
lastName	string	false
enabled	boolean	false
role	string	true

IKVM

Name: IKVM

Table 259. IKVM

Property Name	Property Type	Required
present	boolean	false
manufacturer	string	false
partNumber	string	false
firmwareVersion	string	false
name	string	false
id	string	false

IpAddress

Name: IpAddress

Table 260. IpAddress

Property Name	Property Type	Required
ipRangeld	string	false
iPAsLong	integer	false
iPAddress	string	false
usageld	string	true
id	string	false
state	string	true

IpRange

Name: IpRange

Table 261. IpRange

Property Name	Property Type	Required
startingIp	string	true
endingIp	string	true
id	string	false

Link

Name: Link

Table 262. Link

Property Name	Property Type	Required
rel	string	true
href	string	true
title	string	false
type	string	false

List

Name: List

Table 263. List

Property Name	Property Type	Required
empty	boolean	false

MemoryInventory

Name: MemoryInventory

Table 264. MemoryInventory

Property Name	Property Type	Required
model	string	false
manufacturer	string	false
fqdd	string	false
partNumber	string	false

Property Name	Property Type	Required
bankLabel	string	false
currentOperatingSpeed	integer	false
instanceID	string	false
lastSystemInventoryTime	string	false
manufactureDate	string	false
memoryType	string	false
primaryStatus	string	false
rank	string	false
speed	integer	false
serialNumber	string	false
lastUpdateTime	string	false
id	string	false
size	integer	false

OSRepository

Name: OSRepository

Table 265. OSRepository

Property Name	Property Type	Required
createdDate	string	false
imageType	string	false
sourcePath	string	false
razorName	string	false
inUse	boolean	false
username	string	false
createdBy	string	false
password	string	false
name	string	false
id	string	false
state	string	false

ProcessorInventory

Name: ProcessorInventory

Table 266. ProcessorInventory

Property Name	Property Type	Required
present	boolean	false
model	string	false
manufacturer	string	false
fqdd	string	false
cores	integer	false
enabledCores	integer	false
maxClockSpeed	integer	false
currentClockSpeed	integer	false
id	string	false

RejectedServer

Name: RejectedServer

Table 267. RejectedServer

Property Name	Property Type	Required
refId	string	false
serviceTag	string	false
ipaddress	string	false
reason	string	false

SelectedNIC

Name: SelectedNIC

Table 268. SelectedNIC

Property Name	Property Type	Required
fqdd	string	false
cardNumber	integer	false
portNumber	integer	false
id	string	false

ServerJobStatus

Name: ServerJobStatus

Table 269. ServerJobStatus

Property Name	Property Type	Required
createdDate	string	false
jobType	string	false
jobStatus	string	false
jobName	string	false
startedBy	string	false

ServiceTemplateOption

Name: ServiceTemplateOption

Table 270. ServiceTemplateOption

Property Name	Property Type	Required
dependencyValue	string	false
dependencyTarget	string	false
name	string	false
value	string	false

ServiceTemplateUploadRequest

Name: ServiceTemplateUploadRequest

Table 271. ServiceTemplateUploadRequest

Property Name	Property Type	Required
encryptionPassword	string	false
templateName	string	false
allStandardUsers	boolean	false
managePermissions	boolean	false
useEncPwdFromBackup	boolean	false
createCategory	boolean	false
firmwarePackageld	string	false
assignedUsers	[string]	false
manageFirmware	boolean	false
description	string	false
category	string	false
content	string	false

SoftwareComponent

Name: SoftwareComponent

Table 272. SoftwareComponent

Property Name	Property Type	Required
createdDate	string	false
updatedBy	string	false
updatedAt	string	false
deviceId	string	false
componentId	string	false
packageId	string	false
dellVersion	string	false
vendorVersion	string	false
subDeviceId	string	false
vendorId	string	false
subVendorId	string	false
hashMd5	string	false
systemIDs	[string]	false
category	string	false
createdBy	string	false
componentType	string	false
name	string	false
id	string	false
path	string	false

timeZone

Name: timeZone

Table 273. timeZone

Property Name	Property Type	Required
timeZoneId	string	false
timeZone	string	false

UsageldList

Name: UsageldList

Table 274. UsageldList

Property Name	Property Type	Required
usagelds	[string]	false

VM

Name: VM

Table 275. VM

Property Name	Property Type	Required
vmIpaddress	string	false
vmManufacturer	string	false
vmModel	string	false
vmServiceTag	string	false
certificateName	string	false

connected_switch

Name: connected_switch

Table 276. connected_switch

Property Name	Property Type	Required
localDevice	string	false
localPorts	[string]	false
remoteDeviceType	string	false
remoteDevice	string	false
localDeviceType	string	false
remotePorts	[string]	false

dhcpSettings

Name: dhcpSettings

Table 277. dhcpSettings

Property Name	Property Type	Required
subnet	string	false
netmask	string	false
startingIpAddress	string	false
endingIpAddress	string	false
defaultLeaseTime	integer	false
maxLeaseTime	integer	false
gateway	string	false
dns	string	false
enabled	boolean	false
domain	string	false

fc_interface

Name: fc_interface

Table 278. fc_interface

Property Name	Property Type	Required
fqdd	string	false
wwpn	string	false
activeZoneset	string	false
connectedZones	[string]	false
connectedSwitch	string	false

logEntry

Name: logEntry

Table 279. logEntry

Property Name	Property Type	Required
componentId	string	false
executionId	string	false
level	string	false
timestamp	string	false
message	string	false

ntpSettings

Name: ntpSettings

Table 280. ntpSettings

Property Name	Property Type	Required
preferredNTPServer	string	false
secondaryNTPServer	string	false

proxySettings

Name: proxySettings

Table 281. proxySettings

Property Name	Property Type	Required
userName	string	false
proxyServer	string	false
userCredentialEnabled	boolean	false
enabled	boolean	false
password	string	false
port	string	false

references

Name: references

Table 282. references

Property Name	Property Type	Required
policies	integer	false
devices	integer	false
totalReferences	integer	false

testProxyResponse

Name: testProxyResponse

Table 283. testProxyResponse

Property Name	Property Type	Required
testProxyResult	boolean	false
testProxyDescription	string	false

timeZone

Name: timeZone

Table 284. timeZone

Property Name	Property Type	Required
timeZoneId	string	false
timeZone	string	false

AsmDetailedMessageList

Name: AsmDetailedMessageList

Table 285. testProxyResponse

Property Name	Property Type	Required
messages	[AsmDetailedMessage]	false

DeploymentDevice

Name: DeploymentDevice

Table 286. DeploymentDevice

Property Name	Property Type	Required
refId	string	false
deviceType	DeviceType	false
serviceTag	string	false
ipAddress	string	false
statusMessage	string	false
healthMessage	string	false
brownfield	boolean	false
componentId	string	false
brownfieldStatus	BrownfieldStatus	false
deviceHealth	DeviceHealth	false

Property Name	Property Type	Required
compliantState	CompliantState	false
statusEndTime	string	false
statusStartTime	string	false
logDump	string	false
status	DeploymentStatusType	false
refType	string	false

DeviceDiscoveryRequest

Name: DeviceDiscoveryRequest

Table 287. DeviceDiscoveryRequest

Property Name	Property Type	Required
deviceType	DeviceType	false
deviceStartIp	string	false
deviceServerCredRef	string	false
deviceChassisCredRef	string	false
deviceSwitchCredRef	string	false
deviceSCVMMCredRef	string	false
deviceStorageCredRef	string	false
deviceVCenterCredRef	string	false
deviceEMCredRef	string	false
unmanaged	boolean	false
reserved	boolean	false
serverPoolId	string	false
config	string	false
deviceBMCServerCredRef	string	false
deviceEndIp	string	false

FirmwareComplianceComponents

Name: FirmwareComplianceComponents

Table 288. FirmwareComplianceComponents

Property Name	Property Type	Required
embeddedRepoComponents	[SoftwareComponent]	false
defaultRepoComponents	[SoftwareComponent]	false
compliantState	CompliantState	false

FirmwareUpdateRequest

Name: FirmwareUpdateRequest

Table 289. FirmwareUpdateRequest

Property Name	Property Type	Required
updateType	UpdateType	false
idList	[string]	false
scheduleType	string	false
exitMaintenanceMode	boolean	false
scheduleDate	string	false

GregorianCalendar

Name: GregorianCalendar

Table 290. GregorianCalendar

Property Name	Property Type	Required
weekDateSupported	boolean	false
weekYear	integer	false
weeksInWeekYear	integer	false
gregorianChange	string	false
timeZone	TimeZone	false
lenient	boolean	false
firstDayOfWeek	integer	false
minimalDaysInFirstWeek	integer	false
timeInMillis	integer	false
time	string	false

GroupUserList

Name: GroupUserList

Table 291. GroupUserList

Property Name	Property Type	Required
totalRecords	integer	false
groupUsers	[GroupUser]	false

IOM

Name: IOM

Table 292. IOM

Property Name	Property Type	Required
managementIP	string	false
managementIPStatic	boolean	false
serviceTag	string	false
health	Health	false
model	string	false
location	IOMLocationType	false
id	string	false
slot	integer	false
supported	boolean	false

LogicalNetworkIdentityInventory

Name: LogicalNetworkIdentityInventory

Table 293. LogicalNetworkIdentityInventory

Property Name	Property Type	Required
bPresent	boolean	false
identityType	IdentityType	false
permanentIdentity	string	false
currentIdentity	string	false
id	string	false

PhysicalDisk

Name: PhysicalDisk

Table 294. PhysicalDisk

Property Name	Property Type	Required
fqdd	string	false
hotSpareStatus	HotSpareStatus	false
mediaType	MediaType	false
driveNumber	integer	false
size	integer	false

Server

Name: Server

Table 295. Server

Property Name	Property Type	Required
managementIP	string	false
managementIPStatic	boolean	false
serviceTag	string	false
assetTag	string	false
health	Health	false
model	string	false
slotName	string	false
slotType	ServerSlotType	false
id	string	false
slot	string	false
supported	boolean	false

ServiceTemplateValid

Name: ServiceTemplateValid

Table 296. ServiceTemplateValid

Property Name	Property Type	Required
valid	boolean	false
messages	[AsmDetailedMessage]	false

SoftwareBundle

Name: SoftwareBundle

Table 297. SoftwareBundle

Property Name	Property Type	Required
deviceType	string	false
createdDate	string	false
updatedBy	string	false
updatedAt	string	false
softwareComponents	[SoftwareComponent]	false
userBundlePath	string	false
userBundle	boolean	false
userBundleHashMd5	string	false
deviceModel	string	false
criticality	string	false
fwRepositoryId	string	false
bundleDate	string	false
version	string	false
description	string	false
link	Link	false
createdBy	string	false
name	string	false
id	string	false

StaticNetworkConfiguration

Name: StaticNetworkConfiguration

Table 298. StaticNetworkConfiguration

Property Name	Property Type	Required
ipAddress	string	false
subnet	string	false
gateway	string	false
primaryDns	string	false
secondaryDns	string	false
dnsSuffix	string	false
ipRange	[IpRange]	false

VirtualDiskConfiguration

Name: VirtualDiskConfiguration

Table 299. VirtualDiskConfiguration

Property Name	Property Type	Required
raidlevel	UIRaidLevel	false
disktype	DiskMediaType	false
numberofdisks	integer	false
comparator	ComparatorValue	false
id	string	false

availableTimeZones

Name: availableTimeZones

Table 300. availableTimeZones

Property Name	Property Type	Required
availableTimeZones	[timeZone]	false

credential

Name: credential

Table 301. credential

Property Name	Property Type	Required
createdDate	string	false
updatedBy	string	false

Property Name	Property Type	Required
updatedAt	string	false
label	string	true
domain	string	false
link	Link	false
username	string	true
createdBy	string	false
password	string	false
id	string	false

wizardStatus

Name: wizardStatus

Table 302. wizardStatus

Property Name	Property Type	Required
isConfigureCompleted	boolean	false
isDeploymentCompleted	boolean	false
isNetworkCompleted	boolean	false
isTemplateCompleted	boolean	false
seqId	integer	false
isSetupCompleted	boolean	false
showGettingStarted	boolean	false
link	Link	false

Chassis

Name: Chassis

Table 303. Chassis

Property Name	Property Type	Required
managementIP	string	false
deviceId	string	false
managementIPStatic	boolean	false
credentialRefId	string	false
serviceTag	string	false
assetTag	string	false

Property Name	Property Type	Required
health	Health	false
controllers	[ChassisController]	false
powerSupplies	[ChassisPowerSupply]	false
kVMs	[IKVM]	false
defaultPowerCapUpperBoundBTU	integer	false
defaultPowerCapLowerBoundWatts	integer	false
defaultPowerCapLowerBoundBTU	integer	false
lastFirmwareUpdateTime	GregorianCalendar	false
iomCount	integer	false
serverCount	integer	false
iOMs	[IOM]	false
model	string	false
datacenter	string	false
aisle	string	false
rack	string	false
rackslot	string	false
dnsName	string	false
powerCapPercent	integer	false
defaultPowerCapUpperBoundWatts	integer	false
midPlaneVersion	string	false
lastUpdateTime	GregorianCalendar	false
servers	[Server]	false
name	string	false
refId	string	false
deviceType	string	false
url	Link	false
refType	string	false
displayName	string	false

ChassisLogEntry

Name: ChassisLogEntry

Table 304. ChassisLogEntry

Property Name	Property Type	Required
logTime	GregorianCalendar	false
severity	LogSeverity	false
message	string	false

DeviceDiscoveryRequests

Name: DeviceDiscoveryRequests

Table 305. DeviceDiscoveryRequests

Property Name	Property Type	Required
discoverIpRangeDeviceRequests	[DeviceDiscoveryRequest]	false

LogicalNetworkInterface

Name: LogicalNetworkInterface

Table 306. LogicalNetworkInterface

Property Name	Property Type	Required
vendorName	string	false
fqdd	string	false
bPresent	boolean	false
vendorId	string	false
productName	string	false
identityList	[LogicalNetworkIdentityInventory]	false
productId	string	false
networkMode	NetworkMode	false
relativeBandwidthWeight	integer	false
maxBandwidth	number	false
portId	string	false
id	string	false

Network

Name: Network

Table 307. Network

Property Name	Property Type	Required
vlanId	integer	false
staticNetworkConfiguration	StaticNetworkConfiguration	false
description	string	false
name	string	false
id	string	false
type	string	false
static	boolean	false

User

Name: User

Table 308. User

Property Name	Property Type	Required
userName	string	true
createdDate	GregorianCalendar	false
updatedBy	string	false
updatedDate	GregorianCalendar	false
groupName	string	false
email	string	false
phoneNumber	string	false
systemUser	boolean	false
userSeqId	integer	false
updatePassword	boolean	false
domainName	string	true
groupDN	string	false
firstName	string	false
lastName	string	false
enabled	boolean	false
link	Link	false

Property Name	Property Type	Required
role	string	true
createdBy	string	false
password	string	true

VirtualDisk

Name: VirtualDisk

Table 309. VirtualDisk

Property Name	Property Type	Required
raidLevel	UIRaidLevel	false
physicalDisks	[string]	false
mediaType	MediaType	false
controller	string	false
configuration	VirtualDiskConfiguration	false

asmCredential

Name: asmCredential

Table 310. asmCredential

Property Name	Property Type	Required
credential	credential	true
references	references	false
link	Link	false

Controller

Name: Controller

Table 311. Controller

Property Name	Property Type	Required
enclosures	[Enclosure]	false
fqdd	string	false
virtualDisks	[VirtualDisk]	false
physicalDisks	[PhysicalDisk]	false
numPhysDisks	integer	false

Property Name	Property Type	Required
id	string	false

Partition

Name: Partition

Table 312. Partition

Property Name	Property Type	Required
maximum	integer	false
minimum	integer	false
networks	[string]	false
fqdd	string	false
wwpn	string	false
networkObjects	[Network]	false
iscsiMacAddress	string	false
macAddress	string	false
partitionNo	integer	false
partitionIndex	integer	false
lanMacAddress	string	false
iscsiIQN	string	false
wwnn	string	false
portNo	integer	false
name	string	false
id	string	false

RAIDConfiguration

Name: RAIDConfiguration

Table 313. RAIDConfiguration

Property Name	Property Type	Required
externalSsdHotSpares	[string]	false
externalVirtualDisks	[VirtualDisk]	false
virtualDisks	[VirtualDisk]	false
externalHddHotSpares	[string]	false
hddHotSpares	[string]	false

Property Name	Property Type	Required
ssdHotSpares	[string]	false

credentialList

Name: credentialList

Table 314. credentialList

Property Name	Property Type	Required
totalRecords	integer	false
credentialList	[asmCredential]	false

Interface

Name: Interface

Table 315. Interface

Property Name	Property Type	Required
fqdd	string	false
maxPartitions	integer	false
partitioned	boolean	false
partitions	[Partition]	false
nictype	string	false
redundancy	boolean	false
enabled	boolean	false
name	string	false
id	string	false

SelectedServer

Name: SelectedServer

Table 316. SelectedServer

Property Name	Property Type	Required
refId	string	false
serviceTag	string	false
ipAddress	string	false
nics	[SelectedNIC]	false

Property Name	Property Type	Required
componentId	string	false
matchUnordered	boolean	false
raidConfiguration	RAIDConfiguration	false
exactMatch	boolean	false

DeploymentFilterResponse

Name: DeploymentFilterResponse

Table 317. DeploymentFilterResponse

Property Name	Property Type	Required
numberRequestedServers	integer	false
numberSelectedServers	integer	false
failedPoolId	string	false
failedPoolName	string	false
selectedServers	[SelectedServer]	false
rejectedServers	[RejectedServer]	false

Fabric

Name: Fabric

Table 318. Fabric

Property Name	Property Type	Required
maxPartitions	integer	false
partitioned	boolean	false
nictype	string	false
redundancy	boolean	false
fabrictype	string	false
usedforfc	boolean	false
nPorts	integer	false
nictypeSource	string	false
enabled	boolean	false
name	string	false
interfaces	[Interface]	false
id	string	false

NetworkConfiguration

Name: NetworkConfiguration

Table 319. NetworkConfiguration

Property Name	Property Type	Required
fabrics	[Fabric]	false
usedInterfaces	[Interface]	false
servertype	string	false
allUsedInterfaces	[Interface]	false
allNetworks	[Network]	false
interfaces	[Fabric]	false
id	string	false

ServerNetworkObjects

Name: ServerNetworkObjects

Table 320. ServerNetworkObjects

Property Name	Property Type	Required
physicalType	PhysicalType	false
razorPolicyName	string	false
networkConfig	NetworkConfiguration	false
relatedSwitches	[string]	false
portConnections	[connected_switch]	false
fcInterfaces	[fc_interface]	false
fcoeInterfaces	[fc_interface]	false
serialNumber	string	false
server	string	false
name	string	false

ServiceTemplateSetting

Name: ServiceTemplateSetting

Table 321. ServiceTemplateSetting

Property Name	Property Type	Required
min	integer	false
networks	[Network]	false
dependencyValue	string	false
dependencyTarget	string	false
optionsSortable	boolean	false
step	integer	false
requiredAtDeployment	boolean	false
hideFromTemplate	boolean	false
toolTip	string	false
infoIcon	boolean	false
possibleValues	[string]	false
possibleValuesDisplayName	[string]	false
maxLength	integer	false
networkConfiguration	NetworkConfiguration	false
raidConfiguration	RAIDConfiguration	false
max	integer	false
group	string	false
options	[ServiceTemplateOption]	false
generated	boolean	false
required	boolean	false
value	string	false
id	string	false
type	ServiceTemplateSettingType	false
readOnly	boolean	false
displayName	string	false

ServiceTemplateCategory

Name: ServiceTemplateCategory

Table 322. ServiceTemplateCategory

Property Name	Property Type	Required
parameters	[ServiceTemplateSetting]	false
id	string	false
displayName	string	false

component

Name: component

Table 323. component

Property Name	Property Type	Required
refId	string	false
manageFirmware	boolean	false
asmGUID	string	false
teardown	boolean	false
componentID	string	false
brownfield	boolean	false
helpText	string	false
componentValid	ServiceTemplateValid	false
cloned	boolean	false
subType	ServiceTemplateComponentSub Type	false
puppetCertName	string	false
associatedComponents	Map<string,Map<string,string>>	false
clonedFromId	string	false
relatedComponents	Map<string,string>	false
iP	string	false
serialNumber	string	false
configFile	string	false
name	string	false
resources	[ServiceTemplateCategory]	false
id	string	false
type	ServiceTemplateComponentType	false

FirmwareRepository

Name: FirmwareRepository

Table 324. FirmwareRepository

Property Name	Property Type	Required
createdDate	string	false
updatedBy	string	false
updatedAt	string	false
softwareComponents	[SoftwareComponent]	false
sourceLocation	string	false
sourceType	string	false
diskLocation	string	false
md5Hash	string	false
downloadStatus	RepositoryStatus	false
softwareBundles	[SoftwareBundle]	false
bundleCount	integer	false
defaultCatalog	boolean	false
componentCount	integer	false
deployments	[Deployment]	false
username	string	false
createdBy	string	false
filename	string	false
password	string	false
embedded	boolean	false
name	string	false
id	string	false
state	RepositoryState	false

ServiceTemplate

Name: ServiceTemplate

Table 325. ServiceTemplate

Property Name	Property Type	Required
createdDate	GregorianCalendar	false
updatedBy	string	false
updatedAt	GregorianCalendar	false
firmwareRepository	FirmwareRepository	false
draft	boolean	false
templateName	string	false
templateDescription	string	false
wizardPageNumber	integer	false
assignedUsers	[User]	false
allUsersAllowed	boolean	false
enableApps	boolean	false
enableVMs	boolean	false
enableCluster	boolean	false
enableServer	boolean	false
enableStorage	boolean	false
templateValid	ServiceTemplateValid	false
templateLocked	boolean	false
templateVersion	string	false
lastDeployedDate	GregorianCalendar	false
manageFirmware	boolean	false
components	[component]	false
category	string	false
createdBy	string	false
attachments	[string]	false
id	string	false

Deployment

Name: Deployment

Table 326. Deployment

Property Name	Property Type	Required
createdDate	GregorianCalendar	false
updatedBy	string	false
updatedAt	GregorianCalendar	false
firmwareRepository	FirmwareRepository	false
assignedUsers	[User]	false
allUsersAllowed	boolean	false
teardown	boolean	false
templateValid	boolean	false
brownfield	boolean	false
vDS	boolean	false
deploymentHealthStatusType	DeploymentHealthStatusType	false
canMigrate	boolean	false
scheduleDate	string	false
numberOfDeployments	integer	false
firmwareRepositoryId	string	false
canScaleupStorage	boolean	false
canScaleupNetwork	boolean	false
canScaleupServer	boolean	false
canScaleupVM	boolean	false
canDeleteResources	boolean	false
canRetry	boolean	false
canEdit	boolean	false
canDelete	boolean	false
canCancel	boolean	false
canScaleupApplication	boolean	false
canScaleupCluster	boolean	false
deploymentName	string	false
deploymentDescription	string	false
serviceTemplate	ServiceTemplate	false
deploymentDevice	[DeploymentDevice]	false
vms	[VM]	false
jobDetails	[logEntry]	false

Property Name	Property Type	Required
retry	boolean	false
updateServerFirmware	boolean	false
individualTeardown	boolean	false
overallDeviceHealth	DeviceHealth	false
compliant	boolean	false
createdBy	string	false
status	DeploymentStatusType	false
id	string	false
owner	string	false

DiscoveredDevices

Name: DiscoveredDevices

Table 327. DiscoveredDevices

Property Name	Property Type	Required
refId	string	true
deviceType	DeviceType	false
serviceTag	string	true
ipAddress	string	true
iomCount	integer	false
serverCount	integer	false
model	string	false
healthStatusMessage	string	false
serverType	string	false
credId	string	true
discoverDeviceType	DiscoverDeviceType	false
chassisId	string	false
firmwareDeviceInventories	[FirmwareDeviceInventory]	false
facts	string	false
statusMessage	string	false
deviceRefId	string	true
healthState	string	false
parentJobId	string	true

Property Name	Property Type	Required
unmanaged	boolean	false
reserved	boolean	false
serverPoolId	string	false
systemId	string	false
config	string	false
jobId	string	false
configuration	string	false
status	DiscoveryStatus	false
refType	string	true
vendor	string	false
displayName	string	false

DiscoveryResult

Name: DiscoveryResult

Table 328. DiscoveryResult

Property Name	Property Type	Required
refId	string	false
deviceType	DeviceType	false
serviceTag	string	false
ipAddress	string	false
iomCount	integer	false
serverCount	integer	false
model	string	false
serverType	string	false
discoverDeviceType	DiscoverDeviceType	false
firmwareDeviceInventories	[FirmwareDeviceInventory]	false
statusMessage	string	false
deviceRefId	string	false
healthState	string	false
healthStatusMsg	string	false
parentJobId	string	false
systemId	string	false

Property Name	Property Type	Required
config	string	false
jobId	string	false
status	DiscoveryStatus	false
refType	string	false
vendor	string	false

FirmwareDeviceInventory

Name: FirmwareDeviceInventory

Table 329. FirmwareDeviceInventory

Property Name	Property Type	Required
deviceInventory	ManagedDevice	false
discoveryResult	DiscoveryResult	false
fqdd	string	false
componentID	string	false
deviceId	string	false
vendorID	string	false
subdeviceId	string	false
subvendorID	string	false
ipaddress	string	false
firmwareComplianceComponents	FirmwareComplianceComponents	false
servicetag	string	false
parent_job_id	string	false
version	string	false
systemId	string	false
jobId	string	false
lastUpdateTime	string	false
componentType	string	false
name	string	false
id	string	false

DiscoveryRequest

Name: DiscoveryRequest

Table 330. DiscoveryRequest

Property Name	Property Type	Required
devices	[DiscoveredDevices]	false
statusMessage	string	false
totalCount	integer	false
discoveryRequestList	DeviceDiscoveryRequests	false
link	Link	false
status	DiscoveryStatus	false
id	string	true

ManagedDevice

Name: ManagedDevice

Table 331. ManagedDevice

Property Name	Property Type	Required
refId	string	true
deviceType	DeviceType	true
compliance	CompliantState	false
serviceTag	string	true
health	DeviceHealth	false
ipAddress	string	true
model	string	false
infraTemplateDate	GregorianCalendar	false
infraTemplateId	string	false
serverTemplateDate	GregorianCalendar	false
serverTemplateId	string	false
inventoryDate	GregorianCalendar	false
complianceCheckDate	GregorianCalendar	false
discoveredDate	GregorianCalendar	false
deviceGroupList	DeviceGroupList	false
detailLink	Link	false

Property Name	Property Type	Required
operatingSystem	string	false
numberOfCPUs	integer	false
memoryInGB	integer	false
credId	string	false
nics	integer	false
discoverDeviceType	DiscoverDeviceType	true
failuresCount	integer	false
chassisId	string	false
firmwareUpdateTime	string	false
firmwareDeviceInventories	[FirmwareDeviceInventory]	false
facts	string	false
statusMessage	DeviceState	false
manufacturer	string	false
cpuType	string	false
healthMessage	string	false
systemId	string	false
config	string	false
hostname	string	false
refType	string	false
state	DeviceState	false
displayName	string	false

ManagedDeviceList

Name: ManagedDeviceList

Table 332. ManagedDeviceList

Property Name	Property Type	Required
totalCount	integer	false
managedDevices	[ManagedDevice]	true

DeviceGroup

Name: DeviceGroup

Table 333. DeviceGroup

Property Name	Property Type	Required
createdDate	GregorianCalendar	false
updatedBy	string	false
updatedAt	GregorianCalendar	false
groupDescription	string	false
managedDeviceList	ManagedDeviceList	false
groupUserList	GroupUserList	false
groupSeqId	integer	false
groupName	string	true
link	Link	false
createdBy	string	false

DeviceGroupList

Name: DeviceGroupList

Table 334. DeviceGroupList

Property Name	Property Type	Required
deviceGroup	[DeviceGroup]	true
totalCount	integer	false

Appendix C - Ruby Code Usage

Several configuration options are available for the ASM Ruby module. In order to run the code examples, minimally one will need to set the username, password, and baseURI in the file AsmConfig.rb

```
require 'ASM'
require 'XMLPayload'

ASM::API::userName = 'admin'
ASM::API::password = 'abc123'

ASM::API::debug = false

# Globally enable automatic signing of all RestClient requests
#ASM::API::autoEnableAuth

ASM::API::baseURI = 'https://localhost/Api/V1'

# XMLPayload is a simple module to help process XML payloads
ASM::Payload.basePath = './xml_payloads'

# Set this to false if want to ignore problems with self-signed certs.
ASM::API::sslVerify = false

# hack to deal with self-signed certificates
module RestClient
  class Request
    def self.execute(args, & block)
      unless ASM::API::sslVerify
        args[:verify_ssl] = false
      end
      new(args).execute(& block)
    end
  end
end
```

The following Ruby code illustrates several methods for configuring ASM and signing requests.

```
require 'ASM'

ASM::API::userName = 'admin'
ASM::API::password = 'abc123'

ASM::API::baseURI = 'http://localhost/Api/V1'

# Fetch a service template by name
url = ASM::API::URI('/ServiceTemplate')
template_name = 'uploaded'
template_filter = "eq,name,#{template_name}"

# Enable auto signing for all RestClient calls
ASM::API::enableAutoAuth
result = RestClient.get url, {:params=>{:filter=>template_filter}}
```

```
# Disable auto signing for all RestClient calls
ASM::API::disableAutoAuth
result = RestClient.get url, {:params=>{:filter=>template_filter}}

# Sign a single request
ASM::API::sign do
  result = RestClient.get url, {:params=>{:filter=>template_filter}}
  result = RestClient.get url, {:params=>{:filter=>template_filter}}
end

# Logout to demo automatic login below
ASM::API::logout

# Sign a single request with supplied creds
ASM::API::sign('admin','admin') {
  RestClient.get url, {:params=>{:filter=>template_filter}}
}
```