



Cisco ACI Multi-Site REST API Configuration Guide, Release 1.0 and 1.1

First Published: 2017-08-10

Last Modified: 2017-12-20

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/go/trademarks>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2017 Cisco Systems, Inc. All rights reserved.



CONTENTS

PREFACE

Preface	ix
Audience	ix
Document Conventions	ix
Related Documentation	xi
Documentation Feedback	xii
Obtaining Documentation and Submitting a Service Request	xii

CHAPTER 1

New and Changed Information	1
New and Changed Information	1

CHAPTER 2

Using the REST API	3
About the REST API	3
Object Naming	3
Composing REST API Requests	3
User Roles and Authorization	4
Search for Policies by displayName	4

CHAPTER 3

Data Structure for REST Schemas	5
About Data Structure of the REST Schemas	5
Schema	5
Template	5
Name	6
displayName	6
tenantID	7
Reference Entities	7
ref	7

anpRef and vrfRef	7
Reference Generation	8
Local References	8
References from Another Schema	8
Application Profile	9
EPG	10
Bridge Domains	12
Contract	13
Filter	13
Site	14
External Authentication	16
Providers	16
Login Domains	17
Remote User Login	17
Backup	18
Auditing	18

CHAPTER 4**REST Examples 23**

REST API Examples	23
APIs for Local Authentication	24
User Login	24
User Logout	24
Refresh Token	25
Using Tokens on Authentication on APIs	25
Authenticating and Maintaining an API Session	26
Login to Authenticate and Maintain an API Session	26
Refresh to Authenticate and Maintain an API Session	27
APIs for External Authentication	28
About External Authentication	28
APIs for RADIUS Configurations	28
Create a New RADIUS Provider	28
Read RADIUS Provider Information	29
List Providers	29
Update an Existing RADIUS Provider	30

Delete a RADIUS Provider	31
APIs for TACACS+ Configurations	31
Create a New TACACS Provider	31
Read TACACS Provider Information	32
List TACACS Providers	32
Update an Existing TACAS Provider	33
Delete a TACACS Provider	34
APIs for Login Domain	34
Create a New Login Domain	34
List all Active Login Domains	35
List Login Domains	35
Modify an Existing Login Domain	36
Modify an Existing Login Domain	37
APIs for Remote User Login	37
Remote User Login	37
APIs for Labels	38
Create a New Label	38
Read Label Information	39
Update a Label	39
Delete Labels	40
List Labels	40
APIs for Roles	41
Read Role Information	41
List Roles	41
APIs for Users	42
Create a New User	42
Reset Password for an Existing User	43
Read User Information	44
Update an Existing User	44
Delete a User	45
List Users	46
List Roles That May Be Associated With Users	47
Retrieve Roles Associated With a User	47
Retrieve Permissions Associated With a User	48

APIs for Fabric Connectivity	49
Get Fabric Connectivity for a Site	49
Update Fabric Connectivity	57
Refresh Fabric Connectivity for a Site	59
Fabric Leaf Switches and Physical Interfaces	60
Get Nodes	60
Get Physical Interfaces Details	60
APIs for Tenants	62
Create a Tenant	62
Read Tenant Information	63
Update an Existing Tenant	63
Delete a Tenant	64
List Tenants	64
List Sites Associated with a Tenant	66
List Users Associated with a Tenant	66
List Schemas Associated with a Tenant	68
List Schemas Associated with all Tenants	68
Create a New Tenant Network Mapping	69
Get Existing Tenant Network Mappings	70
APIs for Schemas	70
Create a New Schema	70
Read a Schema	107
Schema Health Summary	109
Schema Health Sync	110
Delete Schema Health Information	112
Retrieve Policy States	112
Retrieve References	113
Retrieving Schema Health and Faults	114
Update an Existing Schema	119
Delete a Schema	122
List Schema Identities	123
List Sites Associated to a Schema	125
List Tenants that Can Be Associated With a Schema	125
List All Tenants Associated to Schemas	126

List Sites That Can Be Associated With a Schema	127
List All Sites Associated to Schemas	128
List Tenants That Can be Associated With a Schema	129
List All Sites Associated to Schemas	130
APIs for Import	131
Import a Tenant from APIC	131
APIs for Sites	137
Create a New Site	137
Read Site Information	138
Read Site Information That Contains Health Faults	138
Update an Existing Site	140
Delete a Site	140
List Sites	141
List Labels	141
APIs for Template	143
Execute a Template	143
Undeploy a Template	143
APIs for Entities	143
Login Domains	143
List L3Outs	144
APIs for Platform	145
Read Node Information	145
Download Troubleshooting Report	146
Expected HTTP Response Code Examples	146
APIs for Audit	147
About Auditing	147
Filter Audit Records	147
Paginate Audit Records	151
About Login Domains Auditing	152
List or Filter Login Domains Audit Records	153
About Users Auditing	154
List or Filter User Audit Records	154
About Providers Auditing	155
List or Filter Provider Audit Records	155

List or Filter RADIUS Provider Audit Records	156
List or Filter TACACS Provider Audit Records	158
List or Filter Audit Records for Tenants	159
Find Users Associated with Tenant Audit Records	162
List or Filter Audit Records for Schemas	163
Find Users Associated with Schema Audit Records	173
List or Filter Audit Records for Fabric-Connectivity	173
Find Users Associated with Fabric-Connectivity Audit Records	174
APIs for Backups	174
About Backup	174
Create a New Backup	174
List Backups	175
Download a Backup	176
Upload a Backup	177



Preface

This preface includes the following sections:

- [Audience, on page ix](#)
- [Document Conventions, on page ix](#)
- [Related Documentation, on page xi](#)
- [Documentation Feedback, on page xii](#)
- [Obtaining Documentation and Submitting a Service Request, on page xii](#)

Audience

This guide is intended primarily for data center administrators with responsibilities and expertise in one or more of the following:

- Virtual machine installation and administration
- Server administration
- Switch and network administration

Document Conventions

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
<i>Italic</i>	Italic text indicates arguments for which the user supplies the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x y]	Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice.
{x y}	Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice.

Convention	Description
[x {y z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.
<i>variable</i>	Indicates a variable for which you supply values, in context where italics cannot be used.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

Examples use the following conventions:

Convention	Description
<code>screen font</code>	Terminal sessions and information the switch displays are in screen font.
boldface screen font	Information you must enter is in boldface screen font.
<i>italic screen font</i>	Arguments for which you supply values are in italic screen font.
< >	Nonprinting characters, such as passwords, are in angle brackets.
[]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

This document uses the following conventions:



Note

Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.



Caution

Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.



Warning

IMPORTANT SAFETY INSTRUCTIONS

This warning symbol means danger. You are in a situation that could cause bodily injury. Before you work on any equipment, be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents. Use the statement number provided at the end of each warning to locate its translation in the translated safety warnings that accompanied this device.

SAVE THESE INSTRUCTIONS

Related Documentation

Application Policy Infrastructure Controller (APIC) Documentation

The following companion guides provide documentation for APIC:

- *Cisco APIC Getting Started Guide*
- *Cisco APIC Basic Configuration Guide*
- *Cisco ACI Fundamentals*
- *Cisco APIC Layer 2 Networking Configuration Guide*
- *Cisco APIC Layer 3 Networking Configuration Guide*
- *Cisco APIC NX-OS Style Command-Line Interface Configuration Guide*
- *Cisco APIC REST API Configuration Guide*
- *Cisco APIC Layer 4 to Layer 7 Services Deployment Guide*
- *Cisco ACI Virtualization Guide*
- *Cisco Application Centric Infrastructure Best Practices Guide*

All these documents are available at the following URL: <http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html>

Cisco Application Centric Infrastructure (ACI) Documentation

The broader ACI documentation is available at the following URL: <http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html>.

Cisco Application Centric Infrastructure (ACI) Simulator Documentation

The Cisco ACI Simulator documentation is available at <http://www.cisco.com/c/en/us/support/cloud-systems-management/application-centric-infrastructure-simulator/tsd-products-support-series-home.html>.

Cisco Nexus 9000 Series Switches Documentation

The Cisco Nexus 9000 Series Switches documentation is available at <http://www.cisco.com/c/en/us/support/switches/nexus-9000-series-switches/tsd-products-support-series-home.html>.

Cisco Application Virtual Switch Documentation

The Cisco Application Virtual Switch (AVS) documentation is available at <http://www.cisco.com/c/en/us/support/switches/application-virtual-switch/tsd-products-support-series-home.html>.

Cisco Application Centric Infrastructure (ACI) Integration with OpenStack Documentation

Cisco ACI integration with OpenStack documentation is available at <http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html>.

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to apic-docfeedback@cisco.com. We appreciate your feedback.

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see *What's New in Cisco Product Documentation* at: <http://www.cisco.com/c/en/us/td/docs/general/whatsnew/whatsnew.html>

Subscribe to *What's New in Cisco Product Documentation*, which lists all new and revised Cisco technical documentation as an RSS feed and delivers content directly to your desktop using a reader application. The RSS feeds are a free service.



CHAPTER 1

New and Changed Information

This chapter contains the following sections:

- [New and Changed Information, on page 1](#)

New and Changed Information

The following table provides an overview of the significant changes to the organization and features in this guide up to this current release. The table does not provide an exhaustive list of all changes made to the guide.

Table 1: New Features and Changed Behavior in Cisco ACI Multi-Site Release 1.1(1e)

Cisco ACI Multi-Site Release	Description	Where Documented
1.1(1e)	Data structure information related to the following areas were added: <ul style="list-style-type: none">• Auditing• External authentication• Backup	<i>Data Structure for REST Schemas</i>
1.1(1e)	In this release, some new APIs are introduced. Examples for those APIs are provided.	<ul style="list-style-type: none">• Audit examples• External authentication examples• Backup examples

Cisco ACI Multi-Site Release	Description	Where Documented
1.1(1e)	<p>In this release, some API examples were modified or added.</p> <p>Note For the examples in this guide, if an example does not specify a release number in the heading, it is applicable to release 1.0(1), release 1.0(2), and release 1.1(1).</p>	<p>The following areas have new examples for this release:</p> <ul style="list-style-type: none"> • Create a New Schema

Table 2: New Features and Changed Behavior in Cisco ACI Multi-Site Release 1.0(2)

Cisco ACI Multi-Site Release	Description	Where Documented
1.0(2)	<p>In this release, some API examples were modified or added.</p> <p>Note For the examples in this guide, if an example does not specify a release number in the heading, it is applicable to release 1.0(1) and release 1.0(2).</p>	<ul style="list-style-type: none"> • Schema examples • Fabric Connectivity Status examples

Table 3: New Features and Changed Behavior in Cisco ACI Multi-Site Release 1.0(1)

Cisco ACI Multi-Site Release	Description	Where Documented
1.0(1)	This document was introduced.	--



CHAPTER 2

Using the REST API

This chapter contains the following sections:

- [About the REST API, on page 3](#)

About the REST API

The Cisco ACI Multi-Site REST API is a programmatic interface that uses REST architecture. The API accepts and returns HTTP (not enabled by default) or HTTPS messages that contain JavaScript Object Notation (JSON) documents. You can use any programming language to generate the messages and the JSON documents that contain the API methods.

Standard REST methods are supported on the API, which includes POST, GET, PUT, and DELETE operations through HTTP. The PUT methods are idempotent, meaning that there is no additional effect if they are called more than once with the same input parameters. The GET method is nullipotent, meaning that it can be called zero or more times without making any changes (or that it is a read-only operation).

Payloads to and from the REST interface must be encapsulated through JSON encoding. Containment is defined by creating child attributes.

Object Naming

Resources are accessed by supplying the database identifier.

Composing REST API Requests

Read and Write Operations

```
/api/v1/tenants/<id>
```

v1 is the first version of the APIs.

Read Operations

After the object payloads are properly encoded as JSON, they can be used in create, read, update, or delete operations on the REST API.

Because the REST API is HTTP-based, defining the URI to access a certain resource type is important. The first two sections of the request URI simply define the protocol. Next in the request URI is the literal string `/api`, indicating that the API is invoked.

Write Operations

POST = create

PUT = update

If an object does not exist, it is created. If an object exists, it is updated to reflect any changes between its existing state and desired state.

Create and update operations use the same syntax as read operations. The create or update operation must target a specific entity. The payload of the POST and PUT operations contains JSON encoded data.

User Roles and Authorization

The API supports user-specific authorization based on the role of a user. A user can be associated with specific roles for access based on their function. While the admin can perform all operations, REST endpoints can be restricted based on a user's role. For more information, see the *Cisco ACI Multi-Site Configuration Guide*.

Search for Policies by displayName

GET

```
GET /api/v1/schemas/search-policies?type=:type&tenant-id=:tenantId&query=:query
```

Examples:

```
/api/v1/schemas/search-policies?type=:type&tenant-id=:tenantId&query=:query  
/api/v1/schemas/search-policies?type=vrf&tenant-id=T1&query=vrf  
/api/v1/schemas/search-policies?type=contract&query=contra
```

Searchable Policy Types are as follows:

- VRF
- BD
- Contract
- Filter

You must follow these guidelines when using searchable Policy Types:

- TenantId is mandatory for the following policy types: VRF, BD, and Filter.
- TenantId is not required and is not used for Policy type Contracts.
- Query must be three or more characters (in other cases, a BadRequest will be returned).
- This API may return 100 records.



CHAPTER 3

Data Structure for REST Schemas

This chapter contains the following sections:

- [About Data Structure of the REST Schemas, on page 5](#)

About Data Structure of the REST Schemas

This section describes how the REST Schemas are structured in Cisco ACI Multi-Site using JSON. The data structure is consistent across the Multi-Site. The top-level entities in a Schema are templates. The templates contain smaller entities such as application profiles (anps), bridge domains (bds), virtual routing and forwarding (vrf) and such. These entities are also composed from other smaller structures. A few high-level entities or concepts are described in this section.

Schema

A schema is a container for templates. When a site is chosen, it automatically gets added to the schema, and it maps to the template in that schema. The high-level attributes of a schema are displayed in the following example API snippet:

```
{
  "id": "583c7c482501002501061985",
  "displayName": "Schema 1",
  "templates": [ ],
  "sites": [ ],
}
```

Template

A template is the detailed structure that contains policies that the user wants to push to Cisco APIC. Templates exist in the context of a Schema and not on their own. A user can define the policy of a new template as desired. If there are multiple templates, the user can choose a template from which to inherit the policy. The high-level attributes of a template are as follows:

- name
- displayName
- tenantId

- anps
- vrfs
- bds
- contracts
- filters

The key structures inside templates are as follows:

- anps (application network profiles)
- vrfs
- bds
- contracts
- filters

When an entity name is pluralized (such as vrfs, bds), it means that an array of that sub-entity is supported under its parent entity, and you can have multiple entries of the entity present in the template.

The high-level attributes of a template are displayed in the following example API snippet:

```
{
  "name": "templatel",
  "displayName": "Template 1",
  "tenantId": "592c7c482501002501061578",
  "anps": [ ],
  "vrfs": [ ],
  "bds": [ ],
  "contracts": [ ],
  "filters": [ ],
}
```

Name

The name field is for entities that have a name. This is the name that is mapped to the names of policies in Cisco APIC. These names are restrictive and are determined based on restrictions placed by Cisco APIC. The following is an example of a Name API snippet:

```
"name": "templatel",
```

displayName

The displayName field contains a user-friendly name that supports spaces and special characters in the displayName. The displayName will be visible in the Cisco ACI Multi-Site UI, and it is generally editable in Multi-Site.

The following is an example of a displayName API snippet:

```
"displayName": "Template 1",
```

tenantID

The tenantID is an association with tenants.

The following is an example of a tenantID API snippet:

```
"tenantId": "592c7c482501002501061578",
```

Reference Entities

ref

A ref field for a policy is optional. However, to reference entities in the system that you create, you must specify the reference in these relations. Users must specify references in order to provide relationships between policies.

anpRef and vrfRef

The anpRef identifies the application profile (anp), for which it is a sub-attribute in the system. The anpRef is an identifier for anp. The anpRef sub-attribute is typically used in situations where other entities have to be referred within a particular entity. For policies within a template, a ref field is consistently generated. A user does not specify the value of the anpRef sub-attribute. The system generates an anpRef for each and every policy.

Providing an input as anpRef is not a requirement. However, if you define a policy within a particular Schema, and you want to reference it immediately, you can provide the anpRef as input.

The following example displays an anpRef in the anps structure:

```
"anps": [
  {
    "name": "anp1",
    "displayName": "Anp 1",
    "anpRef": "/schemas/583c7c482501002501061985/templates/template1/anps/anp1",
    "epgs": [ <-> ]
  }
],
```

The following example displays a VRF reference:

```
"vrfs": [
  {
    "name": "vrf1",
    "displayName": "VRF 1",
    "vrfRef": "/schemas/583c7c482501002501061985/templates/template1/vrfs/vrf1"
  }
]
```

Reference Generation

References are generated for each and every policy. A reference is generated on the first save. However, if you define a policy within the Schema, and you want to reference it right away, you can specify a reference without having to wait for it to be generated.

In a JSON example, the left side data shows what the user provides, such as local references. After an operation concludes and the data persists, the system provides the data that is displayed on the right side. For example, after persistence, the system generates the prefix for a reference, and it is displayed on the right side as a system-generated reference.

Local References

In some cases, the reference that is supplied as input to the post operation, does not contain the Schema ID (as system generated references contain). A local reference means that an entity within this Schema is referenced by the path of that entity. In such a case, the ID is not used as a reference, and the reference does not contain Schema ID.

For example, you create a contract, and you want to reference the contract within the EPG immediately. The system will not yet have generated a reference for the contract, so the generated 'ref' is not available. However, you can still ensure the relationship by referencing the contract to the path of an entity that already exists in the system.

Once it persists, the system makes a local reference an absolute reference. The system prepends the Schema ID even though it is a local reference that is available in the same Schema, and thereby fully populates the reference.

The following is an example of a local contract reference API snippet:

```
{
  "name": "epg1",
  "displayName": "EPG 2",
  "contractRelationships":
  {
    "contractRef": "contractRef": "/templates/templatel/contracts/contract1",
    "relationshipType": "consumer"
  }
},
```

References from Another Schema

It is possible to associate an entity (for example, a provider contract) from another Schema. You can specify the contract reference by prepending it with an ID contained in an absolute reference from another Schema. This references another policy in another Schema.

The following is an API snippet example of a reference from another Schema:

```
{
  "name": "epg1",
  "displayName": "EPG 2",
  "contractRelationships":
  {
    "contractRef":
    "/schemas/590ca1811f000062006eef23/templates/templatel/contracts/contract2",
    "relationshipType": "provider"
  }
},
```



Note The attributes for the **epg** > **contractRelationships** > **relationshipType** are provider and consumer.

Application Profile

The application profile is the **anp** entity. This contains **epgs** as a sub-entity.

The following is an example application profile entity API snippet:

```
"anps": [
  {
    "name": "anp1",
    "displayName": "Anp 1",
    "anpRef": "/schemas/583c7c482501002501061985/templates/template1/anps/anp1",
    "epgs": [
      {
        "name": "epg1",
        "displayName": "EPG 1",
        "epgRef":
"/schemas/583c7c482501002501061985/templates/template1/anps/anp1/epgs/epg1",
        "contractRelationships": [
          {
            "contractRef":
"/schemas/583c7c482501002501061985/templates/template1/contracts/contract1",
            "relationshipType": "provider"
          }
        ],
        "subnets": [ {
          "ip": "192.168.132.1/24",
          "description": "Subnet for IT department",
          "scope" : "private"
          "shared": true
        }, {
          "ip": "192.168.133.1/24",
          "description": "Subnet for sales department",
          "scope" : "public"
          "shared": false
        }
      ]
    },
    "uSegEpg" : true,
    "uSegAttrs" : [ {
      "name" : "IP1",
      "displayName" : "...",
      "description" : "... ",
      "type" : "ip",
      "fvSubnet" : true,
      "value" : "0.0.0.0"
    }, {
      "name" : "IP2",
      "description" : "... ",
      "type" : "ip",
      "fvSubnet" : false,
      "value" : "192.168.132.1"
    }, {
      "name" : "MAC",
      "description" : "...",
      "type" : "mac",
      "value" : "aa:bb:cc:dd:ee:ff"
    }, {
      "name" : "VM Name",
```

```

        "description" : "...",
        "type" : "vm-name",
        "operator" : "equals | contains | startsWith | endsWith",
        "value" : "..."
    }, {
        "name" : "VM Data Center",
        "description" : "... (optional)",
        "type" : "rootContName",
        "operator" : "...",
        "value" : "..."
    }, {
        "name" : "VM Hypervisor Identifier",
        "description" : "... (optional)",
        "type" : "hv",
        "operator" : "...",
        "value" : "..."
    }, {
        "name" : "VmOperatingSystem",
        "description" : "...",
        "type" : "guest-os",
        "operator" : "...",
        "value" : "..."
    }, {
        "name" : "VM Tag",
        "description" : "... ",
        "type" : "tag",
        "operator" : "...",
        "category" : "...",
        "value" : "..."
    }, {
        "name" : "Vm Identifier",
        "description" : "... ",
        "type" : "vm",
        "operator" : "...",
        "value" : "..."
    }, {
        "name" : "Vm Vmm Domain",
        "description" : "...",
        "type" : "domain",
        "operator" : "...",
        "value" : "..."
    }, {
        "name" : "Vm Vnic Dn",
        "description" : "... (optional)",
        "type" : "vnic",
        "operator" : "...",
        "value" : "..."
    } ],
    "bdRef" : "/schemas/583c7c482501002501061985/templates/template1/bds/bd1"
},
]
}
],
],

```

EPG

One EPG can be related to multiple contracts through provider or consumer contract relationships. EPGs are related to contracts and bridge domains (bds).

The following is an example of an epg API snippet:

```
"epgs": [
```

```

{
  "name": "epg1",
  "displayName": "EPG 1",
  "epgRef":
"/schemas/583c7c482501002501061985/templates/template1/anps/anp1/epgs/epg1",
  "contractRelationships": [
    {
      "contractRef":
"/schemas/583c7c482501002501061985/templates/template1/contracts/contract1",
      "relationshipType": "provider|consumer"
    }
  ],
  "subnets": [ {
    "ip": "192.168.132.1/24",
    "description": "Subnet for IT department",
    "scope" : "private"
    "shared": true
  }, {
    "ip": "192.168.133.1/24",
    "description": "Subnet for sales department",
    "scope" : "public"
    "shared": false
  } ],
  "uSegEpg" : true,
  "uSegAttrs" : [ {
    "name" : "IP1",
    "displayName" : "...",
    "description" : "... ",
    "type" : "ip",
    "fvSubnet" : true,
    "value" : "0.0.0.0"
  }, {
    "name" : "IP2",
    "description" : "... ",
    "type" : "ip",
    "fvSubnet" : false,
    "value" : "192.168.132.1"
  }, {
    "name" : "MAC",
    "description" : "...",
    "type" : "mac",
    "value" : "aa:bb:cc:dd:ee:ff"
  }, {
    "name" : "VM Name",
    "description" : "...",
    "type" : "vm-name",
    "operator" : "equals | contains | startsWith | endsWith",
    "value" : "..."
  }, {
    "name" : "VM Data Center",
    "description" : "... (optional)",
    "type" : "rootContName",
    "operator" : "...",
    "value" : "..."
  }, {
    "name" : "VM Hypervisor Identifier",
    "description" : "... (optional)",
    "type" : "hv",
    "operator" : "...",
    "value" : "..."
  }, {
    "name" : "VmOperatingSystem",
    "description" : "...",
    "type" : "guest-os",

```

```

        "operator" : "...",
        "value" : "...",
    }, {
        "name" : "VM Tag",
        "description" : "... ",
        "type" : "tag",
        "operator" : "...",
        "category" : "...",
        "value" : "...",
    }, {
        "name" : "Vm Identifier",
        "description" : "... ",
        "type" : "vm",
        "operator" : "...",
        "value" : "...",
    }, {
        "name" : "Vm Vmm Domain",
        "description" : "...",
        "type" : "domain",
        "operator" : "...",
        "value" : "...",
    }, {
        "name" : "Vm Vnic Dn",
        "description" : "... (optional)",
        "type" : "vnic",
        "operator" : "...",
        "value" : "...",
    } ],
    "bdRef": "/schemas/583c7c482501002501061985/templates/template1/bds/bd1"
},
]

```



Note The attributes for **VM Name** > **uSegAttrs** > **operator** are "equals", "contain", "startsWith", "endsWith".

Bridge Domains

Bridge Domains (bds) have attributes such as multicast flooding.

The following is an example of a bridge domain API snippet:

```

"bds": [
  {
    "name": "bd1",
    "displayName": "BD 1",
    "bdRef": "/schemas/583c7c482501002501061985/templates/template1/bds/bd1",
    "l3UnknownMulticastFlooding": "opt-flood",
    "intersiteBumTrafficAllow": true,
    "multiDestinationFlooding": "encap-flood",
    "l2UnknownUnicast": "flood",
    "l2Stretch": true,
    "subnets": [ ... ],
    "vrfRef": "/schemas/583c7c482501002501061985/templates/template1/vrfs/vrf1"
  }
],

```


**Note**

- The attributes for **bds > I3UnknownMulticastFlooding** are "flood", "opt-flood".
- The attributes for **bds > multiDestinationFlooding** are "bd-flood", "drop", "encap-flood".
- The attributes for **bds > I2UnknownUnicast** are "flood" and "proxy".

Contract

The following is an example of a contract API snippet:

```
"contracts": [
  {
    "name": "contract1",
    "displayName": "Contract 1",
    "filterRelationships": [
      {
        "filterRef": "/templates/templatel/filters/filter1",
        "directives": [
          "log|none"
        ]
      }
    ],
    "scope": "global|tenant|context|application-profile"
  },
  {
    "name": "contract2",
    "displayName": "Contract 2",
    "scope": "global|tenant|context|application-profile"
  }
],
```

**Note**

The attributes for **contracts > scope** are "global", "tenant", "context", "application-profile".

Filter

The following is an example of a filter API snippet:

```
"filters": [
  {
    "name": "filter1",
    "displayName": "Filter 1",
    "description": "",
    "entries": [
      {
        "name": "filterEntry11",
        "displayName": "Filter Entry 11",
        "description": "",
        "etherType": "ip",
        "ipProtocol": "icmp"
      },
      {
        "name": "filterEntry12",
```

```

        "displayName": "Filter Entry 12",
        "description": "",
        "etherType": "ip",
        "ipProtocol": "udp",
        "matchOnlyFragments": false,
        "sourceFrom": "dns",
        "sourceTo": "http",
        "destinationFrom": "dns",
        "destinationTo": "80"
      }
    ]
  },

```

the following is an example of attributes for a filter:

```

{
  "name": "filter2",
  "displayName": "Filter 2",
  "description": "",
  "entries": [
    {
      "name": "filterEntry21",
      "displayName": "Filter Entry 21",
      "description": "",
      "etherType": "arp|fcoe|ip|mac_security|mpls_ucast|trill|unspecified",
      "arpFlag": "request|reply|unspecified",
      "ipProtocol":
"eigrp|egp|icmp|icmpv6|igmp|igp|l2tp|ospfigp|pim|tcp|udp|unspecified",
      "matchOnlyFragments": false,
      "stateful": false,
      "sourceFrom": "dns|ftp-data|http|https|pop3|rtsp|smtp|unspecified|[0..65535]",
      "sourceTo": "dns|ftp-data|http|https|pop3|rtsp|smtp|unspecified|[0..65535]",
      "destinationFrom":
"dns|ftp-data|http|https|pop3|rtsp|smtp|unspecified|[0..65535]",
      "destinationTo":
"dns|ftp-data|http|https|pop3|rtsp|smtp|unspecified|[0..65535]",
      "tcpSessionRules": [
        "acknowledgement|established|finish|synchronize|reset|unspecified",
        "acknowledgement|established|finish|synchronize|reset|unspecified"
      ]
    }
  ]
}

```

Site

Each site is associated with a specific template. The policies from specific templates are pushed to sites based on the site-to-template association.

The `siteId` is an internal identification pointing to a site defined in Cisco ACI Multi-Site.

Some attributes are site-specific attributes and they are defined for a specific site. For example, site-specific attributes for entities in a template.

Site-specific attributes are as follows:

- epgs have domainAssociations and staticPorts
- bds have subnets
- contracts have serviceGraphTemplate

The following is an example of a site API snippet:

```

"sites": [
  {
    "siteId": "58202f7066e6e10001c41236",
    "templateName": "template1",
    "bds": [
      {
        "bdRef": "/templates/template1/bds/bd1",
        "subnets": [ ... ]
      }
    ]
  },
  {
    "siteId": "69202f7066e6e10001c41247",
    "templateName": "template1",
    "anps": [
      {
        "anpRef": "/templates/template1/anps/anp1",
        "epgs": [
          {
            "epgRef": "/templates/template1/anps/anp1/epgs/epg1",
            "domainAssociations": [
              {
                "dn": "uni/vmmp-VMware/dom-TestVMMDomain-DoNotDelete",
                "domainType": "vmmDomain",
                "deployImmediacy": "On Demand|Immediate",
                "resolutionImmediacy": "On Demand|Immediate|Pre-provision",
                "allowMicroSegmentation": true,
                "allowPromiscuous": true,
                "macChanges": false,
                "allowForgedTransmits": true,
                "delimiter": "delimiter"
              },
              {
                "dn": "uni/phys-testPhysicalDomain-DoNotDelete",
                "domainType": "physicalDomain",
                "deployImmediacy": "On Demand|Immediate",
                "resolutionImmediacy": "On Demand|Immediate|Pre-provision"
              },
              {
                "dn": "uni/l2dom-TestL2Domain-DoNotDelete",
                "domainType": "l2ExtDomain",
                "deployImmediacy": "On Demand|Immediate",
                "resolutionImmediacy": "On Demand|Immediate|Pre-provision"
              },
              {
                "dn": "uni/l3dom-TestL3Domain-DoNotDelete",
                "domainType": "l3ExtDomain",
                "deployImmediacy": "On Demand|Immediate",
                "resolutionImmediacy": "On Demand|Immediate|Pre-provision"
              },
              {
                "dn": "uni/fc-TestFibreChannelDomain",
                "domainType": "fibreChannelDomain",
                "deployImmediacy": "On Demand|Immediate",
                "resolutionImmediacy": "On Demand|Immediate|Pre-provision"
              }
            ],
            "staticPorts": [
              {
                "type": "port",
                "path": "topology/pod-1/paths-101/pathep[eth1/11]",
                "portEncapVlan": 10,
                "microSegVlan": 8,

```


}



Note The values for the protocol field are as follows:

- CHAP
- PAP

Login Domains

End users can specify a login domain when they log in to the Multi-Site user interface.

The realm field displays the possible provider values. Currently, they can be radius or tacacs.

The following is an example of a login domain API snippet:

```
{
  "domains": [
    {
      "id": "59d6d0398d0000d2009eb2c4",
      "name": "EngineeringGroup",
      "description": "Used for RADIUS Auth Login",
      "realm": "radius",
      "status": "active",
      "providerAssociations": []
    },
    {
      "id": "59d6cfb58d0000bd009eb2c3",
      "name": "VM-TACACS",
      "description": "Used for Tacacs Auth Login",
      "realm": "tacacs",
      "status": "active",
      "providerAssociations": []
    }
  ]
}
```

Remote User Login

Remote user login is a POST API login endpoint. You can create and read remote user login domains with these APIs. Remote user login APIs do not require the authorization header to be set.

The domainID parameter value is used to login after a domain is defined. Defining a domainID is an optional field. For example, the domainID parameter is not specified when the default domain is Local.

The following is an example of a remote user login API snippet:

```
{
  "username": "bob",
  "password": "welcome!",
  "domainId": "59d6d0398d0000d2009eb2c4",
}
```

Backup

Starting with Cisco ACI Multi-Site release 1.1(1), APIs for backup are supported.

The data structure represents what you require to backup your system. It has fields such as ID, size, timestamps, description etc.

The following is an example of a backup snippet:

```
HTTP/1.1 201
Location: /api/v1/backups/20171116221404
{
  "id": "20171116221404",
  "sizeKB": 18.44,
  "timestamp": "2017-11-16T22:14:04",
  "metadata": {
    "description": "test-description",
    "name": "Check_20171116221404"
  }
}
```

Auditing

Starting with Cisco ACI Multi-Site release 1.1(1), APIs for auditing are supported.

Auditing in the API is exposed through the audit-records endpoint. Through this endpoint, APIs to create, read, filter, and paginate records are available.

Filter Audit Records by **type** are as follows:

Module/Sub-module	Types
Tenants	<ul style="list-style-type: none"> • tenant • tenant-site • tenant-user

Module/Sub-module	Types
Schemas	<ul style="list-style-type: none"> • schema • template • schema-site • anp • bd • bd-vrf • epg • epg-bd • epg-contract • external-epg • external-epg-contract • contract • contract-filter • external-epg • filter • vrf
Sites	<ul style="list-style-type: none"> • site • site-label
Fabric Connectivity	<ul style="list-style-type: none"> • fabric-connectivity • infra-bgp-config • infra-ospf-policy • infra-site • infra-pod • infra-node • infra-port
Users	<ul style="list-style-type: none"> • user • user-role • user-domain
Authentication	<ul style="list-style-type: none"> • user-auth

Module/Sub-module	Types
Login Domains	<ul style="list-style-type: none"> • domain • domain-provider
Providers	<ul style="list-style-type: none"> • radius • tacacs
(not module specific)	attribute

The events that are available in auditing APIs are as follows:

- created
- deleted
- updated
- added
- removed
- associated
- disassociated
- deployed
- undeployed
- downloaded
- uploaded
- restored
- logged-in
- login-failed
- logged-out
- login-failed

The following is an example of an auditing API snippet:

```
HTTP/1.1 200
{
  "auditRecords": [
    {
      "id": "59b5e11f1a0100270319b05e",
      "type": "tenant",
      "key": "59b5e11f3400004900c0ea97",
      "name": "Tenant 001",
      "event": "created",
      "user": {
        "userId": "0000ffff000000000000000020",
        "username": "Admin User",
```



```

        "firstName": "Admin",
        "lastName": "User",
        "domainId": "0000ffff0000000000000090",
        "domainName": "Local"
    },
    "timestamp": "2017-09-10T01:04:31.701"
  },
  {
    "id": "59b5e11f1a0100270319b05e",
    "type": "tenant",
    "key": "59b5e11f3400004900c0ea97",
    "name": "Tenant 001",
    "event": "updated",
    "user": {
      "userId": "0000ffff0000000000000020",
      "username": "Admin User",
      "firstName": "Admin",
      "lastName": "User",
      "domainId": "0000ffff0000000000000090",
      "domainName": "Local"
    },
    "timestamp": "2017-09-10T01:04:35.501"
  }
]
}

```

The data structure from a schema auditing perspective is as follows: `schema/:id/audit-records/`. The context of a site is applicable when we examine site local attributes from a schema auditing perspective.

The following is an example of a schema auditing API snippet:

```

HTTP/1.1 200
{
  "auditRecords": [
    {
      "id": "59ef87743a000022069baf60",
      "type": "schema",
      "key": "59ef87743800008e06881fe5",
      "name": "Test Schema",
      "event": "created",
      "user": {
        "userId": "0000ffff0000000000000020",
        "username": "admin",
        "firstName": "Admin",
        "lastName": "User",
        "domainId": "0000ffff0000000000000090",
        "domainName": "Local"
      },
      "timestamp": "2017-10-24T18:33:24.584"
    },
  ],
}

```

Tenant audit records have a specific usage where `type = attribute`. The name of the attribute is provided at the end of the string, for example `description` is provided in the `key` attribute.

Another specific usage is the following values applied to attributes: `oldValue` and `newValue`.

The following is an example of a tenant auditing API snippet:

```

HTTP/1.1 200
{

```

```
"auditRecords": [  
  {  
    "id": "59db14881e0000f00071f40e",  
    "type": "tenant",  
    "key": "59db147b2500000b01c90429",  
    "name": "Tenant 001",  
    "event": "updated",  
    "user": {  
      "userId": "0000ffff0000000000000020",  
      "username": "admin",  
      "firstName": "Admin",  
      "lastName": "User",  
      "domainId": "0000ffff0000000000000090",  
      "domainName": "Local"  
    },  
    "timestamp": "2017-10-09T06:17:44.141"  
  },  
]  
}
```



CHAPTER 4

REST Examples

This chapter contains the following sections:

- [REST API Examples](#) , on page 23
- [APIs for Local Authentication](#), on page 24
- [APIs for External Authentication](#), on page 28
- [APIs for Labels](#), on page 38
- [APIs for Roles](#), on page 41
- [APIs for Users](#), on page 42
- [APIs for Fabric Connectivity](#), on page 49
- [Fabric Leaf Switches and Physical Interfaces](#), on page 60
- [APIs for Tenants](#), on page 62
- [APIs for Schemas](#), on page 70
- [APIs for Import](#), on page 131
- [APIs for Sites](#), on page 137
- [APIs for Template](#), on page 143
- [APIs for Entities](#), on page 143
- [APIs for Platform](#), on page 145
- [Expected HTTP Response Code Examples](#), on page 146
- [APIs for Audit](#), on page 147
- [APIs for Backups](#), on page 174

REST API Examples



Note The word Schema is alternately known as Profile in this guide.

For the examples in this guide, if an example does not specify a release number in the heading, it is applicable to release 1.0(1) and release 1.0(2).

APIs for Local Authentication

User Login

The following is an example of how a user can login.

POST:

POST `https://<host>:<port>/api/v1/auth/login`

Request:

```
{
  "username": "bob",
  "password": "welcome"
}
```

Response:

```
HTTP/1.1 201 (Created) -> Valid user credentials
{
  "userId": "592717d22000008b01db2749",
  "token":
  "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZi5fQWpfcqzdo-g-OORGUmMzCVspIL7rYHjKJH0srX7Tl3wA",

  "permissions": [
    "view-sites",
    "manage-sites",
    "view-tenants",
    "manage-tenants"
  ]
}
```

```
HTTP/1.1 401(Unauthorized) -> Invalid user credentials
{
  "code": 401,
  "message": "Invalid user credentials"
}
```



Note

- This is the only API in Cisco ACI Multi-Site that does not require the Authorization header to be set.
- The token in the above example has been trimmed for readability.

User Logout

The following is an example of how a user can logout.

DELETE:

DELETE `https://<host>:<port>/api/v1/auth/logout`

Request:


```

    "userAssociations": [
      {
        "userId": "0000ffff0000000000000020"
      }
    ],
    "description": "Common tenant for use with all other tenants"
  }
]
}

```

APIs for External Authentication

About External Authentication

Starting with Cisco ACI Multi-Site, release 1.1(1), the APIs for external authentication are supported. Therefore, the external authentication examples are applicable starting with release 1.1(1) only.

APIs for RADIUS Configurations

Create a New RADIUS Provider

The following is an example of how to create a new RADIUS provider.

POST:

POST https://<host>:<port>/api/v1/auth/providers/radius

Request:

```

{
  "host": "10.23.237.168",
  "description": "ACS RADIUS Provider",
  "sharedSecret": "ins3965!",
  "port": 1812,
  "protocol": "PAP",
  "timeoutInSeconds": 10,
  "retries":5
}

```

Response:

```

HTTP/1.1 201
Location: /api/v1/auth/providers/radius/59cc163c1f00005400ab4ffd
{
  "id": "59cc163c1f00005400ab4ffd",
  "host": "10.23.237.168",
  "description": "Tacacs Provider - 1",
  "sharedSecret": "*****",
  "port": 1812,
  "protocol": "PAP",
  "timeoutInSeconds": 10,
  "retries": 5
}

```


Read RADIUS Provider Information

The following is an example of how to read RADIUS provider information.

GET:

```
GET https://<host>:<port>/api/v1/auth/providers/radius/59cc163c1f00005400ab4ffd
```

Request:

```
<empty >
```

Response:

```
HTTP/1.1 200
{
  "id": "59cc163c1f00005400ab4ffd",
  "host": "10.23.237.168",
  "description": "ACS RADIUS Provider",
  "sharedSecret": "*****",
  "port": 1812,
  "protocol": "PAP",
  "timeoutInSeconds": 10,
  "retries": 5
}
```

List Providers

The following is an example of how to list RADIUS providers.

GET:

```
GET https://<host>:<port>/api/v1/providers/radius
```

Request:

```
<empty>
```

Response:

```
HTTP/1.1 200
{
  "radiusProviders": [
    {
      "id": "59cc163c1f00005400ab4ffd",
      "host": "10.23.237.168",
      "description": "RADIUS",
      "sharedSecret": "*****",
      "port": 1812,
      "protocol": "PAP",
      "timeoutInSeconds": 10,
      "retries": 5
    },
    {
      "id": "59cc19942100004a0077f85e",
      "host": "10.23.237.169",
      "description": "ACS RADIUS Provider",
      "sharedSecret": "*****",
      "port": 1812,

```

```

        "protocol": "PAP",
        "timeoutInSeconds": 10,
        "retries": 5
      }
    ]
  }

```

The following is an example of how to list all providers.

GET:

GET https://<host>:<port>/api/v1/providers

Request:

<empty>

Response:

```

HTTP/1.1 200
{
  "providers": [
    {
      "id": "59cc30942200005e0076c633",
      "host": "10.23.237.101",
      "providerType": "radius",
      "description": "RADIUS",
      "port": 1812,
      "protocol": "PAP",
      "timeoutInSeconds": 10,
      "retries": 5
    },
    {
      "id": "59cc3157220000780076c634",
      "host": "10.23.237.188",
      "providerType": "radius",
      "description": "RADIUS",
      "port": 1812,
      "protocol": "PAP",
      "timeoutInSeconds": 10,
      "retries": 5
    }
  ]
}

```

Update an Existing RADIUS Provider

The following is an example of how to update an existing RADIUS provider.

PUT:

PUT https://<host>:<port>/api/v1/auth/providers/radius/59cc163c1f00005400ab4ffd

Request:

```

{
  "id": "59cc163c1f00005400ab4ffd",
  "host": "10.23.237.168",
  "description": "ACS RADIUS Provider",
  "sharedSecret": "ins3965!",
}

```

```
    "port": 1812,  
    "protocol": "PAP",  
    "timeoutInSeconds": 10,  
    "retries": 3  
  }  
}
```

Response:

```
HTTP/1.1 200  
{  
  "id": "59cc163c1f00005400ab4ffd",  
  "host": "10.23.237.168",  
  "description": "ACS RADIUS Provider",  
  "sharedSecret": "*****",  
  "port": 1812,  
  "protocol": "PAP",  
  "timeoutInSeconds": 10,  
  "retries": 3  
}
```

Delete a RADIUS Provider

The following is an example of how to delete a RADIUS provider.

DELETE:

```
DELETE https://<host>:<port>/api/v1/auth/providers/radius/59cc163c1f00005400ab4ffd
```

Request:

```
<empty>
```

Response:

```
HTTP/1.1 204
```

APIs for TACACS+ Configurations

Create a New TACACS Provider

The following is an example of how to create a new TACACS provider.

POST:

```
POST https://<host>:<port>/api/v1/auth/providers/tacacs
```

Request:

```
{  
  "host": "10.23.237.168",  
  "description": "Tacacs Provider - 1",  
  "sharedSecret": "ins3965!",  
  "port": 49,  
  "protocol": "PAP",  
  "timeoutInSeconds": 10,  
  "retries":5  
}
```

```
}

```

Response:

```
HTTP/1.1 201
Location: /api/v1/auth/providers/tacacs/59cc163c1f00005400ab4ffd
{
  "id": "59cc163c1f00005400ab4ffd",
  "host": "10.23.237.168",
  "description": "Tacacs Provider - 1",
  "sharedSecret": "*****",
  "port": 49,
  "protocol": "PAP",
  "timeoutInSeconds": 10,
  "retries": 5
}
```

Read TACACS Provider Information

The following is an example of how to read TACACS provider information.

GET:

```
GET https://<host>:<port>/api/v1/auth/providers/tacacs/59cc163c1f00005400ab4ffd
```

Request:

```
<empty >
```

Response:

```
HTTP/1.1 200
{
  "id": "59cc163c1f00005400ab4ffd",
  "host": "10.23.237.168",
  "description": "Tacacs Provider - 1",
  "sharedSecret": "*****",
  "port": 49,
  "protocol": "PAP",
  "timeoutInSeconds": 10,
  "retries": 5
}
```

List TACACS Providers

The following is an example of how to list TACACS providers.

GET:

```
GET https://<host>:<port>/api/v1/providers/tacacs
```

Request:

```
<empty>
```

Response:

```

HTTP/1.1 200
{
  "tacacsProviders": [
    {
      "id": "59cc163c1f00005400ab4ffd",
      "host": "10.23.237.168",
      "description": "Tacacs Provider - 1",
      "sharedSecret": "*****",
      "port": 49,
      "protocol": "PAP",
      "timeoutInSeconds": 10,
      "retries": 5
    },
    {
      "id": "59cc19942100004a0077f85e",
      "host": "10.23.237.169",
      "description": "Tacacs Provider - 1",
      "sharedSecret": "*****",
      "port": 49,
      "protocol": "PAP",
      "timeoutInSeconds": 10,
      "retries": 5
    }
  ]
}

```

Update an Existing TACAS Provider

The following is an example of how to update an existing TACACS provider.

PUT:

PUT https://<host>:<port>/api/v1/auth/providers/tacacs/59cc163c1f00005400ab4ffd

Request:

```

{
  "id": "59cc163c1f00005400ab4ffd",
  "host": "10.23.237.168",
  "description": "Tacacs Provider - 1",
  "sharedSecret": "ins3965!",
  "port": 49,
  "protocol": "PAP",
  "timeoutInSeconds": 10,
  "retries": 3
}

```

Response:

```

HTTP/1.1 200
{
  "id": "59cc163c1f00005400ab4ffd",
  "host": "10.23.237.168",
  "description": "Tacacs Provider - 1",
  "sharedSecret": "*****",
  "port": 49,
  "protocol": "PAP",
  "timeoutInSeconds": 10,
  "retries": 3
}

```

Delete a TACACS Provider

The following is an example of how to delete a TACACS provider.

DELETE:

```
DELETE https://<host>:<port>/api/v1/auth/providers/tacacs/59cc163c1f00005400ab4ffd
```

Request:

```
<empty>
```

Response:

```
HTTP/1.1 204
```

APIs for Login Domain

Create a New Login Domain

The following is an example of how to create a new login domain.

POST:

```
POST https://<host>:<port>/api/v1/auth/domains
```

Request:

```
{
  "name" : "EngineeringGroup",
  "description" : "Used for RADIUS Auth Login",
  "realm" : "radius",
  "status": "active",
  "providerAssociations": [
    {
      "providerId": "59d5b4de8d0000ba00909f62",
      "priority" : 1
    }
  ]
}
```

Response:

```
HTTP/1.1 201
{
  "id": "59d6d0398d0000d2009eb2c4",
  "name": "EngineeringGroup",
  "description": "Used for RADIUS Auth Login",
  "realm": "radius",
  "status": "active",
  "providerAssociations": [
    {
      "providerId": "59d5b4de8d0000ba00909f62",
      "priority": 1
    }
  ]
}
```

List all Active Login Domains

The following is an example of how to list all active login domains.



-
- Note**
- This is an unauthorized APIC for initial login.
-

GET:

GET https://<host>:<port>/api/v1/auth/login-domains

Request:

```
{
  {
    "domains": [
      {
        "id": "59d6d0398d0000d2009eb2c4",
        "name": "EngineeringGroup",
        "description": "Used for RADIUS Auth Login",
        "realm": "radius",
        "status": "active",
        "providerAssociations": []
      },
      {
        "id": "59d6cfb58d0000bd009eb2c3",
        "name": "VM-TACACS",
        "description": "Used for Tacacs Auth Login",
        "realm": "tacacs",
        "status": "active",
        "providerAssociations": []
      }
    ]
  }
}
```



-
- Note**
- The token in the above example has been truncated for readability.
-

List Login Domains

The following is an example of how to list all login domains.

GET:

GET https://<host>:<port>/api/v1/auth/domains

Request:

```
{
  {
    "domains": [
      {
        "id": "59d6d0398d0000d2009eb2c4",
        "name": "EngineeringGroup",
        "description": "Used for RADIUS Auth Login",
        "realm": "radius",
        "status": "active",
        "providerAssociations": [
```

```

        {
            "providerId": "59d5b4de8d0000ba00909f62",
            "priority": 1
        },
        {
            "providerId": "59d6d0988d0000cb009eb2c5",
            "priority": 2
        }
    ]
},
{
    "id": "59d6cfb58d0000bd009eb2c3",
    "name": "VM-TACACS",
    "description": "Used for Tacacs Auth Login",
    "realm": "tacacs",
    "status": "active",
    "providerAssociations": [
        {
            "providerId": "59d5b4de8d0000ba00909f62",
            "priority": 1
        }
    ]
}
]
}

```

The following is an example of how to list a login domain detail.

POST:

POST https://<host>:<port>/api/v1/auth/domains/59d2c88f9100000105b849e0

Request:

```

{
    "id": "59d6d0398d0000d2009eb2c4",
    "name": "EngineeringGroup",
    "description": "Used for RADIUS Auth Login",
    "realm": "radius",
    "status": "active",
    "providerAssociations": [
        {
            "providerId": "59d5b4de8d0000ba00909f62",
            "priority": 1
        },
        {
            "providerId": "59d6d0988d0000cb009eb2c5",
            "priority": 2
        }
    ]
}

```

Modify an Existing Login Domain

The following is an example of how to modify an existing login domain.

PUT:

PUT https://<host>:<port>/api/v1/auth/domains/59d6d0398d0000d2009eb2c4

Request:

```

{

```



```

    "id": "59d6d0398d0000d2009eb2c4",
    "name": "EngineeringGroup",
    "description": "Used for RADIUS Auth Login",
    "realm": "radius",
    "status": "active",
    "providerAssociations": [
      {
        "providerId": "59d5b4de8d0000ba00909f62",
        "priority": 1
      },
      {
        "providerId": "59d6d0988d0000cb009eb2c5",
        "priority": 2
      }
    ]
  }
}

```

Modify an Existing Login Domain

The following is an example of how to modify an existing login domain.

PUT:

PUT https://<host>:<port>/api/v1/auth/domains/59d6d0398d0000d2009eb2c4

Request:

```

{
  "id": "59d6d0398d0000d2009eb2c4",
  "name": "EngineeringGroup",
  "description": "Used for RADIUS Auth Login",
  "realm": "radius",
  "status": "active",
  "providerAssociations": [
    {
      "providerId": "59d5b4de8d0000ba00909f62",
      "priority": 1
    },
    {
      "providerId": "59d6d0988d0000cb009eb2c5",
      "priority": 2
    }
  ]
}

```

APIs for Remote User Login

Remote User Login

The following is an example of how to read and create remote user login domains.

POST:

POST https://<host>:<port>/api/v1/auth/login

Request:

```

{
  "username": "bob",
  "password": "welcome!",
}

```

```
"domainId": "59d5b5978d0000d000909f65",
}
```

Response:

```
HTTP/1.1 201 (Created) -> Valid user credentials
```

```
{
  "userId": "0000ffff0000000000000020",
  "needsPasswordUpdate": false,
  "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9...e-g308aqI8",
  "permissions": [
    "view-sites",
    "manage-sites",
    "view-tenants",
    "manage-tenants",
    "view-tenant-schemas",
    "manage-tenant-schemas",
    "view-schemas",
    "manage-schemas",
    "view-users",
    "manage-users",
    "view-roles",
    "manage-roles"
  ]
}
```

```
HTTP/1.1 401 (Unauthorized) -> Invalid user credentials
```

```
{
  "code": 401,
  "message": "Invalid user credentials"
}
```

```
HTTP/1.1 500 (UnsupportedAuthType) -> Authentication domain not supported
```

```
{
  "code": 500,
  "message": "Authentication domain not supported"
}
```

**Note**

- This API does not require the Authorization header to be set.
- The token in the above example has been truncated for readability.

APIs for Labels

Create a New Label

The following is an example of how to create a new label.

POST:

```
POST https://<host>:<port>/api/v1/labels
```

Request:

```
{
  "name" : "devops",
  "type" : "site"
}
```

Response:

```
HTTP/1.1 201
Location: /api/v1/labels/58202f7066e6e10001c41236
{
  "id": "58202f7066e6e10001c41236",
  "name" : "devops",
  "type" : "site"
}
```

Read Label Information

The following is an example of how to read label information.

GET:

```
GET https://<host>:<port>/api/v1/labels/58202f7066e6e10001c41236
```

Request:

```
<empty>
```

Response:

```
HTTP/1.1 200
{
  "id": "58202f7066e6e10001c41236",
  "name" : "devops",
  "type" : "site"
}
```

Update a Label

The following is an example of how to update a label.

PUT:

```
PUT https://<host>:<port>/api/v1/labels/58202f7066e6e10001c41236
```

Request:

```
{
  "id": "58202f7066e6e10001c41236",
  "name" : "test",
  "type" : "site"
}
```

Response:

```
HTTP/1.1 200
{
  "id": "58202f7066e6e10001c41236",
  "name" : "test",
  "type" : "site"
}
```

Delete Labels

The following is an example of how to delete a label.

DELETE:

```
DELETE https://<host>:<port>/api/v1/labels/58202f7066e6e10001c41236
```

Request:

```
<empty>
```

Response:

```
HTTP/1.1 204
```

List Labels

The following is an example of how to list labels.

GET:

```
GET https://<host>:<port>/api/v1/labels
```



Note Filtering, pagination, and sorting is currently not supported.

Request:

```
<empty>
```

Response:

```
HTTP/1.1 200
{
  "labels": [
    {
      "id": "58202f7066e6e10001c41236",
      "name" : "devops",
      "type" : "site"
    },
    {
```

```
        "id": "58202fc466e6e10001c41237",
        "name" : "prod",
        "type" : "site"
    }
  ]
}
```

APIs for Roles

Read Role Information

The following is an example of how to read role information.

GET:

```
GET https://<host>:<port>/api/v1/role/591f58c1220000e406fa80b5
```

Request:

<empty>

Response:

```
HTTP/1.1 200
{
  "id": "591f58c1220000e406fa80b5",
  "name": "role1",
  "displayName": "Role 1",
  "description": "Description for Role 1",
  "permissions": [
    "view-sites",
    "manage-sites",
    "view-tenants",
    "manage-tenants",
    "view-tenant-schemas",
    "view-schemas",
    "manage-schemas",
    "view-users",
    "manage-users",
    "view-roles",
    "manage-roles"
  ]
}
```

List Roles

The following is an example of how to list roles.

GET:

```
GET https://<host>:<port>/api/v1/roles
```

Request:

<empty>

Response:

```
HTTP/1.1 200
{
  "roles": [
    {
      "id": "591f58c1220000e406fa80b5",
      "name": "role1",
      "displayName": "Role 1",
      "description": "Description for Role 1",
      "permissions": [
        "view-sites",
        "manage-sites",
        "view-tenants",
        "manage-tenants",
        "view-tenant-schemas",
        "view-schemas",
        "manage-schemas",
        "view-users",
        "manage-users",
        "view-roles",
        "manage-roles"
      ]
    },
    {
      "id": "591f5903220000f106fa80b6",
      "name": "role2",
      "displayName": "Role 2",
      "description": "Description for Role 2",
      "permissions": [
        "view-sites",
        "manage-sites",
        "view-tenants",
        "manage-tenants"
      ]
    }
  ]
}
```

APIs for Users

Create a New User

The following is an example of how to create a new user.

POST:

POST https://<host>:<port>/api/v1/users

Request:

```
{
```

```
"username": "user1",
"password": "Cisco@123!",
"firstName": "Foo",
"lastName": "Bar",
"emailAddress": "noreply@cisco.com",
"phoneNumber": "408-527-0000",
"accountStatus": "active",
"roles": [
  {
    "roleId": "591f5903220000f106fa80b6"
  }
]
```

Response:

```
HTTP/1.1 201
Location: /api/v1/users/58202f7066e6e10001c41236
{
  "id": "58202f7066e6e10001c41236",
  "username": "user1",
  "password": "*****",
  "firstName": "Foo",
  "lastName": "Bar",
  "emailAddress": "noreply@cisco.com",
  "phoneNumber": "408-527-0000",
  "accountStatus": "active",
  "roles": [
    {
      "roleId": "591f5903220000f106fa80b6"
    }
  ]
}
```

Reset Password for an Existing User

The following is an example of how to reset the password for an existing user.

PUT:

```
PUT https://<host>:<port>/api/v1/users/58202f7066e6e10001c41236/reset-password
```

Request:

```
{
  "currentPassword": "Cisco@123!",
  "newPassword": "Cisco@456!"
}
```

Response:

```
HTTP/1.1 200 OK
```



Note If a user password is system generated or generated by a user-manager in Multi-Site, then upon login, that user (including the admin) will be required to reset their own password.

The minimum requirements for the password field are as follows:

- Minimum 6 characters must be used
- At least one letter must be included
- At least one number must be included
- At least 1 special character must be used. But space and * are not supported.

Read User Information

The following is an example of how to read user information.

GET:

```
GET https://<host>:<port>/api/v1/users/58202f7066e6e10001c41236
```

Request:

```
<empty>
```

Response:

```
HTTP/1.1 200
{
  "id": "58202f7066e6e10001c41236",
  "username": "user1",
  "password": "*****",
  "firstName": "Foo",
  "lastName": "Bar",
  "emailAddress": "noreply@cisco.com",
  "phoneNumber": "408-527-0000",
  "accountStatus": "active",
  "roles": [
    {
      "roleId": "591f5903220000f106fa80b6"
    }
  ]
}
```

Update an Existing User

The following is an example of how to update an existing user.

PUT:


```
PUT https://<host>:<port>/api/v1/users/58202f7066e6e10001c41236
```

Request:

```
{
  "id": "58202f7066e6e10001c41236",
  "username": "user1",
  "password": "*****",
  "firstName": "Foo",
  "lastName": "Bar",
  "emailAddress": "noreply@cisco.com",
  "phoneNumber": "408-527-0000",
  "accountStatus": "active",
  "roles": [
    {
      "roleId": "591f58c1220000e406fa80b5"
    },
    {
      "roleId": "591f5903220000f106fa80b6"
    }
  ]
}
```

Response:

```
HTTP/1.1 200
{
  "id": "58202f7066e6e10001c41236",
  "username": "user1",
  "password": "*****",
  "firstName": "Foo",
  "lastName": "Bar",
  "emailAddress": "noreply@cisco.com",
  "phoneNumber": "408-527-0000",
  "accountStatus": "active",
  "roles": [
    {
      "roleId": "591f58c1220000e406fa80b5"
    },
    {
      "roleId": "591f5903220000f106fa80b6"
    }
  ]
}
```

Delete a User

The following is an example of how to delete a user.

DELETE:

```
DELETE https://<host>:<port>/api/v1/users/58202f7066e6e10001c41236
```

Request:

<empty>

Response:

HTTP/1.1 204

List Users

The following is an example of how to list users.

GET:

GET https://<host>:<port>/api/v1/users

Request:

<empty>

Response:

HTTP/1.1 200

```
{
  "users": [
    {
      "id": "58202f7066e6e10001c41236",
      "username": "user1",
      "password": "",
      "firstName": "Foo",
      "lastName": "Bar",
      "emailAddress": "noreply@cisco.com",
      "phoneNumber": "408-527-000",
      "accountStatus": "active",
      "roles": [
        {
          "roleId": "591f5903220000f106fa80b6"
        }
      ]
    },
    {
      "id": "58202fc466e6e10001c41237",
      "username": "user2",
      "password": "",
      "firstName": "Foo2",
      "lastName": "Bar2",
      "emailAddress": "noreply@cisco.com",
      "phoneNumber": "408-527-0000",
      "accountStatus": "Inactive",
      "roles": []
    }
  ]
}
```

List Roles That May Be Associated With Users

The following is an example of how to list roles that may be associated with users.

GET:

```
GET https://<host>:<port>/api/v1/users/allowed-roles
```

Request:

<empty>

Response:

```
HTTP/1.1 200
{
  "roles": [
    {
      "id": "591f58c1220000e406fa80b5",
      "displayName": "Role 1",
      "permissions": [
        "view-sites",
        "manage-sites",
        "view-tenants",
        "manage-tenants",
        "view-tenant-schemas",
        "view-schemas",
        "manage-schemas",
        "view-users",
        "manage-users",
        "view-roles",
        "manage-roles"
      ]
    },
    {
      "id": "591f5903220000f106fa80b6",
      "displayName": "Role 2",
      "permissions": [
        "view-sites",
        "manage-sites",
        "view-tenants",
        "manage-tenants"
      ]
    }
  ]
}
```

Retrieve Roles Associated With a User

The following is an example of how to retrieve roles associated with a user.

GET:

```
GET https://<host>:<port>/api/v1/users/58202f7066e6e10001c41236/roles
```

Request:

```
<empty>
```

Response:

```
HTTP/1.1 200
{
  "roles": [
    {
      "id": "591f58c1220000e406fa80b5",
      "displayName": "Role 1",
      "permissions": [
        "view-sites",
        "manage-sites",
        "view-tenants",
        "manage-tenants",
        "view-tenant-schemas",
        "view-schemas",
        "manage-schemas",
        "view-users",
        "manage-users",
        "view-roles",
        "manage-roles"
      ]
    }
  ]
}
```

Retrieve Permissions Associated With a User

The following is an example of how to retrieve permissions associated with a user.

GET:

```
GET https://<host>:<port>/api/v1/users/58202f7066e6e10001c41236/permissions
```

Request:

```
<empty>
```

Response:

```
HTTP/1.1 200
{
  "permissions": [
    "view-sites",
    "manage-sites",
    "view-tenants",
    "manage-tenants",
    "view-tenant-schemas",
    "view-schemas",
    "manage-schemas",
    "view-users",
    "manage-users",
    "view-roles",
    "manage-roles"
  ]
}
```

```
}

```

APIs for Fabric Connectivity

Get Fabric Connectivity for a Site

Example for ACI Multi-Site Release 1.0(2)

The following is an example of how to include fabric connectivity status for a site.

GET:

```
GET https://<host>:<port>/api/v1/sites/fabric-connectivity?include=status
```

Request:

```
<empty>
```

Response:

```
HTTP/1.1 200
```

```
{
  "sites" : [
    {
      "siteId": "59a06d4f33000033003ddc01",
      "siteName": "site1",
      "error": "",      <-- When 'error' is empty or doesn't exist, derive combined
connectivity-status for this site      from all individual status under msiteBgpSessionsStatus,
msiteUnicastTunnelsStatus &      msiteMulticastTunnelsStatus under this site

      "msiteBgpSessionsStatus": [
        {
          "podId": 1,
          "nodeId": 101,
          "nodeName": "spine11",
          "peerSiteId": "59a06d4f33000033003ddc02",
          "peerSiteName": "site2",
          "peerPodId": 1,
          "peerNodeId": 201,
          "peerNodeName": "spine21",
          "status": "up | down"
        },
        {
          "podId": 1,
          "nodeId": 101,
          "nodeName": "spine11",
          "peerSiteId": "59a06d4f33000033003ddc02",
          "peerSiteName": "site2",
          "peerPodId": 1,
          "peerNodeId": 202,
          "peerNodeName": "spine22",
          "status": "up | down"
        }
      ],
    }
  ]
}
```

```

{
  "podId": 1,
  "nodeId": 102,
  "nodeName": "spine12",
  "peerSiteId": "59a06d4f33000033003ddc02",
  "peerSiteName": "site2",
  "peerPodId": 1,
  "peerNodeId": 201,
  "peerNodeName": "spine21",
  "status": "up | down"
},
{
  "podId": 1,
  "nodeId": 102,
  "nodeName": "spine12",
  "peerSiteId": "59a06d4f33000033003ddc02",
  "peerSiteName": "site2",
  "peerPodId": 1,
  "peerNodeId": 202,
  "peerNodeName": "spine22",
  "status": "up | down"
},
{
  "podId": 1,
  "nodeId": 101,
  "nodeName": "spine11",
  "peerSiteId": "59a06d4f33000033003ddc03",
  "peerSiteName": "site3",
  "peerPodId": 1,
  "peerNodeId": 301,
  "peerNodeName": "spine31",
  "status": "up | down"
},
{
  "podId": 1,
  "nodeId": 101,
  "nodeName": "spine11",
  "peerSiteId": "59a06d4f33000033003ddc03",
  "peerSiteName": "site3",
  "peerPodId": 1,
  "peerNodeId": 302,
  "peerNodeName": "spine32",
  "status": "up | down"
},
{
  "podId": 1,
  "nodeId": 102,
  "nodeName": "spine12",
  "peerSiteId": "59a06d4f33000033003ddc03",
  "peerSiteName": "site3",
  "peerPodId": 1,
  "peerNodeId": 301,
  "peerNodeName": "spine31",
  "status": "up | down"
},
{
  "podId": 1,
  "nodeId": 102,
  "nodeName": "spine12",
  "peerSiteId": "59a06d4f33000033003ddc03",
  "peerSiteName": "site3",
  "peerPodId": 1,
  "peerNodeId": 302,
  "peerNodeName": "spine32",

```

```

        "status": "up | down"
    }
],
"msiteVxlanUnicastTunnelsStatus": [
    {
        "podId": 1,
        "nodeId": 101,
        "nodeName": "spine11",
        "peerSiteId": "59a06d4f33000033003ddc02",
        "peerSiteName": "site2",
        "peerPodId": 1,
        "peerNodeId": 201,
        "peerNodeName": "spine21",
        "status": "up | down"
    },
    {
        "podId": 1,
        "nodeId": 101,
        "nodeName": "spine11",
        "peerSiteId": "59a06d4f33000033003ddc02",
        "peerSiteName": "site2",
        "peerPodId": 1,
        "peerNodeId": 202,
        "peerNodeName": "spine22",
        "status": "up | down"
    },
    {
        "podId": 1,
        "nodeId": 102,
        "nodeName": "spine12",
        "peerSiteId": "59a06d4f33000033003ddc02",
        "peerSiteName": "site2",
        "peerPodId": 1,
        "peerNodeId": 201,
        "peerNodeName": "spine21",
        "status": "up | down"
    },
    {
        "podId": 1,
        "nodeId": 102,
        "nodeName": "spine12",
        "peerSiteId": "59a06d4f33000033003ddc02",
        "peerSiteName": "site2",
        "peerPodId": 1,
        "peerNodeId": 202,
        "peerNodeName": "spine22",
        "status": "up | down"
    },
    {
        "podId": 1,
        "nodeId": 101,
        "nodeName": "spine11",
        "peerSiteId": "59a06d4f33000033003ddc03",
        "peerSiteName": "site3",
        "peerPodId": 1,
        "peerNodeId": 301,
        "peerNodeName": "spine31",
        "status": "up | down"
    },
    {
        "podId": 1,
        "nodeId": 101,
        "nodeName": "spine11",
        "peerSiteId": "59a06d4f33000033003ddc03",

```

```

        "peerSiteName": "site3",
        "peerPodId": 1,
        "peerNodeId": 302,
        "peerNodeName": "spine32",
        "status": "up | down"
    },
    {
        "podId": 1,
        "nodeId": 102,
        "nodeName": "spine12",
        "peerSiteId": "59a06d4f33000033003ddc03",
        "peerSiteName": "site3",
        "peerPodId": 1,
        "peerNodeId": 301,
        "peerNodeName": "spine31",
        "status": "up | down"
    },
    {
        "podId": 1,
        "nodeId": 102,
        "nodeName": "spine12",
        "peerSiteId": "59a06d4f33000033003ddc03",
        "peerSiteName": "site3",
        "peerPodId": 1,
        "peerNodeId": 302,
        "peerNodeName": "spine32",
        "status": "up | down"
    }
],
"msiteVxlanMulticastTunnelsStatus": [
    {
        "podId": 1,
        "nodeId": 101,
        "nodeName": "spine11",
        "peerSiteId": "59a06d4f33000033003ddc02",
        "peerSiteName": "site2",
        "peerPodId": 1,
        "peerNodeId": 201,
        "peerNodeName": "spine21",
        "status": "up | down"
    },
    {
        "podId": 1,
        "nodeId": 101,
        "nodeName": "spine11",
        "peerSiteId": "59a06d4f33000033003ddc02",
        "peerSiteName": "site2",
        "peerPodId": 1,
        "peerNodeId": 202,
        "peerNodeName": "spine22",
        "status": "up | down"
    },
    {
        "podId": 1,
        "nodeId": 102,
        "nodeName": "spine12",
        "peerSiteId": "59a06d4f33000033003ddc02",
        "peerSiteName": "site2",
        "peerPodId": 1,
        "peerNodeId": 201,
        "peerNodeName": "spine21",
        "status": "up | down"
    }
]

```



```

        "podId": 1,
        "nodeId": 102,
        "nodeName": "spine12",
        "peerSiteId": "59a06d4f33000033003ddc02",
        "peerSiteName": "site2",
        "peerPodId": 1,
        "peerNodeId": 202,
        "peerNodeName": "spine22",
        "status": "up | down"
    },
    {
        "podId": 1,
        "nodeId": 101,
        "nodeName": "spine11",
        "peerSiteId": "59a06d4f33000033003ddc03",
        "peerSiteName": "site3",
        "peerPodId": 1,
        "peerNodeId": 301,
        "peerNodeName": "spine31",
        "status": "up | down"
    },
    {
        "podId": 1,
        "nodeId": 101,
        "nodeName": "spine11",
        "peerSiteId": "59a06d4f33000033003ddc03",
        "peerSiteName": "site3",
        "peerPodId": 1,
        "peerNodeId": 302,
        "peerNodeName": "spine32",
        "status": "up | down"
    },
    {
        "podId": 1,
        "nodeId": 102,
        "nodeName": "spine12",
        "peerSiteId": "59a06d4f33000033003ddc03",
        "peerSiteName": "site3",
        "peerPodId": 1,
        "peerNodeId": 301,
        "peerNodeName": "spine31",
        "status": "up | down"
    },
    {
        "podId": 1,
        "nodeId": 102,
        "nodeName": "spine12",
        "peerSiteId": "59a06d4f33000033003ddc03",
        "peerSiteName": "site3",
        "peerPodId": 1,
        "peerNodeId": 302,
        "peerNodeName": "spine32",
        "status": "up | down"
    }
]
},
{
    "siteId": "59a06d4f33000033003ddc02",
    "siteName": "site2",
    "error" : "site not reachable", <-- When this is non-empty, individual bgp-session
or tunnel statuses would be empty list                                and make combined status to be down and show this
error in summary/details page
    "msiteBgpSessionsStatus": [ ],

```

```

    "msiteVxlanUnicastTunnelsStatus": [],
    "msiteVxlanMulticastTunnelsStatus": []
  },
  {
    "siteId": "59a06d4f33000033003ddc03",
    "siteName": "site3",
    <-- 'error' won't exist or empty when MSC able
to fetch bgp or tunnel statuses from APICs/sites
    "msiteBgpSessionsStatus": [.....],
    "msiteVxlanUnicastTunnelsStatus": [.....],
    "msiteVxlanMulticastTunnelsStatus": [.....]
  }
]

```

Example for ACI Multi-Site Release 1.0(1)

The following is an example of how to get fabric connectivity for a site.

GET:

GET https://<host>:<port>/api/v1/sites/fabric-connectivity

Request:

<empty>

Response:

```

HTTP/1.1 200HTTP/1.1 200 {
  "controlPlaneBgpConfig" : {
    "peeringType" : "full-mesh | route-reflector",
    "ttl" : 16,
    "keepAliveInterval" : 60,
    "holdInterval" : 180,
    "staleInterval" : 300,
    "gracefulRestartEnabled" : true,
    "maxAsLimit" : 0
  },
  "ospfPolicies" : [ {
    "name" : "msc-ospf-policy-default",
    "networkType" : "broadcast | point-to-point | unspecified",
    "priority" : 1,
    "interfaceCost" : 1,
    "interfaceControls" : [ "advertise-subnet", "mtu-ignore", "bfd", "passive-participation"
], // Array of strings.. values are optional.. those are all possible values
    "helloInterval" : 10,
    "deadInterval" : 40,
    "retransmitInterval" : 5,
    "transmitDelay" : 1
  } ],
  "sites" : [ {
    "id" : "5935b0b633000088017723a4",
    "apicSiteId" : 1,
    "fabricId" : 1,
    "msiteEnabled" : true,
    "msiteDataPlaneMulticastTep" : "11.30.1.1",
    "bgpAsn" : 1234,
    "bgpCommunity" : "extended:as2-nn4:4:15",
    "bgpPassword" : "",
    "ospfAreaId" : "57", // Number or in IP address format

```

```

"ospfAreaType" : "nssa | regular | stub",
"ospfAreaCost" : 1,
"externalRoutedDomain" : "uni/l3dom-MultisiteL3Domain1-DoNotDelete",
"externalSubnetsToImport" : [ "20.23.10.0/16", "20.23.12.0/16"],
"pods" : [ {
  "podId" : 1,
  "name" : "pod-1",
  "msiteDataPlaneUnicastTep" : "11.20.1.1",
  "spines" : [ {
    "nodeId" : 201,
    "name" : "spine1",
    "ports" : [ {
      "portId" : "1/29",
      "ipAddress" : "20.23.10.1/24",
      "mtu": "inherit",          // value is 'inherit' or number from range 576 to 9000

      "routingPolicy" : "msc-ospf-policy-default"
    } ],
    "bgpPeeringEnabled" : true,
    "msiteControlPlaneTep" : "11.11.11.11",
    "routeReflectorEnabled" : true,
    "health" : {
      "score" : 93
    },
    "faults" : [ {
      "type" : "minor",
      "message" : "This fault occurs when a ntp configuration on a controller has
problems",
      "code" : "F1700"
    }, {
      "type" : "minor",
      "message" : "This fault occurs when an End Point Group is incompletely or
incorrectly configured.",
      "code" : "F0467"
    } ]
  } ],
  "health" : {
    "score" : 87
  },
  "faults" : [ {
    "type" : "major",
    "message" : "This fault occurs when vpc interface goes down while peer interface
is also down.",
    "code" : "F1296"
  }, { "type" : "minor",
    "message" : "This fault occurs when psu is shut down",
    "code" : "F1451"
  } ]
} ],
"status" : {
  "state" : "success"
},
"health" : {
  "score" : 87
},
"faults" : [ {
  "type" : "major",
  "message" : "This fault occurs when a physical interface on a controller is in the
link-down state.",
  "code" : "F0103"
}, {
  "type" : "major",
  "message" : "Task or FSM failure on object of class dhcpPool",
  "code" : "F608160"
} ]

```

```

    } ]
  } ]
}

```

The following is an example of how to include fabric connectivity status for a site.

GET:

```
GET https://<host>:<port>/api/v1/sites/fabric-connectivity?include=status
```

Request:

```
<empty>
```

Response:

```

HTTP/1.1 200
{
  ...

  "sites" : [ {
    "id" : "58dd7707120000370089450f",
    ...
    "status" : {
      "state" : "success"
    }
  }, {
    "id" : "58dd77071200003600894510",
    ...
    "status" : {
      "state" : "in-progress"
    }
  }, {
    "id" : "58dd77071200003600894520",
    ...
    "status" : {
      "state" : "failure",
      "message" : "Failed to sync with site 58dd77071200003600894520."
    }
  }
} ]
}

```

The following is an example of how to obtain fabric connectivity health faults.

GET:

```
GET https://<host>:<port>/api/v1/sites/fabric-connectivity?include=health,faults
```

Request:

```
<empty>
```

Response:

```

HTTP/1.1 200HTTP/1.1 200
{
  ...

  "sites" : [ {
    "id" : "58dd7707120000370089450f",

```

```

...
"pods" : [ {
  "podId" : 1,
  "name" : "pod-1",
  ...
  "spines" : [ {
    "nodeId" : 201,
    "name" : "spine1",
    ...
    "health" : { "score": 100 }
    "faults" : [ {
      "code": "F0103",
      "message": "...",
      "type": "minor"
    } ]
  } ]
} ]
} ]
}

```

Update Fabric Connectivity

The following is an example of how to get an update for fabric connectivity.

PUT:

PUT https://<host>:<port>/api/v1/sites/fabric-connectivity

Request:

```

{
  "controlPlaneBgpConfig" : {
    "peeringType" : "full-mesh | route-reflector",
    "ttl" : 16,
    "keepAliveInterval" : 60,
    "holdInterval" : 180,
    "staleInterval" : 300,
    "gracefulRestartEnabled" : true,
    "maxAsLimit" : 0
  },
  "ospfPolicies" : [ {
    "name" : "msc-ospf-policy-default",
    "networkType" : "broadcast | point-to-point | unspecified",
    "priority" : 1,
    "interfaceCost" : 1,
    "interfaceControls" : [ "advertise-subnet", "mtu-ignore", "bfd", "passive-participation"
  ], //values are optional
    "helloInterval" : 10,
    "deadInterval" : 40,
    "retransmitInterval" : 5,
    "transmitDelay" : 1
  } ],
  "sites" : [ {
    "id" : "5935b0b633000088017723a4",
    "apicSiteId" : 1,
    "fabricId" : 1,
    "msiteEnabled" : true,
    "msiteDataPlaneMulticastTep" : "11.30.1.1",
    "bgpAsn" : 1234,
    "bgpCommunity" : "extended:as2-nn4:4:15",
    "bgpPassword" : "",
    "ospfAreaId" : "57", // Number or in IP address format
  } ]
}

```

```

"ospfAreaType" : "nssa | regular | stub",
"ospfAreaCost" : 1,
"externalRoutedDomain" : "uni/l3dom-MultisiteL3Domain1-DoNotDelete",
"externalSubnetsToImport" : [ "20.23.10.0/16", "20.23.12.0/16"],
"pods" : [ {
  "podId" : 1,
  "name" : "pod-1",
  "msiteDataPlaneUnicastTep" : "11.20.1.1",
  "spines" : [ {
    "nodeId" : 201,
    "name" : "spine1",
    "ports" : [ {
      "portId" : "1/29",
      "ipAddress" : "20.23.10.1/24",
      "mtu" : "inherit",          // value is 'inherit' or number from range 576 to 9000

      "routingPolicy" : "msc-ospf-policy-default"
    } ],
    "bgpPeeringEnabled" : true,
    "msiteControlPlaneTep" : "11.11.11.11",
    "routeReflectorEnabled" : true,
  } ],
} ],
} ]
}

```

Response:

```

{
  "controlPlaneBgpConfig" : {
    "peeringType" : "full-mesh | route-reflector",
    "ttl" : 16,
    "keepAliveInterval" : 60,
    "holdInterval" : 180,
    "staleInterval" : 300,
    "gracefulRestartEnabled" : true,
    "maxAsLimit" : 0
  },
  "ospfPolicies" : [ {
    "name" : "msc-ospf-policy-default",
    "networkType" : "broadcast | point-to-point | unspecified",
    "priority" : 1,
    "interfaceCost" : 1,
    "interfaceControls" : [ "advertise-subnet", "mtu-ignore", "bfd", "passive-participation"
  ], // Array of strings.. values are optional.. those are all possible values
    "helloInterval" : 10,
    "deadInterval" : 40,
    "retransmitInterval" : 5,
    "transmitDelay" : 1
  } ],
  "sites" : [ {
    "id" : "5935b0b633000088017723a4",
    "apicSiteId" : 1,
    "fabricId" : 1,
    "msiteEnabled" : true,
    "msiteDataPlaneMulticastTep" : "11.30.1.1",
    "bgpAsn" : 1234,
    "bgpCommunity" : "extended:as2-nn4:4:15",
    "bgpPassword" : "",
    "ospfAreaId" : "57",          // Number or in IP address format
    "ospfAreaType" : "nssa | regular | stub",
    "ospfAreaCost" : 1,
    "externalRoutedDomain" : "uni/l3dom-MultisiteL3Domain1-DoNotDelete",
  }
]
}

```

```

"externalSubnetsToImport" : [ "20.23.10.0/16", "20.23.12.0/16"],
"pods" : [ {
  "podId" : 1,
  "name" : "pod-1",
  "msiteDataPlaneUnicastTep" : "11.20.1.1",
  "spines" : [ {
    "nodeId" : 201,
    "name" : "spine1",
    "ports" : [ {
      "portId" : "1/29",
      "ipAddress" : "20.23.10.1/24",
      "mtu" : "inherit",          // value is 'inherit' or number from range 576 to 9000

      "routingPolicy" : "msc-ospf-policy-default"
    } ],
    "bgpPeeringEnabled" : true,
    "msiteControlPlaneTep" : "11.11.11.11",
    "routeReflectorEnabled" : true,
  } ],
} ],
} ]
}

```

Refresh Fabric Connectivity for a Site

The following is an example of how to refresh fabric connectivity for a site.

GET:

```
GET https://<host>:<port>/api/v1/sites/:id/fabric-connectivity/refresh
```

Example: GET /api/v1/sites/58dd77071200003600894510/fabric-connectivity/refresh

Request:

<empty>

Response:

HTTP/1.1 200

```

{
  "id" : "5935b0b633000088017723a4",
  "apicSiteId" : 1,
  "fabricId" : 1,
  "msiteEnabled" : true,
  "msiteDataPlaneMulticastTep" : "11.30.1.1",
  "bgpAsn" : 1234,
  "bgpCommunity" : "extended:as2-nn4:4:15",
  "bgpPassword" : "",
  "ospfAreaId" : "57",          // Number or in IP address format
  "ospfAreaType" : "nssa | regular | stub",
  "ospfAreaCost" : 1,
  "externalRoutedDomain" : "uni/l3dom-MultisiteL3Domain1-DoNotDelete",
  "externalSubnetsToImport" : [ "20.23.10.0/16", "20.23.12.0/16"],
  "pods" : [ {
    "podId" : 1,
    "name" : "pod-1",
    "msiteDataPlaneUnicastTep" : "11.20.1.1",
    "spines" : [ {
      "nodeId" : 201,
      "name" : "spine1",

```

```

    "ports" : [ {
      "portId" : "1/29",
      "ipAddress" : "20.23.10.1/24",
      "mtu" : "inherit",          // value is 'inherit' or number from range 576 to 9000

      "routingPolicy" : "msc-ospf-policy-default"
    } ],
    "bgpPeeringEnabled" : true,
    "msiteControlPlaneTep" : "11.11.11.11",
    "routeReflectorEnabled" : true,
  } ]
}

```

Fabric Leaf Switches and Physical Interfaces

Get Nodes

The following is an example of how to get nodes.

GET:

```
GET https://<host>:<port>/api/v1/aci/sites/:id/nodes?type=<leaf,spine>
```

Request:

<empty>

Response:

```

HTTP/1.1 200
{
  "nodes" : [ {
    "dn" : "topology/pod-1/node-101",
    "name" : "leaf1",
    "type" : "leaf"
  }, {
    "dn" : "topology/pod-1/node-102",
    "name" : "leaf2",
    "type" : "leaf"
  } ]
}

```

Get Physical Interfaces Details

The following is an example of how to get physical interfaces details.

GET:

```
GET https://<host>:<port>/api/v1/aci/sites/:id/nodes/interfaces?node=<node_dn>&type=physical
```

Request:

<empty>

Response:

```
HTTP/1.1 200HTTP/1.1 200
{
  "interfaces": [
    {
      "dn": "topology/pod-1/paths-101/pathep-[eth1/1]",
      "name": "eth1/1",
      "type": "physical"
    },
    {
      "dn": "topology/pod-1/paths-101/pathep-[eth1/10]",
      "name": "eth1/10",
      "type": "physical"
    }
  ]
}
```

The following is an example of how to get physical interfaces, direct port channel type.

GET:

```
GET https://<host>:<port>/api/v1/aci/sites/:id/nodes/interfaces?type=dpc
```

Request:

```
<empty>
```

Response:

```
HTTP/1.1 200HTTP/1.1 200
{
  "interfaces": [
    {
      "dn": "topology/pod-1/paths-101/pathep-[Switch101_1-ports-3-4_PolGrp]",
      "name": "Switch101_1-ports-3-4_PolGrp",
      "type": "dpc"
    }
  ]
}
```

The following is an example of how to get physical interfaces, virtual port channel type.

GET:

```
GET https://<host>:<port>/api/v1/aci/sites/:id/nodes/interfaces?type=vpc
```

Request:

```
<empty>
```

Response:

```
HTTP/1.1 200HTTP/1.1 200
{
  "interfaces": [
    {
      "dn": "topology/pod-1/protpaths-101-102/pathep-[Switch101-102_1-ports-5_PolGrp]",
      "name": "Switch101-102_1-ports-5_PolGrp",
      "type": "vpc"
    }
  ]
}
```

```

    }
  ]
}

```

APIs for Tenants

Create a Tenant

The following is an example of how to create a Tenant.

POST:

POST https://<host>:<port>/api/v1/tenants

Request:

```

{
  "displayName": "Tenant 2",
  "name": "Tenant2",
  "description": "Description for Tenant 2",
  "siteAssociations": [
    {
      "siteId": "58dfeae2fd0000fd00cee617",
      "securityDomains": ["TestSecurityDomain", "domain-mgmt"]
    }
  ],
  "userAssociations": [
    { "userId": "58202f7066e6e10001c41236" }
    { "userId": "58202fc466e6e10001c41237" }
  ]
}

```

Response:

```

HTTP/1.1 201
Location: /api/v1/tenants/58202f7066e6e10001c41236
{
  "id": "58ed806aa10000bb0044dale",
  "name": "Tenant2",
  "displayName": "Tenant 2",
  "description": "Description for Tenant 2",
  "siteAssociations": [
    {
      "siteId": "58dfeae2fd0000fd00cee617",
      "securityDomains": [
        "TestSecurityDomain",
        "mgmt"
      ]
    }
  ],
  "userAssociations": [
    { "userId": "58202f7066e6e10001c41236" }
    { "userId": "58202fc466e6e10001c41237" }
  ]
}

```

Read Tenant Information

The following is an example of how to read Tenant information.

GET:

```
GET https://<host>:<port>/api/v1/tenants/58202f7066e6e10001c41236
```

Request:

<empty>

Response:

```
HTTP/1.1 200
{
  "id": "58ed806aa10000bb0044dale",
  "name": "Tenant2",
  "displayName": "Tenant 2",
  "description": "Description for Tenant 2",
  "siteAssociations": [
    {
      "siteId": "58dfeae2fd0000fd00cee617",
      "securityDomains": [
        "TestSecurityDomain",
        "mgmt"
      ]
    }
  ],
  "userAssociations": [
    { "userId": "58202f7066e6e10001c41236" }
    { "userId": "58202fc466e6e10001c41237" }
  ]
}
```

Update an Existing Tenant

The following is an example of how to update an existing Tenant.

PUT:

```
PUT https://<host>:<port>/api/v1/tenants/58202f7066e6e10001c41236
```

Request:

```
{
  {
    "id": "58ed806aa10000bb0044dale",
    "name": "Tenant2",
    "displayName": "Tenant 2 (Updated)",
    "description": "Description for Tenant 2",
    "siteAssociations": [
      {
        "siteId": "58dfeae2fd0000fd00cee617",
        "securityDomains": [
          "TestSecurityDomain",
          "mgmt"
        ]
      }
    ]
  }
},
"userAssociations": [
```

```

    { "userId": "58202f7066e6e10001c41236" }
    { "userId": "58202fc466e6e10001c41237" }
    { "userId": "58202fc466e6e10001c41238" }
  ]
}

```



Note "id" is optional, other attributes are mandatory even if unchanged.

Response:

```

HTTP/1.1 200
{
  "id": "58ed806aa10000bb0044dale",
  "displayName": "Tenant 2 (Updated)",
  "name": "Tenant2",
  "description": "Description for Tenant 2",
  "siteAssociations": [
    {
      "siteId": "58dfeae2fd0000fd00cee617",
      "securityDomains": ["TestSecurityDomain", "mgmt"]
    }
  ],
  "userAssociations": [
    { "userId": "58202f7066e6e10001c41236" }
    { "userId": "58202fc466e6e10001c41237" }
    { "userId": "58202fc466e6e10001c41238" }
  ]
}

```

Delete a Tenant

The following is an example of how to delete a Tenant.

PUT:

```
DELETE https://<host>:<port>/api/v1/tenants/58202f7066e6e10001c41236
```

Request:

```
<empty>
```

Response:

```
HTTP/1.1 204
```

List Tenants

The following is an example of how to list Tenants.

PUT:

```
PUT https://<host>:<port>/api/v1/tenants/58202f7066e6e10001c41236
```

Request:

<empty>



Note Filtering, pagination, and sorting is currently not supported.

Response:

```
HTTP/1.1 200
{
  "tenants": [
    {
      "id": "58ed7f22a10000a10044da19",
      "name": "test",
      "displayName": "test",
      "description": "Description for test",
      "siteAssociations": [
        {
          "siteId": "58dfeae2fd0000fd00cee617",
          "securityDomains": []
        }
      ],
      "userAssociations": [
        { "userId": "58202f7066e6e10001c41236" }
        { "userId": "58202fc466e6e10001c41237" }
        { "userId": "58202fc466e6e10001c41238" }
      ]
    },
    {
      "id": "58ed7f46a10000a50044dala",
      "name": "Tenant1",
      "displayName": "Tenant 1",
      "description": "Description for Tenant 1",
      "siteAssociations": [
        {
          "siteId": "58dfeae2fd0000fd00cee617",
          "securityDomains": []
        }
      ],
      "userAssociations": [
        { "userId": "58202f7066e6e10001c41236" }
      ]
    },
    {
      "id": "58ed806aa10000bb0044dale",
      "name": "Tenant2",
      "displayName": "Tenant 2 (Updated)",
      "description": "Description for Tenant 2",
      "siteAssociations": [
        {
          "siteId": "58dfeae2fd0000fd00cee617",
          "securityDomains": [
            "TestSecurityDomain",
            "mgmt"
          ]
        }
      ]
    }
  ],
  "userAssociations": [

```

```

        { "userId": "58202f7066e6e10001c41236" }
      ]
    }
  ]
}

```

List Sites Associated with a Tenant

The following is an example of how to list sites associated with a Tenant.

GET:

```
GET /api/v1/tenants/allowed-sites
```

Request:

<empty>

Response:

```

HTTP/1.1 200
{
  "sites": [
    {
      "id": "591b6e5172040072044258ea",
      "name": "site1",
      "urls": [
        "https://10.23.236.232"
      ],
      "username": "admin",
      "password": "*****",
      "siteId": "1"
    },
    {
      "id": "591b71353f00002700a2e8b1",
      "name": "site2",
      "urls": [
        "https://10.23.236.233"
      ],
      "username": "admin",
      "password": "*****",
      "siteId": "2"
    }
  ]
}

```

List Users Associated with a Tenant

The following is an example of how to list users associated with a Tenant.

GET:

```
GET /api/v1/tenants/allowed-users
```

Request:

<empty>

Response:

HTTP/1.1 200

```
{
  "users": [
    {
      "id": "0000ffff0000000000000020",
      "username": "admin",
      "password": "*****",
      "firstName": "Admin",
      "lastName": "Admin",
      "emailAddress": "admin@msc.com",
      "phoneNumber": "(888) 123-4567",
      "accountStatus": "active",
      "roles": [
        {
          "roleId": "0000ffff0000000000000031"
        }
      ]
    },
    {
      "id": "593b12e62000004b00f13b21",
      "username": "bob",
      "password": "*****",
      "firstName": "Bob",
      "lastName": "Smith",
      "emailAddress": "bob@gmail.com",
      "phoneNumber": "(888) 123-7890",
      "accountStatus": "active",
      "roles": [
        {
          "roleId": "0000ffff0000000000000034"
        }
      ]
    },
    {
      "id": "593b12f62000005200f13b22",
      "username": "russ",
      "password": "*****",
      "firstName": "Russ",
      "lastName": "King",
      "emailAddress": "russ@gmail.com",
      "phoneNumber": "(800) 123-4545",
      "accountStatus": "active",
      "roles": [
        {
          "roleId": "0000ffff0000000000000033"
        },
        {
          "roleId": "0000ffff0000000000000035"
        },
        {
          "roleId": "0000ffff0000000000000032"
        }
      ]
    }
  ]
}
```

List Schemas Associated with a Tenant

The following is an example of how to list Schemas associated with a Tenant.

GET:

```
GET /api/v1/tenants/:id/schemas
```

Request:

<empty>

Response:

```
{
  "schemas": [
    {
      "id": "594170153200003000a8b4fc",
      "displayName": "Schema 1",
      "templates": [],
      "sites": []
    },
    {
      "id": "594170153200003200a8b4fd",
      "displayName": "Schema 2",
      "templates": [],
      "sites": []
    },
    {
      "id": "594170153200003300a8b4ff",
      "displayName": "Schema 4",
      "templates": [],
      "sites": []
    }
  ]
}
```

List Schemas Associated with all Tenants

The following is an example of how to list Schemas associated with all Tenants.

GET:

```
GET /api/v1/tenants/schemas
```

Request:

<empty>

Response:

```
{
  "tenants": {
    "59416f953200004c008cc048": [
      {
        "id": "59416f9532000030008cc04a",
        "displayName": "Schema 1",

```



```

        "templates": [],
        "sites": []
    },
    {
        "id": "59416f9532000033008cc04b",
        "displayName": "Schema 2",
        "templates": [],
        "sites": []
    },
    {
        "id": "59416f9532000033008cc04d",
        "displayName": "Schema 4",
        "templates": [],
        "sites": []
    }
],
"59416f9532000036008cc049": [
    {
        "id": "59416f9532000052008cc04c",
        "displayName": "Schema 3",
        "templates": [],
        "sites": []
    },
    {
        "id": "59416f9532000033008cc04d",
        "displayName": "Schema 4",
        "templates": [],
        "sites": []
    }
]
}
}
}

```

Create a New Tenant Network Mapping

The following is an example of how to create a new Tenant network mapping.

PUT:

```
PUT /api/v1/tenants/:id/infra
```

Request:

```

{
  "externalEpgMappings": [
    {
      "siteMappings": [
        {
          "siteId": "59726e7a490000ea095b4cc8",
          "mapping": "uni/tn-Pepsi/out-L3Out-1/instP-Network"
        }
      ]
    }
  ]
}

```

Response:

```
HTTP/1.1 201
{

```

```

    "externalEpgMappings": [
      {
        "siteMappings": [
          {
            "siteId": "59726e7a490000ea095b4cc8",
            "mapping": "uni/tn-Pepsi/out-L3Out-1/instP-Network"
          }
        ]
      }
    ]
  }
}

```

Get Existing Tenant Network Mappings

The following is an example of how to list existing Tenant network mappings.

GET:

```
GET /api/v1/tenants/:id/infra
```

Request:

<empty>

Response:

```

HTTP/1.1 201
{
  "externalEpgMappings": [
    {
      "siteMappings": [
        {
          "siteId": "59726e7a490000ea095b4cc8",
          "mapping": "uni/tn-Pepsi/out-L3Out-1/instP-Network"
        }
      ]
    }
  ]
}

```

APIs for Schemas

Create a New Schema

Example for ACI Multi-Site Release 1.1(1)

The following is an example of how to create a new Schema .

POST:

```
POST https://<host>:<port>/api/v1/schemas
```

Request:

```

{
  "displayName": "Schema 1",
  "templates": [
    {
      "name": "template1",
      "displayName": "Template 1",
      "tenantId": "592c7c482501002501061578",
      "anps": [
        {
          "name": "anp1",
          "displayName": "Anp 1",
          "epgs": [
            {
              "name": "epg1",
              "displayName": "EPG 1",
              "contractRelationships": [
                {
                  "contractRef": "/templates/template1/contracts/contract1",
                  "relationshipType": "provider|consumer"
                }
              ]
            },
            {
              "name": "anp1",
              "displayName": "Anp 1",
              "subnets": [
                {
                  "ip": "192.168.132.1/24",
                  "description": "Subnet for IT department",
                  "scope": "private",
                  "shared": true
                },
                {
                  "ip": "192.168.133.1/24",
                  "description": "Subnet for sales department",
                  "scope": "public",
                  "shared": false
                }
              ]
            }
          ]
        },
        {
          "name": "anp1",
          "displayName": "Anp 1",
          "uSegEpg": true,
          "uSegAttrs": [
            {
              "name": "IP1",
              "displayName": "...",
              "description": "...",
              "type": "ip",
              "fvSubnet": true,
              "value": "0.0.0.0"
            },
            {
              "name": "IP2",
              "displayName": "...",
              "description": "...",
              "type": "ip",
              "fvSubnet": false,
              "value": "192.168.132.1"
            },
            {
              "name": "MAC",
              "displayName": "...",
              "description": "...",
              "type": "mac",
              "value": "aa:bb:cc:dd:ee:ff"
            },
            {
              "name": "VM Name",
              "displayName": "...",
              "description": "...",
              "type": "vm-name",
              "operator": "equals | contains | startsWith | endsWith",
              "value": "..."
            },
            {
              "name": "VM Data Center",
              "displayName": "...",
            }
          ]
        }
      ]
    }
  ]
}

```

```

        "description" : "...",
        "type" : "rootContName",
        "operator" : "...",
        "value" : "..."
    }, {
        "name" : "VM Hypervisor Identifier",
        "displayName" : "...",
        "description" : "...",
        "type" : "hv",
        "operator" : "...",
        "value" : "..."
    }, {
        "name" : "VmOperatingSystem",
        "displayName" : "...",
        "description" : "...",
        "type" : "guest-os",
        "operator" : "...",
        "value" : "..."
    }, {
        "name" : "VM Tag",
        "displayName" : "...",
        "description" : "...",
        "type" : "tag",
        "operator" : "...",
        "category" : "...",
        "value" : "..."
    }, {
        "name" : "Vm Identifier",
        "displayName" : "...",
        "description" : "...",
        "type" : "vm",
        "operator" : "...",
        "value" : "..."
    }, {
        "name" : "Vm Vmm Domain",
        "displayName" : "...",
        "description" : "...",
        "type" : "domain",
        "operator" : "...",
        "value" : "..."
    }, {
        "name" : "Vm Vnic Dn",
        "displayName" : "...",
        "description" : "...",
        "type" : "vnic",
        "operator" : "...",
        "value" : "..."
    } ],
    "bdRef" : "/templates/templatel/bds/bd1"
},
{
    "name" : "epg2",
    "displayName" : "EPG 2",
    "contractRelationships" : [
        {
            "contractRef" : "/templates/templatel/contracts/contract2",
            "relationshipType" : "provider|consumer"
        }
    ],
    "bdRef" : "templates/templatel/bds/bd1"
}
]
},
{

```

```

    "displayName": "AP 10",
    "name": "AP10",
    "anpRef": "/templates/Templat1/anps/AP10",
    "epgs": [{
      "displayName": "EPG 10",
      "name": "EPG10",
      "epgRef": "/templates/Templat1/anps/AP10/epgs/EPG10",
      "bdRef": "/templates/Templat1/bds/BD10",
      "uSegEpg": false,
      "contractRelationships": [{
        "contractRef": "/templates/Templat1/contracts/Contract34",
        "relationshipType": "provider"
      }],
      "subnets": [],
      "uSegAttrs": [],
      "intraEpg": "unenforced",
      "proxyArp": false
    }
  ],
  "vrfs": [
    {
      "name": "vrf1",
      "displayName": "VRF 1"
    }
  ],
  "bds": [
    {
      "name": "bd1",
      "displayName": "BD 1",
      "l3UnknownMulticastFlooding": "flood|opt-flood",
      "intersiteBumTrafficAllow": true,
      "multiDestinationFlooding": "bd-flood|drop|encap-flood",
      "l2UnknownUnicast": "flood|proxy",
      "l2Stretch": true,
      "subnets": [ ... ],
      "vrfRef": "templates/templat1/vrfs/vrf1"
    }
  ],
  "contracts": [
    {
      "name": "contract1",
      "displayName": "Contract 1",
      "filters": "80",
      "filterRelationships": [
        {
          "filterRef": "/templates/templat1/filters/filter1",
          "directives": [
            "log|none"
          ]
        }
      ]
    }
  ],
  "scope": "global|tenant|context|application-profile"
},
{
  "name": "contract2",
  "displayName": "Contract 2",
  "filters": "80",
  "scope": "global|tenant|context|application-profile"
},
{
  "displayName": "Contract 34",
  "name": "Contract34",
  "contractRef": "/templates/Templat1/contracts/Contract34",
  "scope": "global",

```

```

    "filterRelationships": [],
    "filterRelationshipsProviderToConsumer": [{
      "filterRef": "/templates/Templatel/filters/Filter4",
      "directives": ["log"]
    }],
    "filterRelationshipsConsumerToProvider": [{
      "filterRef": "/templates/Templatel/filters/Filter3",
      "directives": ["log"]
    }],
    "filterType": "oneWay"
  }
],
"filters": [
  {
    "name": "filter1",
    "displayName": "Filter 1",
    "description": "",
    "entries": [
      {
        "name": "filterEntry11",
        "displayName": "Filter Entry 11",
        "description": "",
        "etherType": "ip",
        "ipProtocol": "icmp"
      },
      {
        "name": "filterEntry12",
        "displayName": "Filter Entry 12",
        "description": "",
        "etherType": "ip",
        "ipProtocol": "udp",
        "matchOnlyFragments": false,
        "sourceFrom": "dns",
        "sourceTo": "http",
        "destinationFrom": "dns",
        "destinationTo": "80"
      }
    ]
  },
  {
    "name": "filter2",
    "displayName": "Filter 2",
    "description": "",
    "entries": [
      {
        "name": "filterEntry21",
        "displayName": "Filter Entry 21",
        "description": "",
        "etherType": "arp|fcoe|ip|mac_security|mpls_ucast|trill|unspecified",
        "arpFlag": "request|reply|unspecified",
        "ipProtocol":
"eigrp|egp|icmp|icmpv6|igmp|igp|l2tp|ospfigp|pim|tcp|udp|unspecified",
        "matchOnlyFragments": false,
        "stateful": false,
        "sourceFrom": "dns|ftp-data|http|https|pop3|rtsp|smtp|unspecified|[0..65535]",
        "sourceTo": "dns|ftp-data|http|https|pop3|rtsp|smtp|unspecified|[0..65535]",
        "destinationFrom":
"dns|ftp-data|http|https|pop3|rtsp|smtp|unspecified|[0..65535]",
        "destinationTo":
"dns|ftp-data|http|https|pop3|rtsp|smtp|unspecified|[0..65535]",
        "tcpSessionRules": [
          "acknowledgement|established|finish|synchronize|reset|unspecified",
          "acknowledgement|established|finish|synchronize|reset|unspecified"
        ]
      }
    ]
  }
]

```

```

    ]
  }
]
},{
  "displayName": "Filter 3",
  "name": "Filter3",
  "filterRef": "/templates/Template1/filters/Filter3",
  "entries": [{
    "etherType": "ip",
    "tcpSessionRules": ["unspecified"],
    "sourceFrom": "unspecified",
    "destinationTo": "unspecified",
    "arpFlag": "unspecified",
    "stateful": false,
    "displayName": "e1",
    "matchOnlyFragments": false,
    "sourceTo": "unspecified",
    "destinationFrom": "unspecified",
    "description": "",
    "ipProtocol": "unspecified",
    "name": "e1"
  }]
}, {
  "displayName": "Filter 4",
  "name": "Filter4",
  "filterRef": "/templates/Template1/filters/Filter4",
  "entries": [{
    "etherType": "ip",
    "tcpSessionRules": ["unspecified"],
    "sourceFrom": "unspecified",
    "destinationTo": "unspecified",
    "arpFlag": "unspecified",
    "stateful": false,
    "displayName": "e4",
    "matchOnlyFragments": false,
    "sourceTo": "unspecified",
    "destinationFrom": "unspecified",
    "description": "",
    "ipProtocol": "unspecified",
    "name": "e4"
  }]
}
],
"externalEpgs": [
  {
    "name": "Test-extEPG",
    "displayName": "Test-extEPG",
    "subnets": [
      {
        "ip": "2.2.2.2/24"
      }
    ],
    "contractRelationships": [
      {
        "relationshipType": "provider",
        "contractRef": "/templates/Template1/contracts/contract1"
      }
    ],
    "externalEpgRef": "/templates/Template1/externalEpgs/Test-extEPG"
  }
]
}
],
"sites": [

```

```

{
  "siteId": "58202f7066e6e10001c41236",
  "templateName": "template1",
  "bds": [
    {
      "bdRef": "/templates/template1/bds/bd1",
      "subnets": [ ... ]
    }
  ]
},
{
  "siteId": "69202f7066e6e10001c41247",
  "templateName": "template1",
  "anps": [
    {
      "anpRef": "/templates/template1/anps/anp1",
      "eggs": [
        {
          "epgRef": "/templates/template1/anps/anp1/eggs/epg1",
          "domainAssociations": [
            {
              "dn": "uni/vmmp-VMware/dom-TestVMMDomain-DoNotDelete",
              "domainType": "vmmDomain",
              "deployImmediacy": "On Demand|Immediate",
              "resolutionImmediacy": "On Demand|Immediate|Pre-provision",
              "allowMicroSegmentation": true,
              "allowPromiscuous": true,
              "macChanges": false,
              "allowForgedTransmits": true,
              "delimiter": "delimiter",
              "portEncapVlan": {
                "vlanType": "vlan",
                "vlan": 400
              },
              "deployImmediacy": "lazy",
              "microSegVlan": {
                "vlanType": "vlan",
                "vlan": 450
              },
              "vlanEncapMode": "static"
            },
            {
              "dn": "uni/phys-testPhysicalDomain-DoNotDelete",
              "domainType": "physicalDomain",
              "deployImmediacy": "On Demand|Immediate",
              "resolutionImmediacy": "On Demand|Immediate|Pre-provision"
            },
            {
              "dn": "uni/l2dom-TestL2Domain-DoNotDelete",
              "domainType": "l2ExtDomain",
              "deployImmediacy": "On Demand|Immediate",
              "resolutionImmediacy": "On Demand|Immediate|Pre-provision"
            },
            {
              "dn": "uni/l3dom-TestL3Domain-DoNotDelete",
              "domainType": "l3ExtDomain",
              "deployImmediacy": "On Demand|Immediate",
              "resolutionImmediacy": "On Demand|Immediate|Pre-provision"
            },
            {
              "dn": "uni/fc-TestFibreChannelDomain",
              "domainType": "fibreChannelDomain",
              "deployImmediacy": "On Demand|Immediate",
              "resolutionImmediacy": "On Demand|Immediate|Pre-provision"
            }
          ]
        }
      ]
    }
  ]
}

```



```

    }
  ],
  "staticPorts": [
    {
      "type": "port",
      "path": "topology/pod-1/paths-101/pathep[eth1/11]",
      "portEncapVlan": 10,
      "microSegVlan": 8,
      "deploymentImmediacy": "immediate|lazy",
      "mode": "regular|native|untagged"
    }, {
      "type": "VPC",
      "path": "Topology/pod-1/protpaths-101-102/pathep[NFS-Storage]",
      "portEncapVlan": 10,
      "microSegVlan": 8,
      "deploymentImmediacy": "Immediate",
      "mode": "Access"
    }, {
      "type": "DPC",
      "path": "...",
      "portEncapVlan": 9,
      "microSegVlan": 8,
      "deploymentImmediacy": "OnDemand",
      "mode": "AccessUntagged"
    }
  ],
  "staticLeafs": [
    {
      "path": "topology/pod-1/node-201",
      "portEncapVlan": 10
    }
  ]
}
]
}
],
"bds": [
  {
    "bdRef": "/templates/templatel/bds/bd1",
    "subnets": [ (see POST /templates) ]
  }
],
"contracts": [
  {
    "contractRef": "/templates/templatel/contracts/contract1",
    "serviceGraphTemplate": {
      "name": "SGT1"
    }
  }
],
"externalEpgs": [
  {
    "externalEpgRef": "/templates/Templatel/externalEpgs/Test-extEPG",
    "l3outDn": "uni/tn-Tenant/out-TestL3Out-site2",
    "subnets": [],
    "vrfRef": "templates/Templatel/vrfs/vrf1",
    "faults": []
  }
]
}
]
}

```

Response:

```

HTTP/1.1 201
Location: /api/v1/schemas/583c7c482501002501061985
{
  "id": "583c7c482501002501061985",
  "displayName": "Schema 1",
  "templates": [
    {
      "name": "template1",
      "displayName": "Template 1",
      "tenantId": "592c7c482501002501061578",
      "anps": [
        {
          "name": "anp1",
          "displayName": "Anp 1",
          "anpRef": "/schemas/583c7c482501002501061985/templates/template1/anps/anp1",
          "eggs": [
            {
              "name": "epg1",
              "displayName": "EPG 1",
              "eggRef":
"/schemas/583c7c482501002501061985/templates/template1/anps/anp1/eggs/epg1",
              "contractRelationships": [
                {
                  "contractRef":
"/schemas/583c7c482501002501061985/templates/template1/contracts/contract1",
                  "relationshipType": "provider|consumer"
                }
              ],
              "subnets": [ {
                "ip": "192.168.132.1/24",
                "description": "Subnet for IT department",
                "scope": "private"
                "shared": true
              }, {
                "ip": "192.168.133.1/24",
                "description": "Subnet for sales department",
                "scope": "public"
                "shared": false
              }
            ],
            "uSegEpg" : true,
            "uSegAttrs" : [ {
              "name" : "IP1",
              "displayName" : "...",
              "description" : "...",
              "type" : "ip",
              "fvSubnet" : true,
              "value" : "0.0.0.0"
            }, {
              "name" : "IP2",
              "displayName" : "...",
              "description" : "...",
              "type" : "ip",
              "fvSubnet" : false,
              "value" : "192.168.132.1"
            }, {
              "name" : "MAC",
              "displayName" : "...",
              "description" : "...",
              "type" : "mac",
              "value" : "aa:bb:cc:dd:ee:ff"
            }, {

```

```

        "name" : "VM Name",
        "displayName" : "...",
        "description" : "...",
        "type" : "vm-name",
        "operator" : "equals | contains | startsWith | endsWith",
        "value" : "..."
    }, {
        "name" : "VM Data Center",
        "displayName" : "...",
        "description" : "...",
        "type" : "rootContName",
        "operator" : "...",
        "value" : "..."
    }, {
        "name" : "VM Hypervisor Identifier",
        "displayName" : "...",
        "description" : "...",
        "type" : "hv",
        "operator" : "...",
        "value" : "..."
    }, {
        "name" : "VmOperatingSystem",
        "displayName" : "...",
        "description" : "...",
        "type" : "guest-os",
        "operator" : "...",
        "value" : "..."
    }, {
        "name" : "VM Tag",
        "displayName" : "...",
        "description" : "...",
        "type" : "tag",
        "operator" : "...",
        "category" : "...",
        "value" : "..."
    }, {
        "name" : "Vm Identifier",
        "displayName" : "...",
        "description" : "...",
        "type" : "vm",
        "operator" : "...",
        "value" : "..."
    }, {
        "name" : "Vm Vmm Domain",
        "displayName" : "...",
        "description" : "...",
        "type" : "domain",
        "operator" : "...",
        "value" : "..."
    }, {
        "name" : "Vm Vnic Dn",
        "displayName" : "...",
        "description" : "...",
        "type" : "vnic",
        "operator" : "...",
        "value" : "..."
    } ],
    "bdRef": "/schemas/583c7c482501002501061985/templates/template1/bds/bd1"
},
{
    "name": "epg2",
    "displayName": "EPG 2",
    "epgRef":
"/schema/583c7c482501002501061985/template/template1/anp/anp1/epg/epg2",

```

```

        "contractRelationships": [
          {
            "contractRef":
"/schemas/583c7c482501002501061985/templates/template1/contracts/contract2",
            "relationshipType": "provider|consumer"
          }
        ],
        "bdRef": "/schemas/583c7c482501002501061985/templates/template1/bds/bd1"
      }
    ]
  },
  "vrfs": [
    {
      "name": "vrf1",
      "displayName": "VRF 1",
      "vrfRef": "/schemas/583c7c482501002501061985/templates/template1/vrfs/vrf1"
    }
  ],
  "bds": [
    {
      "name": "bd1",
      "displayName": "BD 1",
      "bdRef": "/schemas/583c7c482501002501061985/templates/template1/bds/bd1",
      "l3UnknownMulticastFlooding": "flood|opt-flood",
      "intersiteBumTrafficAllow": true,
      "multiDestinationFlooding": "bd-flood|drop|encap-flood",
      "l2UnknownUnicast": "flood|proxy",
      "l2Stretch": true,
      "subnets": [ ... ],
      "vrfRef": "/schemas/583c7c482501002501061985/templates/template1/vrfs/vrf1"
    }
  ],
  "contracts": [
    {
      "name": "contract1",
      "displayName": "Contract 1",
      "contractRef":
"/schemas/583c7c482501002501061985/templates/template1/contracts/contract1",
      "filters": "80",
      "filterRelationships": [
        {
          "filterRef":
"/schemas/583c7c482501002501061985/templates/template1/filters/filter1",
          "directives": [
            "log"
          ]
        }
      ],
      "scope": "global|tenant|context|application-profile"
    },
    {
      "name": "contract2",
      "displayName": "Contract 2",
      "contractRef":
"/schemas/583c7c482501002501061985/templates/template1/contracts/contract2",
      "filters": "80",
      "filterRelationships": [],
      "scope": "global|tenant|context|anp"
    }
  ],
  "filters": [
    {
      "name": "filter1",

```

```

        "displayName": "Filter 1",
        "description": "",
        "filterRef":
"/schemas/583c7c482501002501061985/templates/template1/filters/filter1",
        "entries": [
            {
                "name": "filterEntry11",
                "displayName": "Filter Entry 11",
                "description": "",
                "etherType": "ip",
                "ipProtocol": "icmp"
            },
            {
                "name": "filterEntry12",
                "displayName": "Filter Entry 12",
                "description": "",
                "etherType": "ip",
                "ipProtocol": "udp",
                "matchOnlyFragments": false,
                "sourceFrom": "dns",
                "sourceTo": "http",
                "destinationFrom": "dns",
                "destinationTo": "http"
            }
        ]
    },
    {
        "name": "filter2",
        "displayName": "Filter 2",
        "description": "",
        "filterRef":
"/schemas/583c7c482501002501061985/templates/template1/filters/filter2",
        "entries": [...]
    }
],
"externalEpgs": [
    {
        "name": "Test-extEPG",
        "displayName": "Test-extEPG",
        "subnets": [
            {
                "ip": "2.2.2.2/24"
            }
        ],
        "contractRelationships": [
            {
                "relationshipType": "provider",
                "contractRef":
"/schemas/583c7c482501002501061985/templates/Template1/contracts/contract1"
            }
        ],
        "externalEpgRef":
"/schemas/583c7c482501002501061985/templates/Template1/externalEpgs/Test-extEPG"
    }
]
}
},
"sites": [
    {
        "siteId": "58202f7066e6e10001c41236",
        "templateName": "template1",
        "bds": [
            {
                "bdRef": "/schemas/583c7c482501002501061985/templates/template1/bds/bd1",

```

```

        "subnets": [ ... ]
    }
]
},
{
    "siteId": "69202f7066e6e10001c41247",
    "templateName": "template1",
    "anps": [
        {
            "anpRef": "/schemas/583c7c482501002501061985/templates/template1/anps/anp1",
            "eggs": [
                {
                    "epgRef":
"/schemas/583c7c482501002501061985/templates/template1/anps/anp1/eggs/epg1",
                    "domainAssociations": [
                        {
                            "dn": "uni/vmmp-VMware/dom-TestVMMDomain-DoNotDelete",
                            "domainType": "vmmDomain",
                            "deployImmediacy": "On Demand",
                            "resolutionImmediacy": "Immediate",
                            "allowMicroSegmentation": true,
                            "allowPromiscuous": true,
                            "macChanges": false,
                            "allowForgedTransmits": true,
                            "delimiter": "delimiter",
                            "microSegVlan": {
                                "vlanType": "vlan",
                                "vlan": 450
                            },
                            "portEncapVlan": {
                                "vlanType": "vlan",
                                "vlan": 400
                            },
                            "vlanEncapMode": "static"
                        },
                        {
                            "dn": "uni/phys-testPhysicalDomain-DoNotDelete",
                            "domainType": "physicalDomain",
                            "deployImmediacy": "On Demand",
                            "resolutionImmediacy": "Immediate"
                        },
                        {
                            "dn": "uni/l2dom-TestL2Domain-DoNotDelete",
                            "domainType": "l2ExtDomain",
                            "deployImmediacy": "On Demand",
                            "resolutionImmediacy": "Immediate"
                        },
                        {
                            "dn": "uni/l3dom-TestL3Domain-DoNotDelete",
                            "domainType": "l3ExtDomain",
                            "deployImmediacy": "On Demand",
                            "resolutionImmediacy": "Immediate"
                        },
                        {
                            "dn": "uni/fc-TestFibreChannelDomain",
                            "domainType": "fibreChannelDomain",
                            "deployImmediacy": "On Demand",
                            "resolutionImmediacy": "Immediate"
                        }
                    ],
                    "staticPorts": [
                        {
                            "type": "port",
                            "path": "topology/pod-1/paths-101/patchep[eth1/11]",

```

```

        "portEncapVlan": 10,
        "microSegVlan" : 8,
        "deploymentImmediacy" : "immediate|lazy",
        "mode" : "regular|native|untagged"
    }, {
        "type" : "VPC",
        "path" : "Topology/pod-1/protopaths-101-102/pathep[NFS-Storage]",
        "portEncapVlan": 10,
        "microSegVlan" : 8,
        "deploymentImmediacy" : "Immediate",
        "mode" : "Access"
    }, {
        "type" : "DPC",
        "path" : "...",
        "portEncapVlan": 9,
        "microSegVlan" : 8,
        "deploymentImmediacy" : "OnDemand",
        "mode" : "AccessUntagged"
    }
    ],
    "staticLeafs": [
        {
            "path" : "topology/pod-1/node-201",
            "portEncapVlan": 10
        }
    ]
    ]
    }
    ],
    "bds": [
        {
            "bdRef": "/schemas/583c7c482501002501061985/templates/template1/bds/bd1",
            "subnet": [ (see POST /templates) ]
        }
    ],
    "contracts": [
        {
            "contractRef":
"/schemas/583c7c482501002501061985/templates/template1/contracts/contract1",
            "serviceGraphTemplate": {
                "name": "SGT1"
            }
        }
    ],
    "externalEpgs": [
        {
            "externalEpgRef":
"/schemas/59a4824425000043017c65ea/templates/Template1/externalEpgs/Test-extEPG",
            "l3outDn": "uni/tn-Tenant/out-TestL3Out-site2",
            "subnets": [],
            "vrfRef": "/schemas/59a4824425000043017c65ea/templates/Template1/vrfs/vrf1",
            "faults": []
        }
    ]
    }
    ]
}

```

Example for ACI Multi-Site Release 1.0(2)

The following is an example of how to create a new Schema .

POST:

POST https://<host>:<port>/api/v1/schemas

Request:

```
{
  "displayName": "Schema 1",
  "templates": [
    {
      "name": "template1",
      "displayName": "Template 1",
      "tenantId": "592c7c482501002501061578",
      "anps": [
        {
          "name": "anp1",
          "displayName": "Anp 1",
          "egps": [
            {
              "name": "epg1",
              "displayName": "EPG 1",
              "contractRelationships": [
                {
                  "contractRef": "/templates/template1/contracts/contract1",
                  "relationshipType": "provider|consumer"
                }
              ]
            }
          ],
          "subnets": [ {
            "ip": "192.168.132.1/24",
            "description": "Subnet for IT department",
            "scope": "private"
            "shared": true
          }, {
            "ip": "192.168.133.1/24",
            "description": "Subnet for sales department",
            "scope": "public"
            "shared": false
          }
        ]
      },
      "uSegEpg": true,
      "uSegAttrs": [ {
        "name": "IP1",
        "displayName": "...",
        "description": "...",
        "type": "ip",
        "fvSubnet": true,
        "value": "0.0.0.0"
      }, {
        "name": "IP2",
        "displayName": "...",
        "description": "...",
        "type": "ip",
        "fvSubnet": false,
        "value": "192.168.132.1"
      }, {
        "name": "MAC",
        "displayName": "...",
        "description": "...",
        "type": "mac",
        "value": "aa:bb:cc:dd:ee:ff"
      }, {
        "name": "VM Name",
        "displayName": "...",
        "description": "...",
        "type": "vm-name",

```



```

        "operator" : "equals | contains | startsWith | endsWith",
        "value" : "...",
    }, {
        "name" : "VM Data Center",
        "displayName" : "...",
        "description" : "...",
        "type" : "rootContName",
        "operator" : "...",
        "value" : "...",
    }, {
        "name" : "VM Hypervisor Identifier",
        "displayName" : "...",
        "description" : "...",
        "type" : "hv",
        "operator" : "...",
        "value" : "...",
    }, {
        "name" : "VmOperatingSystem",
        "displayName" : "...",
        "description" : "...",
        "type" : "guest-os",
        "operator" : "...",
        "value" : "...",
    }, {
        "name" : "VM Tag",
        "displayName" : "...",
        "description" : "...",
        "type" : "tag",
        "operator" : "...",
        "category" : "...",
        "value" : "...",
    }, {
        "name" : "Vm Identifier",
        "displayName" : "...",
        "description" : "...",
        "type" : "vm",
        "operator" : "...",
        "value" : "...",
    }, {
        "name" : "Vm Vmm Domain",
        "displayName" : "...",
        "description" : "...",
        "type" : "domain",
        "operator" : "...",
        "value" : "...",
    }, {
        "name" : "Vm Vnic Dn",
        "displayName" : "...",
        "description" : "...",
        "type" : "vnic",
        "operator" : "...",
        "value" : "...",
    } ],
    "bdRef": "/templates/template1/bds/bd1"
},
{
    "name": "epg2",
    "displayName": "EPG 2",
    "contractRelationships": [
        {
            "contractRef": "/templates/template1/contracts/contract2",
            "relationshipType": "provider|consumer"
        }
    ]
},
],

```

```

        "bdRef": "templates/templatel/bds/bd1"
      }
    ]
  },
  "vrfs": [
    {
      "name": "vrf1",
      "displayName": "VRF 1"
    }
  ],
  "bds": [
    {
      "name": "bd1",
      "displayName": "BD 1",
      "l3UnknownMulticastFlooding": "flood|opt-flood",
      "intersiteBumTrafficAllow": true,
      "multiDestinationFlooding": "bd-flood|drop|encap-flood",
      "l2UnknownUnicast": "flood|proxy",
      "l2Stretch": true,
      "subnets": [ ... ],
      "vrfRef": "templates/templatel/vrfs/vrf1"
    }
  ],
  "contracts": [
    {
      "name": "contract1",
      "displayName": "Contract 1",
      "filters": "80",
      "filterRelationships": [
        {
          "filterRef": "/templates/templatel/filters/filter1",
          "directives": [
            "log|none"
          ]
        }
      ]
    },
    {
      "name": "contract2",
      "displayName": "Contract 2",
      "filters": "80",
      "scope": "global|tenant|context|application-profile"
    }
  ],
  "filters": [
    {
      "name": "filter1",
      "displayName": "Filter 1",
      "description": "",
      "entries": [
        {
          "name": "filterEntry11",
          "displayName": "Filter Entry 11",
          "description": "",
          "etherType": "ip",
          "ipProtocol": "icmp"
        },
        {
          "name": "filterEntry12",
          "displayName": "Filter Entry 12",
          "description": "",
          "etherType": "ip",

```

```

        "ipProtocol": "udp",
        "matchOnlyFragments": false,
        "sourceFrom": "dns",
        "sourceTo": "http",
        "destinationFrom": "dns",
        "destinationTo": "80"
    }
  ]
},
{
  "name": "filter2",
  "displayName": "Filter 2",
  "description": "",
  "entries": [
    {
      "name": "filterEntry21",
      "displayName": "Filter Entry 21",
      "description": "",
      "etherType": "arp|fcoe|ip|mac_security|mpls_ucast|trill|unspecified",
      "arpFlag": "request|reply|unspecified",
      "ipProtocol":
"eigrp|egp|icmp|icmpv6|igmp|igp|l2tp|ospfigp|pim|tcp|udp|unspecified",
      "matchOnlyFragments": false,
      "stateful": false,
      "sourceFrom": "dns|ftp-data|http|https|pop3|rtsp|smtp|unspecified|[0..65535]",
      "sourceTo": "dns|ftp-data|http|https|pop3|rtsp|smtp|unspecified|[0..65535]",
      "destinationFrom":
"dns|ftp-data|http|https|pop3|rtsp|smtp|unspecified|[0..65535]",
      "destinationTo":
"dns|ftp-data|http|https|pop3|rtsp|smtp|unspecified|[0..65535]",
      "tcpSessionRules": [
        "acknowledgement|established|finish|synchronize|reset|unspecified",
        "acknowledgement|established|finish|synchronize|reset|unspecified"
      ]
    }
  ]
}
],
"externalEpgs": [
  {
    "name": "Test-extEPG",
    "displayName": "Test-extEPG",
    "subnets": [
      {
        "ip": "2.2.2.2/24"
      }
    ],
    "contractRelationships": [
      {
        "relationshipType": "provider",
        "contractRef": "/templates/Template1/contracts/contract1"
      }
    ],
    "externalEpgRef": "/templates/Template1/externalEpgs/Test-extEPG"
  }
]
},
"sites": [
  {
    "siteId": "58202f7066e6e10001c41236",
    "templateName": "template1",
    "bds": [

```

```

    {
      "bdRef": "/templates/templatel/bds/bd1",
      "subnets": [ ... ]
    }
  ],
},
{
  "siteId": "69202f7066e6e10001c41247",
  "templateName": "templatel",
  "anps": [
    {
      "anpRef": "/templates/templatel/anps/anpl",
      "eggs": [
        {
          "eggRef": "/templates/templatel/anps/anpl/eggs/epgl",
          "domainAssociations": [
            {
              "dn": "uni/vmmp-VMware/dom-TestVMMDomain-DoNotDelete",
              "domainType": "vmmDomain",
              "deployImmediacy": "On Demand|Immediate",
              "resolutionImmediacy": "On Demand|Immediate|Pre-provision",
              "allowMicroSegmentation": true,
              "allowPromiscuous": true,
              "macChanges": false,
              "allowForgedTransmits": true,
              "delimiter": "delimiter",
            }
          ],
          "portEncapVlan": {
            "vlanType": "vlan",
            "vlan": 400
          },
          "deployImmediacy": "lazy",
          "microSegVlan": {
            "vlanType": "vlan",
            "vlan": 450
          },
        }
      ],
      "vlanEncapMode": "static"
    },
    {
      "dn": "uni/phys-testPhysicalDomain-DoNotDelete",
      "domainType": "physicalDomain",
      "deployImmediacy": "On Demand|Immediate",
      "resolutionImmediacy": "On Demand|Immediate|Pre-provision"
    },
    {
      "dn": "uni/l2dom-TestL2Domain-DoNotDelete",
      "domainType": "l2ExtDomain",
      "deployImmediacy": "On Demand|Immediate",
      "resolutionImmediacy": "On Demand|Immediate|Pre-provision"
    },
    {
      "dn": "uni/l3dom-TestL3Domain-DoNotDelete",
      "domainType": "l3ExtDomain",
      "deployImmediacy": "On Demand|Immediate",
      "resolutionImmediacy": "On Demand|Immediate|Pre-provision"
    },
    {
      "dn": "uni/fc-TestFibreChannelDomain",
      "domainType": "fibreChannelDomain",
      "deployImmediacy": "On Demand|Immediate",
      "resolutionImmediacy": "On Demand|Immediate|Pre-provision"
    }
  ],
  "staticPorts": [

```

```

    "type" : "port",
    "path" : "topology/pod-1/paths-101/pathep[eth1/11]",
    "portEncapVlan": 10,
    "microSegVlan" : 8,
    "deploymentImmediacy" : "immediate|lazy",
    "mode" : "regular|native|untagged"
  }, {
    "type" : "VPC",
    "path" : "Topology/pod-1/protpaths-101-102/pathep[NFS-Storage]",
    "portEncapVlan": 10,
    "microSegVlan" : 8,
    "deploymentImmediacy" : "Immediate",
    "mode" : "Access"
  }, {
    "type" : "DPC",
    "path" : "...",
    "portEncapVlan": 9,
    "microSegVlan" : 8,
    "deploymentImmediacy" : "OnDemand",
    "mode" : "AccessUntagged"
  }
],
"staticLeafs": [
{
  "path" : "topology/pod-1/node-201",
  "portEncapVlan": 10
}
]
}
],
"bds": [
{
  "bdRef": "/templates/templatel/bds/bd1",
  "subnets": [ (see POST /templates) ]
}
],
"contracts": [
{
  "contractRef": "/templates/templatel/contracts/contract1",
  "serviceGraphTemplate": {
    "name": "SGT1"
  }
}
],
"externalEpgs": [
{
  "externalEpgRef": "/templates/Template1/externalEpgs/Test-extEPG",
  "l3outDn": "uni/tn-Tenant/out-TestL3Out-site2",
  "subnets": [],
  "vrfRef": "templates/Template1/vrfs/vrf1",
  "faults": []
}
]
}
]
}
}

```

Response:

```

HTTP/1.1 201
Location: /api/v1/schemas/583c7c482501002501061985
{
  "id": "583c7c482501002501061985",

```

```

"displayName": "Schema 1",
"templates": [
  {
    "name": "template1",
    "displayName": "Template 1",
    "tenantId": "592c7c482501002501061578",
    "anps": [
      {
        "name": "anp1",
        "displayName": "Anp 1",
        "anpRef": "/schemas/583c7c482501002501061985/templates/template1/anps/anp1",
        "eggs": [
          {
            "name": "epg1",
            "displayName": "EPG 1",
            "eggRef":
"/schemas/583c7c482501002501061985/templates/template1/anps/anp1/eggs/epg1",
            "contractRelationships": [
              {
                "contractRef":
"/schemas/583c7c482501002501061985/templates/template1/contracts/contract1",
                "relationshipType": "provider|consumer"
              }
            ],
            "subnets": [ {
              "ip": "192.168.132.1/24",
              "description": "Subnet for IT department",
              "scope": "private"
            }, {
              "ip": "192.168.133.1/24",
              "description": "Subnet for sales department",
              "scope": "public"
            }, {
              "ip": "192.168.134.1/24",
              "description": "Subnet for finance department",
              "scope": "public"
            }
          ],
          "uSegEpg" : true,
          "uSegAttrs" : [ {
            "name" : "IP1",
            "displayName" : "...",
            "description" : "...",
            "type" : "ip",
            "fvSubnet" : true,
            "value" : "0.0.0.0"
          }, {
            "name" : "IP2",
            "displayName" : "...",
            "description" : "...",
            "type" : "ip",
            "fvSubnet" : false,
            "value" : "192.168.132.1"
          }, {
            "name" : "MAC",
            "displayName" : "...",
            "description" : "...",
            "type" : "mac",
            "value" : "aa:bb:cc:dd:ee:ff"
          }, {
            "name" : "VM Name",
            "displayName" : "...",
            "description" : "...",
            "type" : "vm-name",
            "operator" : "equals | contains | startsWith | endsWith",
            "value" : "..."
          }, {

```

```

        "name" : "VM Data Center",
        "displayName" : "...",
        "description" : "...",
        "type" : "rootContName",
        "operator" : "...",
        "value" : "..."
    }, {
        "name" : "VM Hypervisor Identifier",
        "displayName" : "...",
        "description" : "...",
        "type" : "hv",
        "operator" : "...",
        "value" : "..."
    }, {
        "name" : "VmOperatingSystem",
        "displayName" : "...",
        "description" : "...",
        "type" : "guest-os",
        "operator" : "...",
        "value" : "..."
    }, {
        "name" : "VM Tag",
        "displayName" : "...",
        "description" : "...",
        "type" : "tag",
        "operator" : "...",
        "category" : "...",
        "value" : "..."
    }, {
        "name" : "Vm Identifier",
        "displayName" : "...",
        "description" : "...",
        "type" : "vm",
        "operator" : "...",
        "value" : "..."
    }, {
        "name" : "Vm Vmm Domain",
        "displayName" : "...",
        "description" : "...",
        "type" : "domain",
        "operator" : "...",
        "value" : "..."
    }, {
        "name" : "Vm Vnic Dn",
        "displayName" : "...",
        "description" : "...",
        "type" : "vnic",
        "operator" : "...",
        "value" : "..."
    } ],
    "bdRef": "/schemas/583c7c482501002501061985/templates/template1/bds/bd1"
},
{
    "name": "epg2",
    "displayName": "EPG 2",
    "epgRef":
"/schema/583c7c482501002501061985/template/template1/anp/anp1/epg/epg2",
    "contractRelationships": [
        {
            "contractRef":
"/schemas/583c7c482501002501061985/templates/template1/contracts/contract2",
            "relationshipType": "provider|consumer"
        }
    ]
},
],

```

```

        "bdRef": "/schemas/583c7c482501002501061985/templates/template1/bds/bd1"
    }
}
],
"vrfs": [
    {
        "name": "vrf1",
        "displayName": "VRF 1",
        "vrfRef": "/schemas/583c7c482501002501061985/templates/template1/vrfs/vrf1"
    }
],
"bds": [
    {
        "name": "bd1",
        "displayName": "BD 1",
        "bdRef": "/schemas/583c7c482501002501061985/templates/template1/bds/bd1",
        "l3UnknownMulticastFlooding": "flood|opt-flood",
        "intersiteBumTrafficAllow": true,
        "multiDestinationFlooding": "bd-flood|drop|encap-flood",
        "l2UnknownUnicast": "flood|proxy",
        "l2Stretch": true,
        "subnets": [ ... ],
        "vrfRef": "/schemas/583c7c482501002501061985/templates/template1/vrfs/vrf1"
    }
],
"contracts": [
    {
        "name": "contract1",
        "displayName": "Contract 1",
        "contractRef":
"/schemas/583c7c482501002501061985/templates/template1/contracts/contract1",
        "filters": "80",
        "filterRelationships": [
            {
                "filterRef":
"/schemas/583c7c482501002501061985/templates/template1/filters/filter1",
                "directives": [
                    "log"
                ]
            }
        ]
    },
    {
        "name": "contract2",
        "displayName": "Contract 2",
        "contractRef":
"/schemas/583c7c482501002501061985/templates/template1/contracts/contract2",
        "filters": "80",
        "filterRelationships": [],
        "scope": "global|tenant|context|anp"
    }
],
"filters": [
    {
        "name": "filter1",
        "displayName": "Filter 1",
        "description": "",
        "filterRef":
"/schemas/583c7c482501002501061985/templates/template1/filters/filter1",
        "entries": [
            {
                "name": "filterEntry11",

```



```

        "displayName": "Filter Entry 11",
        "description": "",
        "etherType": "ip",
        "ipProtocol": "icmp"
    },
    {
        "name": "filterEntry12",
        "displayName": "Filter Entry 12",
        "description": "",
        "etherType": "ip",
        "ipProtocol": "udp",
        "matchOnlyFragments": false,
        "sourceFrom": "dns",
        "sourceTo": "http",
        "destinationFrom": "dns",
        "destinationTo": "http"
    }
  ],
  },
  {
    "name": "filter2",
    "displayName": "Filter 2",
    "description": "",
    "filterRef":
"/schemas/583c7c482501002501061985/templates/template1/filters/filter2",
    "entries": [...]
  }
],
"externalEpgs": [
  {
    "name": "Test-extEPG",
    "displayName": "Test-extEPG",
    "subnets": [
      {
        "ip": "2.2.2.2/24"
      }
    ],
    "contractRelationships": [
      {
        "relationshipType": "provider",
        "contractRef":
"/schemas/583c7c482501002501061985/templates/Template1/contracts/contract1"
      }
    ],
    "externalEpgRef":
"/schemas/583c7c482501002501061985/templates/Template1/externalEpgs/Test-extEPG"
  }
]
},
"sites": [
  {
    "siteId": "58202f7066e6e10001c41236",
    "templateName": "template1",
    "bds": [
      {
        "bdRef": "/schemas/583c7c482501002501061985/templates/template1/bds/bd1",
        "subnets": [ ... ]
      }
    ]
  },
  {
    "siteId": "69202f7066e6e10001c41247",
    "templateName": "template1",

```

```

"anps": [
  {
    "anpRef": "/schemas/583c7c482501002501061985/templates/template1/anps/anp1",
    "eggs": [
      {
        "epgRef":
"/schemas/583c7c482501002501061985/templates/template1/anps/anp1/eggs/epg1",
        "domainAssociations": [
          {
            "dn": "uni/vmmp-VMware/dom-TestVMMDomain-DoNotDelete",
            "domainType": "vmmDomain",
            "deployImmediacy": "On Demand",
            "resolutionImmediacy": "Immediate",
            "allowMicroSegmentation": true,
            "allowPromiscuous": true,
            "macChanges": false,
            "allowForgedTransmits": true,
            "delimiter": "delimiter",
            "microSegVlan": {
              "vlanType": "vlan",
              "vlan": 450
            },
            "portEncapVlan": {
              "vlanType": "vlan",
              "vlan": 400
            },
            "vlanEncapMode": "static"
          },
          {
            "dn": "uni/phys-testPhysicalDomain-DoNotDelete",
            "domainType": "physicalDomain",
            "deployImmediacy": "On Demand",
            "resolutionImmediacy": "Immediate"
          },
          {
            "dn": "uni/l2dom-TestL2Domain-DoNotDelete",
            "domainType": "l2ExtDomain",
            "deployImmediacy": "On Demand",
            "resolutionImmediacy": "Immediate"
          },
          {
            "dn": "uni/l3dom-TestL3Domain-DoNotDelete",
            "domainType": "l3ExtDomain",
            "deployImmediacy": "On Demand",
            "resolutionImmediacy": "Immediate"
          },
          {
            "dn": "uni/fc-TestFibreChannelDomain",
            "domainType": "fibreChannelDomain",
            "deployImmediacy": "On Demand",
            "resolutionImmediacy": "Immediate"
          }
        ],
        "staticPorts": [
          {
            "type": "port",
            "path": "topology/pod-1/paths-101/pathep[eth1/11]",
            "portEncapVlan": 10,
            "microSegVlan": 8,
            "deploymentImmediacy": "immediate|lazy",
            "mode": "regular|native|untagged"
          },
          {
            "type": "VPC",
            "path": "Topology/pod-1/protpaths-101-102/pathep[NFS-Storage]",

```

```

        "portEncapVlan": 10,
        "microSegVlan" : 8,
        "deploymentImmediacy" : "Immediate",
        "mode" : "Access"
    }, {
        "type" : "DPC",
        "path" : "...",
        "portEncapVlan": 9,
        "microSegVlan" : 8,
        "deploymentImmediacy" : "OnDemand",
        "mode" : "AccessUntagged"
    }
    ],
    "staticLeafs": [
    {
        "path" : "topology/pod-1/node-201",
        "portEncapVlan": 10
    }
    ]
    }
    ]
    },
    "bds": [
    {
        "bdRef": "/schemas/583c7c482501002501061985/templates/template1/bds/bd1",
        "subnet": [ (see POST /templates) ]
    }
    ],
    "contracts": [
    {
        "contractRef":
"/schemas/583c7c482501002501061985/templates/template1/contracts/contract1",
        "serviceGraphTemplate": {
            "name": "SGT1"
        }
    }
    ],
    "externalEpgs": [
    {
        "externalEpgRef":
"/schemas/59a4824425000043017c65ea/templates/Template1/externalEpgs/Test-extEPG",
        "l3outDn": "uni/tn-Tenant/out-TestL3Out-site2",
        "subnets": [],
        "vrfRef": "/schemas/59a4824425000043017c65ea/templates/Template1/vrfs/vrf1",
        "faults": []
    }
    ]
    }
    ]
    }
}

```

Example for ACI Multi-Site Release 1.0(1)

The following is an example of how to create a new Schema .

POST:

POST <https://<host>:<port>/api/v1/schemas>

Request:

```

{
  "displayName": "Schema 1",
  "templates": [
    {
      "name": "template1",
      "displayName": "Template 1",
      "tenantId": "592c7c482501002501061578",
      "anps": [
        {
          "name": "anp1",
          "displayName": "Anp 1",
          "eggs": [
            {
              "name": "epg1",
              "displayName": "EPG 1",
              "contractRelationships": [
                {
                  "contractRef": "/templates/template1/contracts/contract1",
                  "relationshipType": "provider|consumer"
                }
              ]
            },
            {
              "name": "subnets": [ {
                "ip": "192.168.132.1/24",
                "description": "Subnet for IT department",
                "scope": "private",
                "shared": true
              }, {
                "ip": "192.168.133.1/24",
                "description": "Subnet for sales department",
                "scope": "public",
                "shared": false
              }
            ]
          ],
          "uSegEpg": true,
          "uSegAttrs": [ {
            "name": "IP1",
            "displayName": "...",
            "description": "...",
            "type": "ip",
            "fvSubnet": true,
            "value": "0.0.0.0"
          }, {
            "name": "IP2",
            "displayName": "...",
            "description": "...",
            "type": "ip",
            "fvSubnet": false,
            "value": "192.168.132.1"
          }, {
            "name": "MAC",
            "displayName": "...",
            "description": "...",
            "type": "mac",
            "value": "aa:bb:cc:dd:ee:ff"
          }, {
            "name": "VM Name",
            "displayName": "...",
            "description": "...",
            "type": "vm-name",
            "operator": "equals | contains | startsWith | endsWith",
            "value": "..."
          }, {
            "name": "VM Data Center",
            "displayName": "...",

```

```

        "description" : "...",
        "type" : "rootContName",
        "operator" : "...",
        "value" : "..."
    }, {
        "name" : "VM Hypervisor Identifier",
        "displayName" : "...",
        "description" : "...",
        "type" : "hv",
        "operator" : "...",
        "value" : "..."
    }, {
        "name" : "VmOperatingSystem",
        "displayName" : "...",
        "description" : "...",
        "type" : "guest-os",
        "operator" : "...",
        "value" : "..."
    }, {
        "name" : "VM Tag",
        "displayName" : "...",
        "description" : "...",
        "type" : "tag",
        "operator" : "...",
        "category" : "...",
        "value" : "..."
    }, {
        "name" : "Vm Identifier",
        "displayName" : "...",
        "description" : "...",
        "type" : "vm",
        "operator" : "...",
        "value" : "..."
    }, {
        "name" : "Vm Vmm Domain",
        "displayName" : "...",
        "description" : "...",
        "type" : "domain",
        "operator" : "...",
        "value" : "..."
    }, {
        "name" : "Vm Vnic Dn",
        "displayName" : "...",
        "description" : "...",
        "type" : "vnic",
        "operator" : "...",
        "value" : "..."
    } ],
    "bdRef": "/templates/templatel/bds/bd1"
},
{
    "name": "epg2",
    "displayName": "EPG 2",
    "contractRelationships": [
        {
            "contractRef": "/templates/templatel/contracts/contract2",
            "relationshipType": "provider|consumer"
        }
    ],
    "bdRef": "templates/templatel/bds/bd1"
}
]
}
],

```

```

"vrfs": [
  {
    "name": "vrf1",
    "displayName": "VRF 1"
  }
],
"bds": [
  {
    "name": "bd1",
    "displayName": "BD 1",
    "l3UnknownMulticastFlooding": "flood|opt-flood",
    "intersiteBumTrafficAllow": true,
    "multiDestinationFlooding": "bd-flood|drop|encap-flood",
    "l2UnknownUnicast": "flood|proxy",
    "l2Stretch": true,
    "subnets": [ ... ],
    "vrfRef": "templates/templatel/vrfs/vrf1"
  }
],
"contracts": [
  {
    "name": "contract1",
    "displayName": "Contract 1",
    "filters": "80",
    "filterRelationships": [
      {
        "filterRef": "/templates/templatel/filters/filter1",
        "directives": [
          "log|none"
        ]
      }
    ]
  },
  {
    "name": "contract2",
    "displayName": "Contract 2",
    "filters": "80",
    "scope": "global|tenant|context|application-profile"
  }
],
"filters": [
  {
    "name": "filter1",
    "displayName": "Filter 1",
    "description": "",
    "entries": [
      {
        "name": "filterEntry11",
        "displayName": "Filter Entry 11",
        "description": "",
        "etherType": "ip",
        "ipProtocol": "icmp"
      },
      {
        "name": "filterEntry12",
        "displayName": "Filter Entry 12",
        "description": "",
        "etherType": "ip",
        "ipProtocol": "udp",
        "matchOnlyFragments": false,
        "sourceFrom": "dns",
        "sourceTo": "http",
        "destinationFrom": "dns",

```

```

        "destinationTo": "80"
      }
    ]
  },
  {
    "name": "filter2",
    "displayName": "Filter 2",
    "description": "",
    "entries": [
      {
        "name": "filterEntry21",
        "displayName": "Filter Entry 21",
        "description": "",
        "etherType": "arp|fcoe|ip|mac_security|mpls_ucast|trill|unspecified",
        "arpFlag": "request|reply|unspecified",
        "ipProtocol":
"eigrp|egp|icmp|icmpv6|igmp|igp|l2tp|ospfigp|pim|tcp|udp|unspecified",
        "matchOnlyFragments": false,
        "stateful": false,
        "sourceFrom": "dns|ftp-data|http|https|pop3|rtsp|smtp|unspecified|[0..65535]",
        "sourceTo": "dns|ftp-data|http|https|pop3|rtsp|smtp|unspecified|[0..65535]",
        "destinationFrom":
"dns|ftp-data|http|https|pop3|rtsp|smtp|unspecified|[0..65535]",
        "destinationTo":
"dns|ftp-data|http|https|pop3|rtsp|smtp|unspecified|[0..65535]",
        "tcpSessionRules": [
          "acknowledgement|established|finish|synchronize|reset|unspecified",
          "acknowledgement|established|finish|synchronize|reset|unspecified"
        ]
      }
    ]
  }
]
},
"sites": [
  {
    "siteId": "58202f7066e6e10001c41236",
    "templateName": "template1",
    "bds": [
      {
        "bdRef": "/templates/template1/bds/bd1",
        "subnets": [ ... ]
      }
    ]
  },
  {
    "siteId": "69202f7066e6e10001c41247",
    "templateName": "template1",
    "anps": [
      {
        "anpRef": "/templates/template1/anps/anp1",
        "epgs": [
          {
            "epgRef": "/templates/template1/anps/anp1/epgs/epg1",
            "domainAssociations": [
              {
                "dn": "uni/vmmp-VMware/dom-TestVMMDomain-DoNotDelete",
                "domainType": "vmmDomain",
                "deployImmediacy": "On Demand|Immediate",
                "resolutionImmediacy": "On Demand|Immediate|Pre-provision",
                "allowMicroSegmentation": true,
                "allowPromiscuous": true,

```

```

"macChanges": false,
"allowForgedTransmits": true,
"delimiter": "delimiter",
"portEncapVlan": {
  "vlanType": "vlan",
  "vlan": 400
},
"deployImmediacy": "lazy",
"microSegVlan": {
  "vlanType": "vlan",
  "vlan": 450
},
"vlanEncapMode": "static"
},
{
  "dn": "uni/phys-testPhysicalDomain-DoNotDelete",
  "domainType": "physicalDomain",
  "deployImmediacy": "On Demand|Immediate",
  "resolutionImmediacy": "On Demand|Immediate|Pre-provision"
},
{
  "dn": "uni/l2dom-TestL2Domain-DoNotDelete",
  "domainType": "l2ExtDomain",
  "deployImmediacy": "On Demand|Immediate",
  "resolutionImmediacy": "On Demand|Immediate|Pre-provision"
},
{
  "dn": "uni/l3dom-TestL3Domain-DoNotDelete",
  "domainType": "l3ExtDomain",
  "deployImmediacy": "On Demand|Immediate",
  "resolutionImmediacy": "On Demand|Immediate|Pre-provision"
},
{
  "dn": "uni/fc-TestFibreChannelDomain",
  "domainType": "fibreChannelDomain",
  "deployImmediacy": "On Demand|Immediate",
  "resolutionImmediacy": "On Demand|Immediate|Pre-provision"
}
],
"staticPorts": [
  {
    "type": "port",
    "path": "topology/pod-1/paths-101/pathep[eth1/11]",
    "portEncapVlan": 10,
    "microSegVlan": 8,
    "deploymentImmediacy": "immediate|lazy",
    "mode": "regular|native|untagged"
  },
  {
    "type": "VPC",
    "path": "Topology/pod-1/protpaths-101-102/pathep[NFS-Storage]",
    "portEncapVlan": 10,
    "microSegVlan": 8,
    "deploymentImmediacy": "Immediate",
    "mode": "Access"
  },
  {
    "type": "DPC",
    "path": "...",
    "portEncapVlan": 9,
    "microSegVlan": 8,
    "deploymentImmediacy": "OnDemand",
    "mode": "AccessUntagged"
  }
],
"staticLeafs": [

```



```

        {
            "path" : "topology/pod-1/node-201",
            "portEncapVlan": 10
        }
    ]
}
]
},
"bds": [
    {
        "bdRef": "/templates/template1/bds/bd1",
        "subnets": [ (see POST /templates) ]
    }
],
"contracts": [
    {
        "contractRef": "/templates/template1/contracts/contract1",
        "serviceGraphTemplate": {
            "name": "SGT1"
        }
    }
]
}
]
}
}

```

Response:

```

HTTP/1.1 201
Location: /api/v1/schemas/583c7c482501002501061985
{
  "id": "583c7c482501002501061985",
  "displayName": "Schema 1",
  "templates": [
    {
      "name": "template1",
      "displayName": "Template 1",
      "tenantId": "592c7c482501002501061578",
      "anps": [
        {
          "name": "anp1",
          "displayName": "Anp 1",
          "anpRef": "/schemas/583c7c482501002501061985/templates/template1/anps/anp1",
          "epgs": [
            {
              "name": "epg1",
              "displayName": "EPG 1",
              "epgRef":
"/schemas/583c7c482501002501061985/templates/template1/anps/anp1/epgs/epg1",
              "contractRelationships": [
                {
                  "contractRef":
"/schemas/583c7c482501002501061985/templates/template1/contracts/contract1",
                  "relationshipType": "provider|consumer"
                }
              ],
              "subnets": [ {
                "ip": "192.168.132.1/24",
                "description": "Subnet for IT department",
                "scope" : "private"
                "shared": true
              }, {
                "ip": "192.168.133.1/24",

```

```

        "description": "Subnet for sales department",
        "scope" : "public"
        "shared": false
    }],
    "uSegEpg" : true,
    "uSegAttrs" : [ {
        "name" : "IP1",
        "displayName" : "...",
        "description" : "...",
        "type" : "ip",
        "fvSubnet" : true,
        "value" : "0.0.0.0"
    }, {
        "name" : "IP2",
        "displayName" : "...",
        "description" : "...",
        "type" : "ip",
        "fvSubnet" : false,
        "value" : "192.168.132.1"
    }, {
        "name" : "MAC",
        "displayName" : "...",
        "description" : "...",
        "type" : "mac",
        "value" : "aa:bb:cc:dd:ee:ff"
    }, {
        "name" : "VM Name",
        "displayName" : "...",
        "description" : "...",
        "type" : "vm-name",
        "operator" : "equals | contains | startsWith | endsWith",
        "value" : "..."
    }, {
        "name" : "VM Data Center",
        "displayName" : "...",
        "description" : "...",
        "type" : "rootContName",
        "operator" : "...",
        "value" : "..."
    }, {
        "name" : "VM Hypervisor Identifier",
        "displayName" : "...",
        "description" : "...",
        "type" : "hv",
        "operator" : "...",
        "value" : "..."
    }, {
        "name" : "VmOperatingSystem",
        "displayName" : "...",
        "description" : "...",
        "type" : "guest-os",
        "operator" : "...",
        "value" : "..."
    }, {
        "name" : "VM Tag",
        "displayName" : "...",
        "description" : "...",
        "type" : "tag",
        "operator" : "...",
        "category" : "...",
        "value" : "..."
    }, {
        "name" : "Vm Identifier",
        "displayName" : "...",

```

```

        "description" : "...",
        "type" : "vm",
        "operator" : "...",
        "value" : "..."
    }, {
        "name" : "Vm Vmm Domain",
        "displayName" : "...",
        "description" : "...",
        "type" : "domain",
        "operator" : "...",
        "value" : "..."
    }, {
        "name" : "Vm Vnic Dn",
        "displayName" : "...",
        "description" : "...",
        "type" : "vnic",
        "operator" : "...",
        "value" : "..."
    } ],
    "bdRef": "/schemas/583c7c482501002501061985/templates/template1/bds/bd1"
},
{
    "name": "epg2",
    "displayName": "EPG 2",
    "epgRef":
"/schema/583c7c482501002501061985/template/template1/anp/anp1/epg/epg2",
    "contractRelationships": [
        {
            "contractRef":
"/schemas/583c7c482501002501061985/templates/template1/contracts/contract2",
            "relationshipType": "provider|consumer"
        }
    ],
    "bdRef": "/schemas/583c7c482501002501061985/templates/template1/bds/bd1"
}
]
},
"vrfs": [
{
    "name": "vrf1",
    "displayName": "VRF 1",
    "vrfRef": "/schemas/583c7c482501002501061985/templates/template1/vrfs/vrf1"
}
],
"bds": [
{
    "name": "bd1",
    "displayName": "BD 1",
    "bdRef": "/schemas/583c7c482501002501061985/templates/template1/bds/bd1",
    "l3UnknownMulticastFlooding": "flood|opt-flood",
    "intersiteBumTrafficAllow": true,
    "multiDestinationFlooding": "bd-flood|drop|encap-flood",
    "l2UnknownUnicast": "flood|proxy",
    "l2Stretch": true,
    "subnets": [ ... ],
    "vrfRef": "/schemas/583c7c482501002501061985/templates/template1/vrfs/vrf1"
}
],
"contracts": [
{
    "name": "contract1",
    "displayName": "Contract 1",
    "contractRef":

```

```

"/schemas/583c7c482501002501061985/templates/templatel/contracts/contract1",
  "filters": "80",
  "filterRelationships": [
    {
      "filterRef":
"/schemas/583c7c482501002501061985/templates/templatel/filters/filter1",
      "directives": [
        "log"
      ]
    }
  ],
  "scope": "global|tenant|context|application-profile"
},
{
  "name": "contract2",
  "displayName": "Contract 2",
  "contractRef":
"/schemas/583c7c482501002501061985/templates/templatel/contracts/contract2",
  "filters": "80",
  "filterRelationships": [],
  "scope": "global|tenant|context|anp"
}
],

"filters": [
  {
    "name": "filter1",
    "displayName": "Filter 1",
    "description": "",
    "filterRef":
"/schemas/583c7c482501002501061985/templates/templatel/filters/filter1",
    "entries": [
      {
        "name": "filterEntry11",
        "displayName": "Filter Entry 11",
        "description": "",
        "etherType": "ip",
        "ipProtocol": "icmp"
      },
      {
        "name": "filterEntry12",
        "displayName": "Filter Entry 12",
        "description": "",
        "etherType": "ip",
        "ipProtocol": "udp",
        "matchOnlyFragments": false,
        "sourceFrom": "dns",
        "sourceTo": "http",
        "destinationFrom": "dns",
        "destinationTo": "http"
      }
    ]
  },
  {
    "name": "filter2",
    "displayName": "Filter 2",
    "description": "",
    "filterRef":
"/schemas/583c7c482501002501061985/templates/templatel/filters/filter2",
    "entries": [...]
  }
]
],

```

```

"sites": [
  {
    "siteId": "58202f7066e6e10001c41236",
    "templateName": "template1",
    "bds": [
      {
        "bdRef": "/schemas/583c7c482501002501061985/templates/template1/bds/bd1",
        "subnets": [ ... ]
      }
    ]
  },
  {
    "siteId": "69202f7066e6e10001c41247",
    "templateName": "template1",
    "anps": [
      {
        "anpRef": "/schemas/583c7c482501002501061985/templates/template1/anps/anp1",
        "epgs": [
          {
            "epgRef":
"/schemas/583c7c482501002501061985/templates/template1/anps/anp1/epgs/epg1",
            "domainAssociations": [
              {
                "dn": "uni/vmmp-VMware/dom-TestVMMDomain-DoNotDelete",
                "domainType": "vmmDomain",
                "deployImmediacy": "On Demand",
                "resolutionImmediacy": "Immediate",
                "allowMicroSegmentation": true,
                "allowPromiscuous": true,
                "macChanges": false,
                "allowForgedTransmits": true,
                "delimiter": "delimiter",
                "microSegVlan": {
                  "vlanType": "vlan",
                  "vlan": 450
                },
                "portEncapVlan": {
                  "vlanType": "vlan",
                  "vlan": 400
                },
                "vlanEncapMode": "static"
              },
              {
                "dn": "uni/phys-testPhysicalDomain-DoNotDelete",
                "domainType": "physicalDomain",
                "deployImmediacy": "On Demand",
                "resolutionImmediacy": "Immediate"
              },
              {
                "dn": "uni/l2dom-TestL2Domain-DoNotDelete",
                "domainType": "l2ExtDomain",
                "deployImmediacy": "On Demand",
                "resolutionImmediacy": "Immediate"
              },
              {
                "dn": "uni/l3dom-TestL3Domain-DoNotDelete",
                "domainType": "l3ExtDomain",
                "deployImmediacy": "On Demand",
                "resolutionImmediacy": "Immediate"
              }
            ],
            "vlanEncapMode": "static"
          }
        ]
      }
    ]
  }
]

```

```

        "resolutionImmediacy": "Immediate"
    }
  ],
  "staticPorts": [
    {
      "type": "port",
      "path": "topology/pod-1/paths-101/pathep[eth1/11]",
      "portEncapVlan": 10,
      "microSegVlan": 8,
      "deploymentImmediacy": "immediate|lazy",
      "mode": "regular|native|untagged"
    }, {
      "type": "vPC",
      "path": "Topology/pod-1/protpaths-101-102/pathep[NFS-Storage]",
      "portEncapVlan": 10,
      "microSegVlan": 8,
      "deploymentImmediacy": "Immediate",
      "mode": "Access"
    }, {
      "type": "DPC",
      "path": "...",
      "portEncapVlan": 9,
      "microSegVlan": 8,
      "deploymentImmediacy": "OnDemand",
      "mode": "AccessUntagged"
    }
  ],
  "staticLeafs": [
    {
      "path": "topology/pod-1/node-201",
      "portEncapVlan": 10
    }
  ]
}
],
"bds": [
  {
    "bdRef": "/schemas/583c7c482501002501061985/templates/template1/bds/bd1",
    "subnet": [ (see POST /templates) ]
  }
],
"contracts": [
  {
    "contractRef":
"/schemas/583c7c482501002501061985/templates/template1/contracts/contract1",
    "serviceGraphTemplate": {
      "name": "SGT1"
    }
  }
]
}
]
}

```

Read a Schema

Example for ACI Multi-Site Release 1.0(2)

The following is an example for reading a Schema.

GET

GET https://<host>:<port>/api/v1/schemas/583c7c482501002501061985

REQUEST

GET

Request Body:
<empty>

RESPONSE

```
HTTP/1.1 200
{
  "associations": [
    {
      "templateName": "template1",
      "bdAssociations": [
        {
          "bdRef": "/schemas/583c7c482501002501061985/templates/template1/bds/bd1",
          "epgIdentities": [
            {
              "epgRef": "/schemas/583c7c482501002501061986/templates/template1/epgs/epg1",
              "schemaName": "Schema 1",
              "templateName": "Template 1"
            },
            {
              "displayName": "EPG1"
            }
          ],
          ...
        }
      ],
      ...
    },
    ...
  ],
  "vrfAssociations": [
    {
      "vrfRef": "/schemas/583c7c482501002501061985/templates/template1/vrfs/vrf1",
      "bdIdentities": [
        {
          "bdRef": "/schemas/583c7c482501002501061986/templates/template1/bds/bd1",
          "schemaName": "Schema 1",
          "templateName": "Template 1"
        },
        {
          "displayName": "BD 1"
        }
      ],
      ...
    },
    ...
  ],
  "externalEpgIdentities": [
    {
      "externalEpgRef":
"/schemas/583c7c482501002501061986/templates/template1/externalEpgs/Test-extEPG",
      "schemaName": "schema1",
      "templateName": "template1",
      "displayName": "Test-extEPG"
    },
    ...
  ],
  ...
},
...
],
"filterAssociations": [
{

```

```

    "filterRef": "/schemas/583c7c482501002501061985/templates/templatel/filters/filter1",
    "contractIdentities": [
      {
        "contractRef":
"/schemas/583c7c482501002501061986/templates/templatel/contracts/contract1",
        "schemaName": "Schema 1",
        "templateName": "Template 1"
        "displayName": "Contract 1"
      }, ....
    ]
  }, ...
],
"contractAssociations" : [
  {
    "contractRef":
"/schemas/583c7c482501002501061985/templates/templatel/contracts/contract1",
    "epgIdentities": [
      {
        "epgRef": "/schemas/583c7c482501002501061986/templates/templatel/epgs/epg1",
        "schemaName": "Schema 1",
        "templateName": "Template 1"
        "displayName": "EPG1"
      }, ....
    ],
    "externalEpgIdentities": [
      {
        "externalEpgRef":
"/schemas/583c7c482501002501061986/templates/templatel/externalEpgs/Test-extEPG",
        "schemaName": "Demo-extEPG",
        "templateName": "Provider",
        "displayName": "Test-extEPG"
      }, ....
    ]
  }, ...
],
}, ....
]
}

```

Example for ACI Multi-Site Release 1.0(1)

The following is an example for reading a Schema.

GET

GET https://<host>:<port>/api/v1/schemas/583c7c482501002501061985

REQUEST

GET

Request Body:
<empty>

RESPONSE

```

HTTP/1.1 200
{
  "id": "583c7c482501002501061985",
  "displayName": "Schema 1",
  ...
}

```



```
<< additional attributes >>
}
```

Schema Health Summary

The following is an example of a Schema health summary.



Note This API call returns a summary of the health of all Schemas to which the currently logged in user has access. The summary includes aggregated health of all policies in a Schema that has been pushed to a site, on a per template/site basis, as well as template/tenant identity information. This API is used for the Policy Heatmap on the dashboard page in the GUI .

Only Schemas that have been deployed will have health information.

GET

```
GET /api/v1/schemas/health-summary
```

REQUEST

```
<empty>
```

RESPONSE

```
{
  "schemas": [
    {
      "id": "5969609c400000570068d69d",
      "schemaId": "5965bb96340000d800908add",
      "displayName": "Schema-Provider-AD",
      "templates": [
        {
          "name": "Templatel",
          "displayName": "AD-Template-500"
        }
      ],
      "healthFaults": [
        {
          "siteId": "595e9796c00000c00000d632",
          "templateName": "Templatel"
        }
      ]
    },
    {
      "id": "596e94253e00005f01347df0",
      "schemaId": "596e93c0860000e802274ac6",
      "displayName": "Common Schema",
      "templates": [
        {
          "name": "Simulator1CommonTenant",
          "displayName": "Simulator 1 Common Tenant"
        }
      ],
    }
  ]
}
```

```

        "name": "ifav87-site3",
        "displayName": "ifav87-site3"
    }
  ],
  "healthFaults": [
    {
      "siteId": "596d00644400008d00ca4ed0",
      "templateName": "Simulator1CommonTenant",
      "health": {
        "score": 100
      }
    },
    {
      "siteId": "596ff6e61f00007a002d1880",
      "templateName": "ifav87-site3",
      "health": {
        "score": 100
      }
    }
  ]
},
{
  "id": "5971314e2100007800076f36",
  "schemaId": "59713115850000230144a54f",
  "displayName": "T2 Schema",
  "templates": [
    {
      "name": "Template1",
      "displayName": "Template 1"
    }
  ],
  "healthFaults": [
    {
      "siteId": "596d00644400008d00ca4ed0",
      "templateName": "Template1"
    }
  ]
}
]
}

```

Schema Health Sync

The following is an example of how to sync health and fault information for all tenants in a schema from all the associated sites

GET

```
GET /api/v1/health/schemas/:schemaId/sync
```

REQUEST

<empty>

RESPONSE

```
{
```

```

"id": "5977ba963f0000de04760560",
"schemaId": "5977ba913500006403c31911",
"healthFaults": [
  {
    "siteId": "59726e7a490000ea095b4cc8",
    "templateName": "PespiTenant",
    "health": {
      "score": 100
    },
    "anps": [
      {
        "anpRef": {
          "schemaId": "5977ba913500006403c31911",
          "templateName": "PespiTenant",
          "anpName": "Pepsi-App"
        },
        "health": {
          "score": 100
        },
        "faults": [],
        "epgs": [
          {
            "epgRef": {
              "schemaId": "5977ba913500006403c31911",
              "templateName": "PespiTenant",
              "anpName": "Pepsi-App",
              "epgName": "App-EPG"
            },
            "health": {
              "score": 100
            },
            "faults": []
          }
        ]
      }
    ]
  },
  {
    "vrfRef": {
      "schemaId": "5977ba913500006403c31911",
      "templateName": "PespiTenant",
      "vrfName": "TestVrf"
    },
    "health": {
      "score": 100
    },
    "faults": []
  }
],
"bds": [
  {
    "bdRef": {
      "schemaId": "5977ba913500006403c31911",
      "templateName": "PespiTenant",
      "bdName": "TestBD"
    },
    "health": {
      "score": 100
    },
    "faults": []
  }
],
"contracts": [
  {

```

```

        "contractRef": {
          "schemaId": "5977ba913500006403c31911",
          "templateName": "PespiTenant",
          "contractName": "TestContract"
        },
        "faults": []
      },
      {
        "contractRef": {
          "schemaId": "5977ba913500006403c31911",
          "templateName": "PespiTenant",
          "contractName": "TestContract2"
        },
        "faults": [
          {
            "type": "minor",
            "message": "Configuration failed for Tenant Pepsi Contract
TestContract2 due to Filter Not Present",
            "code": "F1228"
          }
        ]
      }
    ]
  }
}

```

Delete Schema Health Information

The following is an example of how to delete health and fault information for a schema.

GET

GET /api/v1/schemas/health-summary

REQUEST

<empty>

RESPONSE

<empty>

Retrieve Policy States

Example for ACI Multi-Site Release 1.0(2)

The following is an example that shows how to list policy states.

GET

GET https://<host>:<port>/api/v1/:id/policy-states

REQUEST

```
<empty>
```

RESPONSE

```
HTTP/1.1 200
{
  "policyStates": [
    {
      "siteId": "59631b3942000042001f1790",
      "templateName": "Template1",
      "anps": [...],
      "vrfs": [
        {
          "vrfRef": "/schemas/5963c0ff240000e900b34c55/templates/Template1/vrfs/VRf1",
          "apicState": "deleted"
        },
        {
          "vrfRef": "/schemas/5963c0ff240000e900b34c55/templates/Template1/vrfs/VRf2",
          "state": "created"
        }
      ],
      "bds": [
        {
          "bdRef": "/schemas/5963c0ff240000e900b34c55/templates/Template1/bds/BD10",
          "state": "modified"
        },
        {
          "bdRef": "/schemas/5963c0ff240000e900b34c55/templates/Template1/bds/BD11",
          "state": "deleted"
        },
        {
          "bdRef": "/schemas/5963c0ff240000e900b34c55/templates/Template1/bds/BD12",
          "state": "created"
        }
      ],
      "contracts": [...],
      "filters": [...],
      "externalEpgs": [
        {
          "externalEpgRef":
"/schemas/5963c0ff240000e900b34c55/templates/template1/externalEpgs/Test-extEPG",
          "state": "created"
        }
      ]
    }
  ]
}
```

Retrieve References

Example for ACI Multi-Site Release 1.0(2)

The following is an example that shows how to retrieve references.

GET

```
GET https://<host>:<port>/api/v1/:id/references
```

REQUEST

```
<empty>
```

RESPONSE

```
HTTP/1.1 200
{
  "references": {
    "vrfs": [
      {
        "identity": {
          "vrfRef": "/schemas/5963c0ff240000e900b34c55/templates/Template1/vrfs/VRF11",
          "schemaName": "Schema 11",
          "templateName": "Template 1",
          "displayName": "VRF 11"
        },
        "deployedOn": [
          "59631b3942000042001f1790"
        ]
      }
    ]
  }
}
```

Retrieving Schema Health and Faults

The following is an example for retrieving Schema health faults included.



Note

Health and Fault information is sideloaded with Schema configuration in a separate internal structure called "HealthFaults". This structure closely resembles the template structure in terms of hierarchy of policies, and it includes health information for policies that can have health (ANP, EPG, VRF, BD) and faults for all policies that can have faults (ANP, EPG, VRF, BD, Contract). The HealthFaults structures exist on a per template/site basis. Health and fault data will only exist for policies that have been pushed.

GET

```
GET /api/v1/schemas/583c7c482501002501061985?include=health,faults
```

REQUEST

```
<empty>
```

RESPONSE

```
HTTP/1.1 200
{
  "id": "5971480c1b000067319d7b3b",
  "displayName": "Schema 1",
}
```

```

"templates": [
  {
    "name": "Templatel",
    "displayName": "Template 1",
    "tenantId": "5952ebc21e00007b16c1a022",
    "anps": [
      {
        "name": "ANP1",
        "displayName": "ANP1",
        "anpRef":
"/schemas/5971480c1b000067319d7b3b/templates/Templatel/anps/ANP1",
        "epgs": [
          {
            "name": "EPG1",
            "displayName": "EPG1",
            "epgRef":
"/schemas/5971480c1b000067319d7b3b/templates/Templatel/anps/ANP1/epgs/EPG1",
            "contractRelationships": [],
            "subnets": [],
            "uSegEpg": false,
            "uSegAttrs": [],
            "intraEpg": "unenforced",
            "proxyArp": false,
            "bdRef":
"/schemas/5971480c1b000067319d7b3b/templates/Templatel/bds/BD1"
          }
        ]
      }
    ],
    "vrfs": [
      {
        "name": "VRF1",
        "displayName": "VRF1",
        "vrfRef":
"/schemas/5971480c1b000067319d7b3b/templates/Templatel/vrfs/VRF1"
      }
    ],
    "bds": [
      {
        "name": "BD1",
        "displayName": "BD1",
        "bdRef": "/schemas/5971480c1b000067319d7b3b/templates/Templatel/bds/BD1",
        "l2UnknownUnicast": "proxy",
        "intersiteBumTrafficAllow": true,
        "optimizeWanBandwidth": false,
        "l2Stretch": true,
        "subnets": [],
        "vrfRef":
"/schemas/5971480c1b000067319d7b3b/templates/Templatel/vrfs/VRF1"
      }
    ],
    "contracts": [
      {
        "name": "Contract1",
        "displayName": "Contract1",
        "contractRef":
"/schemas/5971480c1b000067319d7b3b/templates/Templatel/contracts/Contract1",
        "filterRelationships": [],
        "scope": "context"
      },
      {
        "name": "Contract2",
        "displayName": "Contract2",

```

```

        "contractRef":
"/schemas/5971480c1b000067319d7b3b/templates/Template1/contracts/Contract2",
        "filterRelationships": [
            {
                "filterRef":
"/schemas/5971480c1b000067319d7b3b/templates/Template1/filters/Filter1",
                "directives": [
                    "none"
                ]
            },
            {
                "filterRef":
"/schemas/5971480c1b000067319d7b3b/templates/Template1/filters/Filter2",
                "directives": [
                    "none"
                ]
            }
        ],
        "scope": "context"
    }
],
"filters": [
    {
        "name": "Filter1",
        "displayName": "Filter1",
        "filterRef":
"/schemas/5971480c1b000067319d7b3b/templates/Template1/filters/Filter1",
        "entries": []
    },
    {
        "name": "Filter2",
        "displayName": "Filter2",
        "filterRef":
"/schemas/5971480c1b000067319d7b3b/templates/Template1/filters/Filter2",
        "entries": [
            {
                "name": "Entry1",
                "displayName": "Entry1",
                "description": "",
                "etherType": "ip",
                "ipProtocol": "tcp",
                "matchOnlyFragments": false,
                "stateful": false,
                "sourceFrom": "unspecified",
                "sourceTo": "unspecified",
                "destinationFrom": "unspecified",
                "destinationTo": "unspecified",
                "tcpSessionRules": [
                    "unspecified"
                ]
            }
        ]
    }
]
}
],
"sites": [
    {
        "siteId": "594b77e84400004000752a18",
        "templateName": "Template1",
        "anps": [],
        "vrfs": [],
        "bds": [],
        "contracts": []
    }
]

```



```

    },
    {
      "siteId": "594b77dc4400004400752a17",
      "templateName": "Templatel",
      "anps": [],
      "vrfs": [],
      "bds": [],
      "contracts": []
    }
  ],
  "healthFaults": [
    {
      "siteId": "594b77e84400004000752a18",
      "templateName": "Templatel",
      "health": {
        "score": 100
      },
      "anps": [
        {
          "anpRef":
"/schemas/5971480c1b000067319d7b3b/templates/Templatel/anps/ANP1",
          "health": {
            "score": 100
          },
          "faults": [
            {
              "type": "minor",
              "message": "Configuration failed for EPG EPG1 due to BD IDs Not
Allocated,Context IDs Not Allocated",
              "code": "F0523"
            }
          ],
          "epgs": [
            {
              "epgRef":
"/schemas/5971480c1b000067319d7b3b/templates/Templatel/anps/ANP1/epgs/EPG1",
              "health": {
                "score": 100
              },
              "faults": [
                {
                  "type": "minor",
                  "message": "Configuration failed for EPG EPG1 due to
BD IDs Not Allocated,Context IDs Not Allocated",
                  "code": "F0523"
                }
              ]
            }
          ]
        }
      ],
      "vrfs": [
        {
          "vrfRef":
"/schemas/5971480c1b000067319d7b3b/templates/Templatel/vrfs/VRF1",
          "health": {
            "score": 100
          },
          "faults": []
        }
      ],
      "bds": [
        {
          "bdRef": "/schemas/5971480c1b000067319d7b3b/templates/Templatel/bds/BD1",

```

```

        "health": {
            "score": 100
        },
        "faults": []
    }
],
"contracts": [
    {
        "contractRef":
"/schemas/5971480c1b000067319d7b3b/templates/Template1/contracts/Contract1",
        "faults": [
            {
                "type": "minor",
                "message": "Configuration failed for Tenant DISA Contract
Contract1 due to Filter Not Present",
                "code": "F1228"
            }
        ]
    },
    {
        "contractRef":
"/schemas/5971480c1b000067319d7b3b/templates/Template1/contracts/Contract2",
        "faults": []
    }
]
},
{
    "siteId": "594b77dc4400004400752a17",
    "templateName": "Template1",
    "health": {
        "score": 100
    },
    "anps": [
        {
            "anpRef":
"/schemas/5971480c1b000067319d7b3b/templates/Template1/anps/ANP1",
            "health": {
                "score": 100
            },
            "faults": [
                {
                    "type": "minor",
                    "message": "Configuration failed for EPG EPG1 due to BD IDs Not
Allocated",
                    "code": "F0523"
                }
            ],
            "epgs": [
                {
                    "epgRef":
"/schemas/5971480c1b000067319d7b3b/templates/Template1/anps/ANP1/epgs/EPG1",
                    "health": {
                        "score": 100
                    },
                    "faults": [
                        {
                            "type": "minor",
                            "message": "Configuration failed for EPG EPG1 due to
BD IDs Not Allocated",
                            "code": "F0523"
                        }
                    ]
                }
            ]
        }
    ]
}

```

```

    ]
  },
  "vrfs": [
    {
      "vrfRef":
"/schemas/5971480c1b000067319d7b3b/templates/Template1/vrfs/VRF1",
      "health": {
        "score": 100
      },
      "faults": []
    }
  ],
  "bds": [
    {
      "bdRef": "/schemas/5971480c1b000067319d7b3b/templates/Template1/bds/BD1",
      "health": {
        "score": 100
      },
      "faults": []
    }
  ],
  "contracts": [
    {
      "contractRef":
"/schemas/5971480c1b000067319d7b3b/templates/Template1/contracts/Contract1",
      "faults": [
        {
          "type": "minor",
          "message": "Configuration failed for Tenant DISA Contract
Contract1 due to Filter Not Present",
          "code": "F1228"
        }
      ]
    },
    {
      "contractRef":
"/schemas/5971480c1b000067319d7b3b/templates/Template1/contracts/Contract2",
      "faults": []
    }
  ]
}
]
}

```

Update an Existing Schema

Example for ACI Multi-Site Release 1.0(2)

The following is an example to update an existing Schema.

PUT

PUT https://<host>:<port>/api/v1/schemas/583c7c482501002501061985

REQUEST

```

{
  "id": "583c7c482501002501061985",
  "displayName": "Schema 1 (Updated)",
  "templates": [...],

```

```

"sites": [
  {
    "siteId": "58202f7066e6e10001c41236",
    "templateName": "template1",
    "vrfs": [
      {
        "vrfRef": "/schemas/583c7c482501002501061985/templates/template1/vrfs/vrf1",
        "bds": [
          {
            "bdRef":
"/schemas/583c7c482501002501061985/templates/template1/vrfs/vrf1/bds/bd1",
            "subnet": "<subnet>"
          }
        ]
      }
    ]
  }
]
}

```

RESPONSE

HTTP/1.1 200

```

{
  "schemas": [
    {
      "id": "594b78792d00008000d5fb6f",
      "displayName": "Saad_Stretchd",
      "templates": [
        {
          "name": "Template1",
          "displayName": "Template 3",
          "tenantId": "594b77c82d00006000d5fb6c",
          "anps": [],
          "vrfs": [],
          "bds": [],
          "contracts": [],
          "filters": [],
          "externalEpgs": []
        },
        {
          "name": "Template2",
          "displayName": "Template 2safsd",
          "tenantId": "594b77c82d00006000d5fb6c",
          "anps": [],
          "vrfs": [],
          "bds": [],
          "contracts": [],
          "filters": [],
          "externalEpgs": []
        }
      ],
      "sites": [
        {
          "siteId": "594b77dc4400004400752a17",
          "templateName": "Template1",
          "anps": [],
          "vrfs": [],
          "bds": [],
          "contracts": [],
          "externalEpgs": []
        },
        {

```

```

    "siteId": "594b77e84400004000752a18",
    "templateName": "Template1",
    "anps": [],
    "vrfs": [],
    "bds": [],
    "contracts": [],
    "externalEpgs": []
  },
  {
    "siteId": "594b77e84400004000752a18",
    "templateName": "Template2",
    "anps": [],
    "vrfs": [],
    "bds": [],
    "contracts": [],
    "externalEpgs": []
  }
]
},
{
  "id": "594b790a2d0000a900d5fb70",
  "displayName": "SharedService",
  "templates": [
    {
      "name": "Template1",
      "displayName": "Template 1",
      "tenantId": "594b782d2d00006f00d5fb6d",
      "anps": [],
      "vrfs": [],
      "bds": [],
      "contracts": [],
      "filters": [],
      "externalEpgs": []
    }
  ],
  "sites": [
    {
      "siteId": "594b77dc4400004400752a17",
      "templateName": "Template1",
      "anps": [],
      "vrfs": [],
      "bds": [],
      "contracts": [],
      "externalEpgs": []
    }
  ]
}
}

```

**Note**

- "id" is optional, other attributes are mandatory even if unchanged.
- Many attributes in this example have been omitted for brevity, but they must be included in the PUT request. Refer to the full list of attributes in the POST example.
- The "ref" values for existing sub-entities must be specified.

Example for ACI Multi-Site Release 1.0(1)

The following is an example to updated an existing Schema.

PUT

```
PUT https://<host>:<port>/api/v1/schemas/583c7c482501002501061985
```

REQUEST

```
{
  "id": "583c7c482501002501061985",
  "displayName": "Schema 1 (Updated)",
  "templates": [...],
  "sites": [
    {
      "siteId": "58202f7066e6e10001c41236",
      "templateName": "template1",
      "vrfs": [
        {
          "vrfRef": "/schemas/583c7c482501002501061985/templates/template1/vrfs/vrf1",
          "bds": [
            {
              "bdRef":
"/schemas/583c7c482501002501061985/templates/template1/vrfs/vrf1/bds/bd1",
              "subnet": "<subnet>"
            }
          ]
        }
      ]
    }
  ]
}
```

RESPONSE

```
HTTP/1.1 200
{
  "id": "583c7c482501002501061985",
  "displayName": "Schema 1 (Updated)",
  "templates": [...],
  "sites": [...]
```

**Note**

- "id" is optional, other attributes are mandatory even if unchanged.
- Many attributes in this example have been omitted for brevity, but they must be included in the PUT request. Refer to the full list of attributes in the POST example.
- The "ref" values for existing sub-entities must be specified.

Delete a Schema

The following is an example for deleting a Schema.

DELETE

```
DELETE https://<host>:<port>/api/v1/schemas/583c7c482501002501061985
```

REQUEST

```
DELETE
Request Body:
  <empty>
...
```

RESPONSE

```
HTTP/1.1 204
```

List Schema Identities

The following is an example for listing Schema identities.

GET

```
GET https://<host>:<port>/api/v1/schemas/list-identity
```



Note This provides a list of Schemas in the system with a few properties. Only template-tenant associations and template-site associations are returned. Policy configuration information is not included.

REQUEST

```
<empty>
```

RESPONSE

```
HTTP/1.1 200
{
  "schemas": [
    {
      "id": "594b78792d00008000d5fb6f",
      "displayName": "Saad_Stretched",
      "templates": [
        {
          "name": "Template1",
          "displayName": "Template 3",
          "tenantId": "594b77c82d00006000d5fb6c",
          "anps": [],
          "vrfs": [],
          "bds": [],
          "contracts": [],
          "filters": []
        },
        {
          "name": "Template2",
          "displayName": "Template 2saf sdf",
          "tenantId": "594b77c82d00006000d5fb6c",
          "anps": [],
          "vrfs": [],

```

```

        "bds": [],
        "contracts": [],
        "filters": []
      }
    ],
    "sites": [
      {
        "siteId": "594b77dc4400004400752a17",
        "templateName": "Template1",
        "anps": [],
        "vrfs": [],
        "bds": [],
        "contracts": []
      },
      {
        "siteId": "594b77e84400004000752a18",
        "templateName": "Template1",
        "anps": [],
        "vrfs": [],
        "bds": [],
        "contracts": []
      },
      {
        "siteId": "594b77e84400004000752a18",
        "templateName": "Template2",
        "anps": [],
        "vrfs": [],
        "bds": [],
        "contracts": []
      }
    ]
  },
  {
    "id": "594b790a2d0000a900d5fb70",
    "displayName": "SharedService",
    "templates": [
      {
        "name": "Template1",
        "displayName": "Template 1",
        "tenantId": "594b782d2d00006f00d5fb6d",
        "anps": [],
        "vrfs": [],
        "bds": [],
        "contracts": [],
        "filters": []
      }
    ]
  },
  "sites": [
    {
      "siteId": "594b77dc4400004400752a17",
      "templateName": "Template1",
      "anps": [],
      "vrfs": [],
      "bds": [],
      "contracts": []
    }
  ]
}
]
}

```


List Sites Associated to a Schema

The following is an example for listing sites associated to a Schema.

GET

```
GET https://<host>:<port>/api/v1/schemas/allowed-sites
```

REQUEST

GET

Request Body:
<empty>

RESPONSE

```
HTTP/1.1 200
{
  "sites": [
    {
      "id": "591b6e5172040072044258ea",
      "name": "site1",
      "urls": [
        "https://10.23.236.232"
      ],
      "username": "admin",
      "password": "*****",
      "siteId": "1"
    },
    {
      "id": "591b71353f00002700a2e8b1",
      "name": "site2",
      "urls": [
        "https://10.23.236.233"
      ],
      "username": "admin",
      "password": "*****",
      "siteId": "2"
    }
  ]
}
```

List Tenants that Can Be Associated With a Schema

The following is an example of how to list Tenants that can be associated with a Schema.

POST:

```
POST https://<host>:<port>/api/v1/schemas
```

Request:

<empty>

Response:

```
HTTP/1.1 200
{
```

```

"tenants": [
  {
    "id": "591b6dal200000dc03b47a10",
    "name": "TenantA",
    "displayName": "Tenant A",
    "description": "Tenant A",
    "siteAssociations": [
      {
        "siteId": "591b6e5172040072044258ea",
        "securityDomains": []
      }
    ],
    "userAssociations": [
      {
        "userId": "0000ffff00000000000000020"
      },
      {
        "userId": "591b562f210000d000823836"
      }
    ]
  },
  {
    "id": "591b6e35200000f303b47a11",
    "name": "TenantB",
    "displayName": "Tenant B",
    "description": "Tenant B",
    "siteAssociations": [
      {
        "siteId": "591b6e5172040072044258ea",
        "securityDomains": []
      },
      {
        "siteId": "591b71353f00002700a2e8b1",
        "securityDomains": []
      }
    ],
    "userAssociations": [
      {
        "userId": "0000ffff00000000000000020"
      }
    ]
  }
]
}

```

List All Tenants Associated to Schemas

The following is an example of how to list all Tenants associated to Schemas:

Endpoint:

`https://<host>:<port>/api/v1/schemas/sites`

Request:

`<empty>`

Response:

```

HTTP/1.1 200
{
  "tenants": [

```

```

{
  "id": "0000ffff0000000000000010",
  "name": "common",
  "displayName": "common",
  "siteAssociations": [],
  "userAssociations": [
    {
      "userId": "0000ffff0000000000000020"
    }
  ],
  "description": "Common tenant for use with all other tenants"
},
{
  "id": "593b102322000088009ff785",
  "name": "TenantA",
  "displayName": "Tenant A",
  "siteAssociations": [
    {
      "siteId": "593b101169000069002de4c6",
      "securityDomains": [
        "TestSecurityDomain"
      ]
    }
  ],
  "userAssociations": [
    {
      "userId": "593b12e62000004b00f13b21"
    }
  ],
  "description": ""
},
{
  "id": "593b12cc2300002600eb63bd",
  "name": "TenantB",
  "displayName": "Tenant B",
  "siteAssociations": [
    {
      "siteId": "593b101169000069002de4c6",
      "securityDomains": []
    }
  ],
  "userAssociations": [
    {
      "userId": "593b12f62000005200f13b22"
    },
    {
      "userId": "593eba36210000140183e1a1"
    }
  ],
  "description": ""
}
]
}

```

List Sites That Can Be Associated With a Schema

The following is an example of how to list Sites that can be associated with a Schema.

POST:

POST <https://<host>:<port>/api/v1/schemas/allowed-sites>

Request:

<empty>

Response:

```
HTTP/1.1 200
{
  "sites": [
    {
      "id": "591b6e5172040072044258ea",
      "name": "site1",
      "urls": [
        "https://10.23.236.232"
      ],
      "username": "admin",
      "password": "*****",
      "siteId": "1"
    },
    {
      "id": "591b71353f00002700a2e8b1",
      "name": "site2",
      "urls": [
        "https://10.23.236.233"
      ],
      "username": "admin",
      "password": "*****",
      "siteId": "2"
    }
  ]
}
```

List All Sites Associated to Schemas

The following is an example of all Sites associated to Schemas.

POST:

POST https://<host>:<port>/api/v1/schemas/sites

Request:

<empty>

Response:

```
HTTP/1.1 200
{
  "sites": [
    {
      "id": "591b6e5172040072044258ea",
      "name": "site1",
      "urls": [
        "https://10.23.236.232"
      ],
      "username": "admin",
      "password": "*****",
      "siteId": "1"
    }
  ]
}
```

```

    },
    {
      "id": "591b71353f00002700a2e8b1",
      "name": "site2",
      "urls": [
        "https://10.23.236.233"
      ],
      "username": "admin",
      "password": "*****",
      "siteId": "2"
    }
  ]
}

```

List Tenants That Can be Associated With a Schema

The following is an example of how to list Tenants that can be associated with a Schema.

POST:

```
POST https://<host>:<port>/api/v1/schemas/allowed-tenants
```

Request:

```
<empty>
```

Response:

```

HTTP/1.1 200
{
  "tenants": [
    {
      "id": "591b6da1200000dc03b47a10",
      "name": "TenantA",
      "displayName": "Tenant A",
      "description": "Tenant A",
      "siteAssociations": [
        {
          "siteId": "591b6e5172040072044258ea",
          "securityDomains": []
        }
      ],
      "userAssociations": [
        {
          "userId": "0000ffff000000000000000020"
        },
        {
          "userId": "591b562f210000d000823836"
        }
      ]
    },
    {
      "id": "591b6e35200000f303b47a11",
      "name": "TenantB",
      "displayName": "Tenant B",
      "description": "Tenant B",
      "siteAssociations": [
        {
          "siteId": "591b6e5172040072044258ea",

```

```

        "securityDomains": []
      },
      {
        "siteId": "591b71353f00002700a2e8b1",
        "securityDomains": []
      }
    ],
    "userAssociations": [
      {
        "userId": "0000ffff00000000000000020"
      }
    ]
  }
]
}

```

List All Sites Associated to Schemas

The following is an example of all Sites associated to Schemas.

POST:

POST https://<host>:<port>/api/v1/schemas/sites

Request:

<empty>

Response:

```

HTTP/1.1 200
{
  "sites": [
    {
      "id": "591b6e5172040072044258ea",
      "name": "site1",
      "urls": [
        "https://10.23.236.232"
      ],
      "username": "admin",
      "password": "*****",
      "siteId": "1"
    },
    {
      "id": "591b71353f00002700a2e8b1",
      "name": "site2",
      "urls": [
        "https://10.23.236.233"
      ],
      "username": "admin",
      "password": "*****",
      "siteId": "2"
    }
  ]
}

```

APIs for Import

Import a Tenant from APIC

The following is an example showing how to import a tenant from APIC.

GET

```
GET https://<host>:<port>/api/v1/import/sites/:id/tenants/:tenantId/templates/:templateName
```

An example:

```
/api/v1/import/sites/58b5f5ca390000390078e257/tenants/58b5f5ca390000390078e257/templates/template1
```

REQUEST

<empty>

RESPONSE

```
{
  "displayName": "",
  "templates": [
    {
      "name": "Template1",
      "displayName": "Template1",
      "tenantId": "5923588f800000e302e2a4fd",
      "anps": [
        {
          "name": "ANP1",
          "displayName": "ANP1",
          "anpRef": "/templates/Template1/anps/ANP1",
          "epgs": [
            {
              "name": "EPG1",
              "displayName": "EPG1",
              "epgRef": "/templates/Template1/anps/ANP1/epgs/EPG1",
              "contractRelationships": [
                {
                  "relationshipType": "provider",
                  "contractRef": "/templates/Template1/contracts/Contract1"
                }
              ]
            }
          ],
          "subnets": [
            {
              "ip": "10.0.0.1/24",
              "scope": "private",
              "shared": false
            },
            {
              "ip": "5.5.5.5/21",
              "scope": "private",
              "shared": false
            }
          ]
        },
        "uSegEpg": false,
        "uSegAttrs": [],
        "intraEpg": "unenforced",
        "bdRef": "/templates/Template1/bds/BD1"
      }
    ]
  }
}
```

```

    },
    {
      "name": "UntitledEPG1",
      "displayName": "UntitledEPG1",
      "epgRef": "/templates/Template1/anps/ANP1/epgs/UntitledEPG1",
      "contractRelationships": [],
      "subnets": [],
      "uSegEpg": false,
      "uSegAttrs": [],
      "intraEpg": "unenforced",
      "bdRef": "/templates/Template1/bds/BD1"
    }
  ]
},
{
  "name": "UntitledANP1",
  "displayName": "UntitledANP1",
  "anpRef": "/templates/Template1/anps/UntitledANP1",
  "epgs": [
    {
      "name": "EPG-Test",
      "displayName": "EPG-Test",
      "epgRef": "/templates/Template1/anps/UntitledANP1/epgs/EPG-Test",
      "contractRelationships": [],
      "subnets": [],
      "uSegEpg": false,
      "uSegAttrs": [],
      "intraEpg": "unenforced",
      "bdRef": "/templates/Template1/bds/BD1"
    }
  ]
},
{
  "name": "anp323",
  "displayName": "anp323",
  "anpRef": "/templates/Template1/anps/anp323",
  "epgs": [
    {
      "name": "sadas",
      "displayName": "sadas",
      "epgRef": "/templates/Template1/anps/anp323/epgs/sadas",
      "contractRelationships": [],
      "subnets": [],
      "uSegEpg": false,
      "uSegAttrs": [],
      "intraEpg": "unenforced",
      "bdRef": "/templates/Template1/bds/BD1"
    }
  ]
}
],
"vrf": [
  {
    "name": "VRF1",
    "displayName": "VRF1",
    "vrfRef": "/templates/Template1/vrfs/VRF1"
  },
  {
    "name": "VRF2",
    "displayName": "VRF2",
    "vrfRef": "/templates/Template1/vrfs/VRF2"
  }
],
"bds": [

```



```

    {
      "name": "BD1",
      "displayName": "BD1",
      "bdRef": "/templates/Templat1/bds/BD1",
      "l2UnknownUnicast": "proxy",
      "intersiteBumTrafficAllow": false,
      "l2Stretch": true,
      "subnets": [],
      "vrfRef": "/templates/Templat1/vrfs/VRF1"
    }
  ],
  "contracts": [
    {
      "name": "Contract1",
      "displayName": "Contract1",
      "contractRef": "/templates/Templat1/contracts/Contract1",
      "filterRelationships": [
        {
          "filterRef": "/templates/Templat1/filters/Filter1",
          "directives": []
        }
      ],
      "scope": "global",
      "subjectName": "subj1"
    },
    {
      "name": "Contract2",
      "displayName": "Contract2",
      "contractRef": "/templates/Templat1/contracts/Contract2",
      "filterRelationships": [
        {
          "filterRef": "/templates/Templat1/filters/Filter2",
          "directives": []
        },
        {
          "filterRef": "/templates/Templat1/filters/Filter3",
          "directives": []
        }
      ],
      "scope": "global",
      "subjectName": "msc-subject"
    },
    {
      "name": "Contract3",
      "displayName": "Contract3",
      "contractRef": "/templates/Templat1/contracts/Contract3",
      "filterRelationships": [],
      "scope": "context",
      "subjectName": "msc-subject"
    }
  ],
  "filters": [
    {
      "name": "Filter1",
      "displayName": "Filter1",
      "filterRef": "/templates/Templat1/filters/Filter1",
      "entries": []
    },
    {
      "name": "Filter2",
      "displayName": "Filter2",
      "filterRef": "/templates/Templat1/filters/Filter2",
      "entries": [
        {

```

```

        "name": "Entry2",
        "displayName": "Entry2",
        "etherType": "ip",
        "arpFlag": "unspecified",
        "ipProtocol": "unspecified",
        "matchOnlyFragments": false,
        "stateful": false,
        "sourceFrom": "unspecified",
        "sourceTo": "unspecified",
        "destinationFrom": "unspecified",
        "destinationTo": "unspecified",
        "tcpSessionRules": [
            "unspecified"
        ]
    }
}
],
{
    "name": "Filter3",
    "displayName": "Filter3",
    "filterRef": "/templates/Template1/filters/Filter3",
    "entries": []
}
]
},
"sites": [
    {
        "siteId": "591b427d82000a70055bbd5",
        "templateName": "Template1",
        "anps": [
            {
                "anpRef": "/templates/Template1/anps/ANP1",
                "eggs": [
                    {
                        "eggRef": "/templates/Template1/anps/ANP1/eggs/EPG1",
                        "domainAssociations": [
                            {
                                "dn": "uni/phys-testPhysicalDomain-DoNotDelete",
                                "domainType": "physicalDomain",
                                "deployImmediacy": "lazy",
                                "resolutionImmediacy": "lazy"
                            },
                            {
                                "dn": "uni/l3dom-TestL3Domain-DoNotDelete",
                                "domainType": "l3ExtDomain",
                                "deployImmediacy": "lazy",
                                "resolutionImmediacy": "lazy"
                            },
                            {
                                "dn": "uni/fc-TestFibreChannelDomain",
                                "domainType": "fibreChannelDomain",
                                "deployImmediacy": "lazy",
                                "resolutionImmediacy": "lazy"
                            },
                            {
                                "dn": "uni/l2dom-TestL2Domain-DoNotDelete",
                                "domainType": "l2ExtDomain",
                                "deployImmediacy": "lazy",
                                "resolutionImmediacy": "lazy"
                            }
                        ]
                    },
                    "staticPorts": [
                        {

```

```

        "type": "port",
        "path": "topology/pod-1/paths-101/pathep-[eth1/1]",
        "portEncapVlan": 1234,
        "deploymentImmediacy": "lazy",
        "mode": "regular"
    }
],
"staticLeafs": [
    {
        "path": "topology/pod-1/node-101",
        "portEncapVlan": 1234
    }
],
"faults": []
},
{
    "epgRef": "/templates/Templat1/anps/ANP1/epgs/UntitledEPG1",
    "domainAssociations": [
        {
            "dn": "uni/vmmp-VMware/dom-Fab1-DVS1",
            "domainType": "vmmDomain",
            "deployImmediacy": "lazy",
            "resolutionImmediacy": "immediate",
            "vlanEncapMode": "dynamic",
            "allowMicroSegmentation": true
        },
        {
            "dn": "uni/vmmp-VMware/dom-TestDomain",
            "domainType": "vmmDomain",
            "deployImmediacy": "lazy",
            "resolutionImmediacy": "immediate",
            "portEncapVlan": {
                "vlanType": "vlan",
                "vlan": 3
            },
            "vlanEncapMode": "static",
            "allowMicroSegmentation": false
        },
        {
            "dn": "uni/vmmp-VMware/dom-TestDomain-DoNotDelete",
            "domainType": "vmmDomain",
            "deployImmediacy": "lazy",
            "resolutionImmediacy": "immediate",
            "microSegVlan": {
                "vlanType": "vlan",
                "vlan": 2
            },
            "portEncapVlan": {
                "vlanType": "vlan",
                "vlan": 1
            },
            "vlanEncapMode": "static",
            "allowMicroSegmentation": true
        }
    ],
"staticPorts": [
    {
        "type": "dpc",
        "path": "topology/pod-1/paths-101/pathep-[abc]",
        "portEncapVlan": 1234,
        "deploymentImmediacy": "lazy",
        "mode": "regular"
    }
],

```

```

        "staticLeafs": [],
        "faults": []
    }
  ],
  "faults": [],
},
{
  "anpRef": "/templates/Template1/anps/UntitledANP1",
  "eggs": [
    {
      "eggRef": "/templates/Template1/anps/UntitledANP1/eggs/EPG-Test",
      "domainAssociations": [
        {
          "dn": "uni/vmmp-VMware/dom-TestDomain-DoNotDelete",
          "domainType": "vmmDomain",
          "deployImmediacy": "lazy",
          "resolutionImmediacy": "immediate",
          "portEncapVlan": {
            "vlanType": "vlan",
            "vlan": 243
          },
          "vlanEncapMode": "static",
          "allowMicroSegmentation": false
        }
      ],
      "staticPorts": [],
      "staticLeafs": [],
      "faults": []
    }
  ],
  "faults": []
},
{
  "anpRef": "/templates/Template1/anps/anp323",
  "eggs": [
    {
      "eggRef": "/templates/Template1/anps/anp323/eggs/sadfas",
      "domainAssociations": [],
      "staticPorts": [
        {
          "type": "vpc",
          "path": "topology/pod-1/protopaths-101-102/pathep-[vpc1]",
          "portEncapVlan": 1234,
          "deploymentImmediacy": "lazy",
          "mode": "regular"
        }
      ],
      "staticLeafs": [],
      "faults": []
    }
  ],
  "faults": []
}
],
"vrfs": [],
"bds": [
  {
    "bdRef": "/templates/Template1/bds/BD1",
    "subnets": [],
    "faults": []
  }
],
"contracts": [
  {

```

```

        "contractRef": "/templates/Template1/contracts/Contract1",
        "serviceGraphTemplate": {
            "name": "TestSGT"
        },
        "faults": []
    },
    {
        "contractRef": "/templates/Template1/contracts/Contract2",
        "serviceGraphTemplate": {
            "name": "TestSGT"
        },
        "faults": []
    },
    {
        "contractRef": "/templates/Template1/contracts/Contract3",
        "faults": []
    }
]
}
],
"messages": {
    "contractMessages": [
        {
            "contractRef": "/templates/Template1/contracts/Contract1",
            "message": "Contract has multiple subjects. Only the first subject is imported."
        }
    ],
    "vrfMessages": [
        {
            "vrfRef": "/templates/Template1/vrfs/VRF1",
            "message": "VRF contains vzAny contracts which are not imported."
        },
        {
            "vrfRef": "/templates/Template1/vrfs/VRF2",
            "message": "VRF contains vzAny contracts which are not imported."
        }
    ]
}
}
}

```

APIs for Sites

Create a New Site

The following is an example of how to create a new Site.

POST:

POST https://<host>:<port>/api/v1/sites

Request:

```

{
  "name": "APIC Site San Francisco",
  "urls": "http://16.105.47.201:8982/sf",
  "username": "admin",
  "password": "w3lcome3",
  "apicSiteId" : "1"
}

```

Response:

```

HTTP/1.1 201
Location: /api/v1/sites/58202f7066e6e10001c41236
{
  "id": "58202f7066e6e10001c41236",
  "name": "APIC Site San Francisco",
  "urls": "http://16.105.47.201:8982/sf",
  "username": "admin",
  "password": "w3lcome3",
  "siteId" : "1"
}

```

Read Site Information

The following is an example of how to get basic information about a new Site:

Endpoint:

```
https://<host>:<port>/api/v1/sites/58202f7066e6e10001c41236
```

Request:

```

GET
Request Body:
<empty>

```

Response:

```

HTTP/1.1 200
{
  "id": "58202f7066e6e10001c41236",
  "name": "APIC Site San Francisco",
  "urls": [
    "http://16.105.47.201:8982/sf"
  ],
  "username": "admin",
  "password": "*****",
  "apicSiteId": "1"
}

```

Read Site Information That Contains Health Faults

The following is an example of how to get basic information about a new Site that also contains health faults:

Endpoint:

```
https://<host>:<port>/api/v1/sites/58202f7066e6e10001c41236?include=health,faults,cluster-status
```

Request:

```

GET
Request Body:
<empty>

```

Response:

```
HTTP/1.1 200
{
  "id": "58202f7066e6e10001c41236",
  "name": "APIC Site San Francisco",
  "urls": [
    "http://16.105.47.201:8982/sf"
  ],
  "username": "admin",
  "password": "*****",
  "apicSiteId": "1",
  "labels" : ["58202f7066e6e10001c41240", "58202f7066e6e10001c41241"],
  "health": {
    "score": 97
  },
  "clusterStatus": [
    {
      "controllerId": 1,
      "controllerName": "apic1",
      "ipAddress": "10.0.0.1",
      "adminState": "in-service",
      "operationalState": "available",
      "health": "fully-fit"
    },
    {
      "controllerId": 2,
      "controllerName": "apic2",
      "ipAddress": "10.0.0.2",
      "adminState": "in-service",
      "operationalState": "available",
      "health": "fully-fit"
    },
    {
      "controllerId": 3,
      "controllerName": "apic3",
      "ipAddress": "10.0.0.3",
      "adminState": "in-service",
      "operationalState": "available",
      "health": "fully-fit"
    }
  ],
  "faults": [
    {
      "count": 0,
      "type": "critical"
    },
    {
      "count": 2,
      "type": "major"
    },
    {
      "count": 736,
      "type": "minor"
    },
    {
      "count": 914,
      "type": "warning"
    }
  ]
}
```

Update an Existing Site

The following is an example for updating an existing Site.

Endpoint:

```
https://<host>:<port>/api/v1/sites/58202f7066e6e10001c41236
```

Request

```
PUT
{
  "id": "58202f7066e6e10001c41236",
  "name": "APIC Site San Francisco (Updated)",
  "urls": "http://16.105.47.201:8982/sf",
  "username": "root",
  "password": "w3lcome3",
  "apicSiteId" : "1"
}
```

Response

```
HTTP/1.1 200
{
  "id": "58202f7066e6e10001c41236",
  "name": "APIC Site San Francisco (Updated)",
  "urls": "http://16.105.47.201:8982/sf",
  "username": "root",
  "password": "w3lcome3",
  "apicSiteId" : "1"
}
```

Delete a Site

The following is an example of how to delete a Site:

Endpoint:

```
https://<host>:<port>/api/v1/sites/58202f7066e6e10001c41236
```

Request:

```
DELETE
Request Body:
<empty>
```

Response:

```
HTTP/1.1 204
Response Body:
<empty>
```


List Sites

The following is an example of how to list all Sites:

Endpoint:

```
https://<host>:<port>/api/v1/sites
```

Request:

```
GET
Request Body:
<empty>
```

Response:

```
HTTP/1.1 200
{
  "sites": [
    {
      "id": "58202f7066e6e10001c41236",
      "name": "APIC Site San Francisco",
      "urls": [
        "http://16.105.47.201:8982/sf"
      ],
      "username": "admin",
      "password": "*****",
      "apicSiteId": "1"
    },
    {
      "id": "58202fc466e6e10001c41237",
      "name": "APIC Site San Jose",
      "urls": [
        "http://16.105.47.201:8982/sjc"
      ],
      "username": "root",
      "password": "*****",
      "apicSiteId": "1"
    }
  ]
}
```



Note Filtering, pagination, and sorting is currently not supported.

List Labels

List Labels of Site Type Associated to a Site

The following is an example of how to list labels of site type associated to a site.

GET:

```
GET /api/v1/sites/allowed-labels
```

Request:

```
<empty>
```



Note Filtering, pagination, and sorting is currently not supported.

Response:

```
HTTP/1.1 200
{
  "labels": [
    {
      "id": "58202f7066e6e10001c41240",
      "name": "devops",
      "type": "site"
    },
    {
      "id": "58202fc466e6e10001c41241",
      "name": "prod",
      "type": "site"
    },
    {
      "id": "58202f7066e6e10001c41242",
      "name": "prod2",
      "type": "site"
    }
  ]
}
```

List Labels associated to a Site

The following is an example of how to list labels associated to a site.

GET:

```
GET /api/v1/sites/58202f7066e6e10001c41236/labels
```

Request:

```
<empty>
```



Note Filtering, pagination, and sorting is currently not supported.

Response:

```
HTTP/1.1 200
{
  "labels": [
    {
      "id": "58202f7066e6e10001c41240",
      "name": "devops",
      "type": "site"
    }
  ],
}
```

```
{
  "id": "58202fc466e6e10001c41241",
  "name" : "prod",
  "type" : "site"
}
```

APIs for Template

Execute a Template

The following is an example of how to execute a Template under a Schema to its associated sites.

GET:

```
api/v1/execute/schema/{schema-id}/template/{template-name}
```

Request:

Response:

```
HTTP/1.1 200
{"msg":"Successfully deployed"}
```

Undeploy a Template

The following is an example of how to undeploy a Template from its associated sites.

GET:

```
api/v1/execute/schema/{schema-id}/template/{template-name}?undeploy={site-id}
```

Request:

Response:

```
HTTP/1.1 200
{"msg":"Successfully deployed"}
```

APIs for Entities

Login Domains

The following is an example of how to list login domains.

List Domains:

```
GET https://<host>:<port>/api/v1/sites/:siteId/aci/domains?type=:type
```

Example:

```
GET https://<host>:<port>/api/v1/sites/58202f7066e6e10001c41236/aci/domains?type=vmm
```

Domain Types:

- vmm
- physical
- L2Ext
- l3Ext
- fibreChannel

Request:

```
{
  <empty>
}
```

Response:

```
{
  "domains": [
    {
      "name": "cSphere",
      "dn": "uni/vmmp-VMware/dom-cSphere"
    },
    {
      "name": "Fab1-DVS1",
      "dn": "uni/vmmp-VMware/dom-Fab1-DVS1"
    },
    {
      "name": "TestDomain-DoNotDelete",
      "dn": "uni/vmmp-VMware/dom-TestDomain-DoNotDelete"
    }
  ]
}
```

List L3Outs

The following is an example for listing L3Outs.

GET:

```
GET https://<host>:<port>/api/v1/sites/:id/aci/tenants/:name/l3-outs
```

Request

```
{
  <empty>
}
```

Response

```
{
  "l3Outs": [
    {
      "name": "TestL3Out-DoNotDelete",
      "dn": "uni/tn-TestTenant-DoNotDelete/out-TestL3Out-DoNotDelete",

```

```

        "networkInstanceProfiles": [
          {
            "name": "TestNIP-1-DoNotDelete",
            "dn":
"uni/tn-TestTenant-DoNotDelete/out-TestL3Out-DoNotDelete/instP-TestNIP-1-DoNotDelete"
          },
          {
            "name": "TestNIP-2-DoNotDelete",
            "dn":
"uni/tn-TestTenant-DoNotDelete/out-TestL3Out-DoNotDelete/instP-TestNIP-2-DoNotDelete"
          }
        ]
      },
      {
        "name": "TestL3Out2",
        "dn": "uni/tn-TestTenant-DoNotDelete/out-TestL3Out2",
        "networkInstanceProfiles": [
          {
            "name": "N1",
            "dn": "uni/tn-TestTenant-DoNotDelete/out-TestL3Out2/Instp-N1"
          }
        ]
      }
    ]
  }
}

```

APIs for Platform

Read Node Information

The following is an example of how to read node information.

GET:

```
GET /api/v1/platform/nodes
```

Request:

<empty>

Response:

```

HTTP/1.1 200
{
  "nodes": [
    {
      "id": "cfihme4upn2yfpgad7xyv9u7",
      "state": "ready",
      "ip": "192.168.65.2",
      "hostname": "moby"
    }
  ]
}

```

Download Troubleshooting Report

The following is an example of how to download troubleshooting report.

GET:

```
GET /api/v1/platform/logs?include=schemas,sites,tenants,users,infra_logs
```

Request:

```
<empty>
```

Response:

```
HTTP/1.1 200  
zip file named msc_report_<timestamp>.zip
```

Expected HTTP Response Code Examples

The following are examples of a few HTTP response codes:

400: Bad Request

```
{  
  "code": 125,  
  "info": {  
    "obj.password": [  
      "Missing path"  
    ],  
    "obj.username": [  
      "Missing path"  
    ],  
    "obj.urls": [  
      "Invalid urls"  
    ]  
  },  
  "message": "Malformed body"  
}
```

404: Entity not found or invalid endpoint

```
{  
  "code": 404,  
  "message": "Not Found"  
}
```

409: Conflict - Duplicate entity/resource

```
{  
  "code": 140,  
  "message": "Duplicate Resource"  
}
```

APIs for Audit

About Auditing

Starting with Cisco ACI Multi-Site, release 1.1(1), the API to create, read, and filter audit records is supported. Therefore, the auditing examples are applicable starting with release 1.1(1) only.

APIs listed in this section are granular and enable you to read and insert audit records across various Multi-Site entities.



Note The auditing APIs are accessible only to administrators and users with the Power User role.

Downloading Audit Records: When downloading audit logs, one of the following APIs are invoked:

GET:

```
GET https://<host>:<port>/api/v1/audit-records/download?format=csv
GET https://<host>:<port>/api/v1/audit-records/download?Format=csv
```

Filter Audit Records

GET:

```
GET https://<host>:<port>/api/v1/audit-records?type=tenant
```

Request:

```
<empty>
```

Response:

```
HTTP/1.1 200
{
  "auditRecords": [
    {
      "id": "59b5e11f1a0100270319b05e",
      "type": "tenant",
      "key": "59b5e11f3400004900c0ea97",
      "name": "Tenant 001",
      "event": "created",
      "user": {
        "userId": "0000ffff00000000000000020",
        "username": "Admin User",
        "firstName": "Admin",
        "lastName": "User",
        "domainId": "0000ffff00000000000000090",
        "domainName": "Local"
      },
      "timestamp": "2017-09-10T01:04:31.701"
    },
    {
      "id": "59b5e11f1a0100270319b05e",
```

```

    "type": "tenant",
    "key": "59b5e11f3400004900c0ea97",
    "name": "Tenant 001",
    "event": "updated",
    "user": {
      "userId": "0000ffff0000000000000020",
      "username": "Admin User",
      "firstName": "Admin",
      "lastName": "User",
      "domainId": "0000ffff0000000000000090",
      "domainName": "Local"
    },
    "timestamp": "2017-09-10T01:04:35.501"
  }
]
}

```

Filter Audit Records by **container**

The following are examples of container-type:

- tenant
- schema
- site
- fabric-connectivity-site

The following are examples of container-key:

- <id>
- <ref>

GET:

```

GET
https://<host>:<port>/api/v1/audit-records?container-type=tenant&container-key=59b5e11f3400004900c0ea97

```

Request:

<empty>

Response:

```

HTTP/1.1 200
{
  "auditRecords": [
    {
      "id": "59b5e1561a0100300319b05f",
      "type": "tenant-site",
      "key": "59cbf2194500004500741bd3",
      "name": "Site 1",
      "event": "added",
      "container": {
        "type": "tenant",
        "key": "59b5e11f3400004900c0ea97",
        "name": "Tenant 001"
      },
      "user": {

```



```

        "userId": "0000ffff0000000000000020",
        "username": "Admin User",
        "firstName": "Admin",
        "lastName": "User",
        "domainId": "0000ffff0000000000000090",
        "domainName": "Local"
    },
    "timestamp": "2017-09-10T01:04:35.501"
}
]
}

```

Filter Audit Records by **container** and **context**.

The following are examples of container-type:



Note (Values are the same as "type" values)

- tenant
- schema
- site

The following are examples of container-key:

- <id>
- <ref>

The following are examples of context-type:

- site
- template

The following are examples of context-key:

- <id>
- <ref>

The following are examples of context-key:

GET:

```
GET https://<host>:<port>/api/v1/audit-records?container-type=<type>
&container-key=<key>&context-key=<key>&context-type=<type>&context-key=<key>
```

```
GET https://<host>:<port>/api/v1/audit-records?container-type=tenant&container-key=
/schemas/59ef87743800008e06881fe5/templates/Template1/ans/App-1/
epgs/Web-EPG&context-type=site&context-key=/schemas/59ef87743800008e06881fe5/templates
/Template1/sites/59726e89490000ea095b4cc9
```

Request:

<empty>

Response:

```
HTTP/1.1 200
{
  "auditRecords": [
    {
      "id": "5a3aacc52100003d007f4c1a",
      "type": "attribute",
      "key":
"/schemas/59ef87743800008e06881fe5/templates/Template1/anps/App-1/epgs/Web-EPG/subnets",
      "event": "added",
      "container": {
        "type": "epg",
        "key":
"/schemas/59ef87743800008e06881fe5/templates/Template1/anps/App-1/epgs/Web-EPG",
        "name": "Web-EPG",
        "context": {
          "type": "site",
          "key":
"/schemas/59ef87743800008e06881fe5/templates/Template1/sites/59726e89490000ea095b4cc9",
          "name": "New York"
        }
      },
      "newValue": "10.0.0.1/24",
      "user": {
        "userId": "0000ffff00000000000000020",
        "username": "admin",
        "firstName": "Admin",
        "lastName": "User",
        "domainId": "0000ffff00000000000000090",
        "domainName": "Local"
      },
      "timestamp": "2017-12-20T18:32:37"
    }
  ]
}
```

Filter Audit Records by **username**

GET:

GET https://<host>:<port>/api/v1/audit-records?username=admin

GET:

GET https://<host>:<port>/api/v1/audit-records?username=admin,bob,lisa

Filter Audit Records by **event**

GET:

GET https://<host>:<port>/api/v1/audit-records?event=updated

GET:

GET https://<host>:<port>/api/v1/audit-records?event=associated,added

Filter Audit Records by **timestamp**

GET:

```
GET https://<host>:<port>/api/v1/audit-records?timestamp=2017-09-10T01:04:31.701
```

Filter Audit Records by date/time range

GET:

```
GET
```

```
https://<host>:<port>/api/v1/audit-records?start=2017-09-10T01:04:31.701&end=2017-09-15T08:06:45.811
```

GET:

```
GET https://<host>:<port>/api/v1/audit-records?start=2017-09-10T07:00&end=2017-09-15T07:00
```

Combine multiple filters

GET:

```
GET https://<host>:<port>/api/v1/audit-records?username=admin&event=associated,added
```

```
GET
```

```
https://<host>:<port>/api/v1/audit-records?username=admin&start=2017-09-10T07:00&end=2017-09-15T07:00
```

Paginate Audit Records

The following is an example of how to paginate audit records.

Use the parameters offset and limit for this purpose. Both parameters are optional but defaults apply to every GET request that retrieves audit records. The default values are as follows:

- offset: 0
- limit: 20



Note The maximum value for the limit is 250.

The total number of records is available in the response header named "X-Total-Count".

The prev and next "Link" headers are populated when applicable.

GET:

```
GET https://<host>:<port>/api/v1/audit-records?offset=3&limit=3
```

Request:

```
<empty>
```

Response:

```
HTTP/1.1 200
Link: </api/v1/audit-records?offset=0&limit=3>; rel="prev",
</api/v1/audit-records?offset=6&limit=3>; rel="next"
X-Total-Count: 76
{
  "auditRecords": [
    {
      "id": "59ed6de03d00003800175a2c",
      "type": "site",
      "key": "59ed6de04900004c00fcd96b",
```

```

    "name": "Site 1",
    "event": "created",
    "user": {
      "userId": "0000ffff0000000000000020",
      "username": "admin",
      "firstName": "Admin",
      "lastName": "User",
      "domainId": "0000ffff0000000000000090",
      "domainName": "Local"
    },
    "timestamp": "2017-10-23T04:19:44.355"
  },
  {
    "id": "59ed6de03d00003800175a2d",
    "type": "attribute",
    "key": "/sites/59ed6de04900004c00fcd96b/name",
    "event": "created",
    "container": {
      "type": "site",
      "key": "59ed6de04900004c00fcd96b",
      "name": "Site 1"
    },
    "newValue": "Site 1",
    "user": {
      "userId": "0000ffff0000000000000020",
      "username": "admin",
      "firstName": "Admin",
      "lastName": "User",
      "domainId": "0000ffff0000000000000090",
      "domainName": "Local"
    },
    "timestamp": "2017-10-23T04:19:44.355"
  },
  {
    "id": "59ed6de03d00003800175a2e",
    "type": "attribute",
    "key": "/sites/59ed6de04900004c00fcd96b/username",
    "event": "created",
    "container": {
      "type": "site",
      "key": "59ed6de04900004c00fcd96b",
      "name": "Site 1"
    },
    "newValue": "admin",
    "user": {
      "userId": "0000ffff0000000000000020",
      "username": "admin",
      "firstName": "Admin",
      "lastName": "User",
      "domainId": "0000ffff0000000000000090",
      "domainName": "Local"
    },
    "timestamp": "2017-10-23T04:19:44.355"
  }
}
]
}

```

About Login Domains Auditing

Login Domains Auditing APIs are accessible by the following users with the following roles:

- admin

- Power User
- User and Role Manager

List or Filter Login Domains Audit Records

Login Domains Audit Records

The following examples provide a high-level audit view of Login Domains that are created, updated, or deleted.



Note Default limit and offset values apply. Multiple additional filters (such as username, event, timestamp, date range) may be applied. Filters may also be combined.

GET:

```
GET https://<host>:<port>/api/v1/auth/domains/audit-records
```

GET:

```
GET https://<host>:<port>/api/v1/auth/domains/audit-records?offset=0&limit=3
```

GET:

```
GET https://<host>:<port>/api/v1/auth/domains/audit-records?username=admin&event=created,updated
```

Single Login Domain Audit Records

The following examples provide a detailed audit view of a single Login Domain.



Note Default limit and offset values apply. Multiple additional filters (such as username, event, timestamp, date range) may be applied. Filters may also be combined.

GET:

```
GET https://<host>:<port>/api/v1/auth/domains/:id/audit-records
```

GET:

```
GET https://<host>:<port>/api/v1/auth/domains/:id/audit-records?offset=0&limit=3
```

GET:

```
GET https://<host>:<port>/api/v1/auth/domains/:id/audit-records?username=admin&event=created,updated
```

GET:

```
GET https://<host>:<port>/api/v1/auth/domains/:id/audit-records?timestamp=2017-10-15T01:04:35.603
```

All Users Associated with Login Domain Audit Records

The following examples are for all users associated with Login Domain audit records.

GET:

```
GET https://<host>:<port>/api/v1/auth/domains/audit-records/users
```

GET:

```
GET https://<host>:<port>/api/v1/auth/domains/:id/audit-records/users
```

About Users Auditing

The Users Auditing APIs are accessible with the following roles:

- admin
- Power User
- User and Role Manager

List or Filter User Audit Records

User Audit Records

The following examples provide a high-level audit view of users that were created, updated, or deleted.



Note Default limit and offset values apply. Multiple additional filters (such as username, event, timestamp, date range) may be applied. Filters may also be combined.

GET:

```
GET https://<host>:<port>/api/v1/users/audit-records
```

GET:

```
GET https://<host>:<port>/api/v1/users/audit-records?offset=0&limit=3
```

GET:

```
GET https://<host>:<port>/api/v1/users/audit-records?username=admin&event=created,updated
```

Single User Audit Records

The following examples provide a detailed audit view of a single user.



Note Default limit and offset values apply. Multiple additional filters (such as username, event, timestamp, date range) may be applied. Filters may also be combined.

GET:

```
GET https://<host>:<port>/api/v1/users/:id/audit-records
```

GET:

```
GET https://<host>:<port>/api/v1/users/:id/audit-records?offset=0&limit=3
```

GET:

```
GET https://<host>:<port>/api/v1/users/:id/audit-records?username=admin&event=created,updated
```

GET:

```
GET https://<host>:<port>/api/v1/users/:id/audit-records?timestamp=2017-10-15T01:04:35.603
```

All Users Associated with User Audit Records

The following examples are for all users associated with user audit records.

GET:

```
GET https://<host>:<port>/api/v1/users/audit-records/users
```

GET:

```
GET https://<host>:<port>/api/v1/users/:id/audit-records/users
```

About Providers Auditing

The Providers Auditing APIs are accessible by the following users with the following roles:

- admin
- Power User
- User and Role Manager

List or Filter Provider Audit Records

Providers Audit Records

The following examples provide a high-level audit view of providers that were created, updated, or deleted.



Note Default limit and offset values apply. Multiple additional filters (such as username, event, timestamp, date range) may be applied. Filters may also be combined.

GET:

```
GET https://<host>:<port>/api/v1/auth/providers/audit-records
```

GET:

```
GET https://<host>:<port>/api/v1/auth/providers/audit-records?offset=0&limit=3
```

GET:

```
GET
https://<host>:<port>/api/v1/auth/providers/audit-records?username=admin&event=created,updated
```

Single Provider Audit Records

The following examples provide a detailed audit view of a single provider.



Note Default limit and offset values apply. Multiple additional filters (such as username, event, timestamp, date range) may be applied. Filters may also be combined.

GET:

```
GET https://<host>:<port>/api/v1/auth/providers/:id/audit-records
```

GET:

```
GET https://<host>:<port>/api/v1/auth/providers/:id/audit-records?offset=0&limit=3
```

GET:

```
GET
https://<host>:<port>/api/v1/auth/providers/:id/audit-records?username=admin&event=created,updated
```

GET:

```
GET
https://<host>:<port>/api/v1/auth/providers/:id/audit-records?timestamp=2017-10-15T01:04:35.603
```

All Users Associated with Provider Audit Records

The following examples are for all users associated with provider audit records.

GET:

```
GET https://<host>:<port>/api/v1/auth/providers/audit-records/users
```

GET:

```
GET https://<host>:<port>/api/v1/auth/providers/:id/audit-records/users
```

List or Filter RADIUS Provider Audit Records

RADIUS Providers Audit Records

The following examples provide a high-level audit view of RADIUS providers that were created, updated, or deleted.



Note Default limit and offset values apply. Multiple additional filters (such as username, event, timestamp, date range) may be applied. Filters may also be combined.

GET:

```
GET https://<host>:<port>/api/v1/auth/providers/radius/audit-records
```

GET:

```
GET https://<host>:<port>/api/v1/auth/providers/radius/audit-records?offset=0&limit=3
```

GET:

```
GET
```

```
https://<host>:<port>/api/v1/auth/providers/radius/audit-records?username=admin&event=created,updated
```

Single RADIUS Provider Audit Records

The following examples provide a detailed audit view of a single RADIUS provider.



Note Default limit and offset values apply. Multiple additional filters (such as username, event, timestamp, date range) may be applied. Filters may also be combined.

GET:

```
GET https://<host>:<port>/api/v1/auth/providers/radius/:id/audit-records
```

GET:

```
GET https://<host>:<port>/api/v1/auth/providers/radius/:id/audit-records?offset=0&limit=3
```

GET:

```
GET
```

```
https://<host>:<port>/api/v1/auth/providers/radius/:id/audit-records?username=admin&event=created,updated
```

GET:

```
GET
```

```
https://<host>:<port>/api/v1/auth/providers/radius/:id/audit-records?timestamp=2017-10-15T01:04:35.603
```

All Users Associated with RADIUS Provider Audit Records

The following examples are for all users associated with RADIUS provider audit records.

GET:

```
GET https://<host>:<port>/api/v1/auth/providers/radius/audit-records/users
```

GET:

```
GET https://<host>:<port>/api/v1/auth/providers/radius/:id/audit-records/users
```

List or Filter TACACS Provider Audit Records

TACACS Providers Audit Records

The following examples provide a high-level audit view of TACACS providers that were created, updated, or deleted.



Note Default limit and offset values apply. Multiple additional filters (such as username, event, timestamp, date range) may be applied. Filters may also be combined.

GET:

```
GET https://<host>:<port>/api/v1/auth/providers/tacacs/audit-records
```

GET:

```
GET https://<host>:<port>/api/v1/auth/providers/tacacs/audit-records?offset=0&limit=3
```

GET:

```
GET
https://<host>:<port>/api/v1/auth/providers/tacacs/audit-records?username=admin&event=created,updated
```

Single TACACS Provider Audit Records

The following examples provide a detailed audit view of a single TACACS provider.



Note Default limit and offset values apply. Multiple additional filters (such as username, event, timestamp, date range) may be applied. Filters may also be combined.

GET:

```
GET https://<host>:<port>/api/v1/auth/providers/tacacs/:id/audit-records
```

GET:

```
GET https://<host>:<port>/api/v1/auth/providers/tacacs/:id/audit-records?offset=0&limit=3
```

GET:

```
GET
https://<host>:<port>/api/v1/auth/providers/tacacs/:id/audit-records?username=admin&event=created,updated
```

GET:

```
GET
https://<host>:<port>/api/v1/auth/providers/tacacs/:id/audit-records?timestamp=2017-10-15T01:04:35.603
```

All Users Associated with TACACS Provider Audit Records

The following examples are for all users associated with TACACS provider audit records.

GET:

```
GET https://<host>:<port>/api/v1/auth/providers/tacacs/audit-records/users
```

GET:

```
GET https://<host>:<port>/api/v1/auth/providers/tacacs/:id/audit-records/users
```

List or Filter Audit Records for Tenants



Note

- Default limit and offset values apply.
- Multiple additional filters (such as username, event, timestamp, date range) may be applied. Filters may also be combined.

The following example shows how to list or filter audit records for all Tenants. These APIs provide a high-level audit view of tenants that were created, updated, or deleted.

GET:

```
GET https://<host>:<port>/api/v1/tenants/audit-records
```

```
GET https://<host>:<port>/api/v1/tenants/audit-records?offset=0&limit=3
```

```
GET https://<host>:<port>/api/v1/tenants/audit-records?username=admin&event=created,updated
```

Request:

<empty>

Response:

HTTP/1.1 200

```
{
  "auditRecords": [
    {
      "id": "59db14881e0000f00071f40e",
      "type": "tenant",
      "key": "59db147b2500000b01c90429",
      "name": "Tenant 001",
      "event": "updated",
      "user": {
        "userId": "0000ffff0000000000000020",
        "username": "admin",
        "firstName": "Admin",
        "lastName": "User",
        "domainId": "0000ffff0000000000000090",
        "domainName": "Local"
      },
      "timestamp": "2017-10-09T06:17:44.141"
    },
    {
      "id": "59db147c1e0000f00071f40c",
      "type": "tenant",
      "key": "59db147b2500000b01c90429",
      "name": "Tenant 001",
      "event": "created",
    }
  ]
}
```

```

    "user": {
      "userId": "0000ffff00000000000000020",
      "username": "admin",
      "firstName": "Admin",
      "lastName": "User",
      "domainId": "0000ffff00000000000000090",
      "domainName": "Local"
    },
    "timestamp": "2017-10-09T06:17:31.873"
  },
  {
    "id": "59b5e11f1a0100270319b06a",
    "type": "tenant",
    "key": "59b5e11f3400004900c0ea98",
    "name": "Tenant 000",
    "event": "created",
    "user": {
      "userId": "0000ffff00000000000000020",
      "username": "admin",
      "firstName": "Admin",
      "lastName": "User",
      "domainId": "0000ffff00000000000000090",
      "domainName": "Local"
    },
    "timestamp": "2017-09-15T01:04:35.501"
  }
]
}

```

The following example shows how to list or filter a single Tenant's audit records.

GET:

```
GET https://<host>:<port>/api/v1/tenants/:id/audit-records
```

```
GET https://<host>:<port>/api/v1/tenants/:id/audit-records?offset=0&limit=4
```

```
GET
```

```
https://<host>:<port>/api/v1/tenants/:id/audit-records?username=admin&event=created,updated
```

```
GET https://<host>:<port>/api/v1/tenants/:id/audit-records?timestamp=2017-09-15T01:04:35.501
```

Request:

```
<empty>
```

Response:

```
HTTP/1.1 200
```

```

{
  "auditRecords": [
    {
      "id": "59db14881e0000f00071f40f",
      "type": "attribute",
      "key": "/tenants/59db147b2500000b01c90429/description",
      "event": "updated",
      "container": {
        "type": "tenant",
        "key": "59db147b2500000b01c90429",
        "name": "Tenant 001"
      }
    }
  ]
}

```

```

    },
    "oldValue": "",
    "newValue": "Description for Tenant 001",
    "user": {
      "userId": "0000ffff0000000000000020",
      "username": "admin",
      "firstName": "Admin",
      "lastName": "User",
      "domainId": "0000ffff0000000000000090",
      "domainName": "Local"
    },
    "timestamp": "2017-10-09T06:17:44.141"
  },
  {
    "id": "59db14881e0000f00071f410",
    "type": "tenant-site",
    "key": "59d3f3254c00004c008172c5",
    "name": "Site 1",
    "event": "associated",
    "container": {
      "type": "tenant",
      "key": "59db147b250000b01c90429",
      "name": "Tenant 001"
    },
    "user": {
      "userId": "0000ffff0000000000000020",
      "username": "admin",
      "firstName": "Admin",
      "lastName": "User",
      "domainId": "0000ffff0000000000000090",
      "domainName": "Local"
    },
    "timestamp": "2017-10-09T06:17:44.141"
  },
  {
    "id": "59db14881e0000f00071f411",
    "type": "tenant-user",
    "key": "0000ffff0000000000000020",
    "name": "admin",
    "event": "associated",
    "container": {
      "type": "tenant",
      "key": "59db147b250000b01c90429",
      "name": "Tenant 001"
    },
    "user": {
      "userId": "0000ffff0000000000000020",
      "username": "admin",
      "firstName": "Admin",
      "lastName": "User",
      "domainId": "0000ffff0000000000000090",
      "domainName": "Local"
    },
    "timestamp": "2017-10-09T06:17:31.873"
  },
  {
    "id": "59db147c1e0000f00071f40d",
    "type": "attribute",
    "key": "/tenants/59db147b250000b01c90429/displayName",
    "event": "created",
    "container": {
      "type": "tenant",
      "key": "59db147b250000b01c90429",
      "name": "Tenant 001"
    }
  }

```

```

    },
    "oldValue": "",
    "newValue": "Tenant 001",
    "user": {
      "userId": "0000ffff00000000000000020",
      "username": "admin",
      "firstName": "Admin",
      "lastName": "User",
      "domainId": "0000ffff00000000000000090",
      "domainName": "Local"
    },
    "timestamp": "2017-10-09T06:17:31.873"
  }
]
}

```

Find Users Associated with Tenant Audit Records

The following example shows how to find all users associated with Tenant audit records.

GET:

```
GET https://<host>:<port>/api/v1/tenants/audit-records/users
```

```
GET https://<host>:<port>/api/v1/tenants/:id/audit-records/users
```

Request:

<empty>

Response:

```

HTTP/1.1 200
{
  "auditUsers": [
    {
      "userId": "0000ffff00000000000000020",
      "username": "admin",
      "firstName": "Admin",
      "lastName": "User",
      "domainId": "0000ffff00000000000000090",
      "domainName": "Local"
    },
    {
      "userId": "59ed6de03d00003800175a2e",
      "username": "bob",
      "firstName": "Bob",
      "lastName": "Smith",
      "domainId": "0000ffff00000000000000090",
      "domainName": "Local"
    }
  ]
}

```

List or Filter Audit Records for Schemas



Note

- Default limit and offset values apply.
- Multiple additional filters (such as username, event, timestamp, date range) may be applied. Filters may also be combined. This is a powerful API, and it allows you to filter records related to, for example, all ANPs in a template, a particular ANP, or all policies in a template.

The following example shows how to list or filter audit records for all Schemas. These APIs provide a high-level audit view of schemas that were created, updated, or deleted.

GET:

```
GET https://<host>:<port>/api/v1/schemas/audit-records
```

```
GET https://<host>:<port>/api/v1/schemas/audit-records?offset=0&limit=3
```

```
GET https://<host>:<port>/api/v1/schemas/audit-records?username=admin&event=created,updated
```

Request:

<empty>

Response:

```
HTTP/1.1 200
{
  "auditRecords": [
    {
      "id": "59ef87953a00001f069baf83",
      "type": "schema",
      "key": "59ef87743800008e06881fe5",
      "name": "Test Schema",
      "event": "updated",
      "user": {
        "userId": "0000ffff0000000000000020",
        "username": "admin",
        "firstName": "Admin",
        "lastName": "User",
        "domainId": "0000ffff0000000000000090",
        "domainName": "Local"
      },
      "timestamp": "2017-10-24T18:33:57.37"
    },
    {
      "id": "59ef87743a000022069baf60",
      "type": "schema",
      "key": "59ef87743800008e06881fe5",
      "name": "Test Schema",
      "event": "created",
      "user": {
        "userId": "0000ffff0000000000000020",
        "username": "admin",
        "firstName": "Admin",
        "lastName": "User",

```

```

        "domainId": "0000ffff0000000000000090",
        "domainName": "Local"
    },
    "timestamp": "2017-10-24T18:33:24.584"
}
]
}

```

The following example shows how to list or filter a single Schema audit records. It shows audit records generated by a single schema.

GET:

```
GET https://<host>:<port>/api/v1/schemas/:id/audit-records
```

```
GET https://<host>:<port>/api/v1/schemas/:id/audit-records?offset=0&limit=4
```

```
GET
```

```
https://<host>:<port>/api/v1/schemas/:id/audit-records?username=admin&event=created,updated
```

Request:

```
<empty>
```

Response:

```
HTTP/1.1 200
```

```

{
  "auditRecords": [
    {
      "id": "59ef87743a000022069baf60",
      "type": "schema",
      "key": "59ef87743800008e06881fe5",
      "name": "Test Schema",
      "event": "created",
      "user": {
        "userId": "0000ffff0000000000000020",
        "username": "admin",
        "firstName": "Admin",
        "lastName": "User",
        "domainId": "0000ffff0000000000000090",
        "domainName": "Local"
      },
      "timestamp": "2017-10-24T18:33:24.584"
    },
    {
      "id": "59ef87743a000022069baf61",
      "type": "template",
      "key": "schemas/59ef87743800008e06881fe5/templates/Template1",
      "name": "Template 1",
      "event": "added",
      "container": {
        "type": "schema",
        "key": "59ef87743800008e06881fe5",
        "name": "Test Schema"
      },
      "user": {
        "userId": "0000ffff0000000000000020",
        "username": "admin",

```



```

        "firstName": "Admin",
        "lastName": "User",
        "domainId": "0000ffff0000000000000090",
        "domainName": "Local"
    },
    "timestamp": "2017-10-24T18:33:24.584"
},
{
    "id": "59ef87743a000022069baf62",
    "type": "attribute",
    "key": "schemas/59ef87743800008e06881fe5/templates/Template1/displayName",
    "event": "created",
    "container": {
        "type": "template",
        "key": "schemas/59ef87743800008e06881fe5/templates/Template1",
        "name": "Template 1"
    },
    "newValue": "Template 1",
    "user": {
        "userId": "0000ffff0000000000000020",
        "username": "admin",
        "firstName": "Admin",
        "lastName": "User",
        "domainId": "0000ffff0000000000000090",
        "domainName": "Local"
    },
    "timestamp": "2017-10-24T18:33:24.584"
},
{
    "id": "59ef87743a000022069baf63",
    "type": "anp",
    "key": "/schemas/59ef87743800008e06881fe5/templates/Template1/anps/App-1",
    "name": "App-1",
    "event": "created",
    "container": {
        "type": "template",
        "key": "schemas/59ef87743800008e06881fe5/templates/Template1",
        "name": "Template 1"
    },
    "user": {
        "userId": "0000ffff0000000000000020",
        "username": "admin",
        "firstName": "Admin",
        "lastName": "User",
        "domainId": "0000ffff0000000000000090",
        "domainName": "Local"
    },
    "timestamp": "2017-10-24T18:33:24.584"
},
{
    "id": "59ef87743a000022069baf64",
    "type": "attribute",
    "key": "/schemas/59ef87743800008e06881fe5/templates/Template1/anps/App-1/displayName",
    "event": "created",
    "container": {
        "type": "anp",
        "key": "/schemas/59ef87743800008e06881fe5/templates/Template1/anps/App-1",
        "name": "App-1"
    },
    "newValue": "App-1",
    "user": {
        "userId": "0000ffff0000000000000020",
        "username": "admin",

```

```

        "firstName": "Admin",
        "lastName": "User",
        "domainId": "0000ffff0000000000000090",
        "domainName": "Local"
    },
    "timestamp": "2017-10-24T18:33:24.584"
},
{
    "id": "59ef87743a000022069baf65",
    "type": "epg",
    "key": "/schemas/59ef87743800008e06881fe5/templates/Template1/ans/App-1/eggs/Web-EPG",

    "name": "Web-EPG",
    "event": "created",
    "container": {
        "type": "template",
        "key": "schemas/59ef87743800008e06881fe5/templates/Template1",
        "name": "Template 1"
    },
    "user": {
        "userId": "0000ffff0000000000000020",
        "username": "admin",
        "firstName": "Admin",
        "lastName": "User",
        "domainId": "0000ffff0000000000000090",
        "domainName": "Local"
    },
    "timestamp": "2017-10-24T18:33:24.584"
},
{
    "id": "59ef87743a000022069baf66",
    "type": "attribute",
    "key":
"/schemas/59ef87743800008e06881fe5/templates/Template1/ans/App-1/eggs/Web-EPG/displayName",

    "event": "created",
    "container": {
        "type": "epg",
        "key":
"/schemas/59ef87743800008e06881fe5/templates/Template1/ans/App-1/eggs/Web-EPG",
        "name": "Web-EPG"
    },
    "newValue": "Web-EPG",
    "user": {
        "userId": "0000ffff0000000000000020",
        "username": "admin",
        "firstName": "Admin",
        "lastName": "User",
        "domainId": "0000ffff0000000000000090",
        "domainName": "Local"
    },
    "timestamp": "2017-10-24T18:33:24.584"
},
{
    "id": "59ef87743a000022069baf67",
    "type": "attribute",
    "key":
"/schemas/59ef87743800008e06881fe5/templates/Template1/ans/App-1/eggs/Web-EPG/uSegEpg",

    "event": "created",
    "container": {
        "type": "epg",
        "key":
"/schemas/59ef87743800008e06881fe5/templates/Template1/ans/App-1/eggs/Web-EPG",
        "name": "Web-EPG"
    }
}

```

```

    },
    "newValue": "false",
    "user": {
      "userId": "0000ffff00000000000000020",
      "username": "admin",
      "firstName": "Admin",
      "lastName": "User",
      "domainId": "0000ffff00000000000000090",
      "domainName": "Local"
    },
    "timestamp": "2017-10-24T18:33:24.584"
  },
  {
    "id": "59ef87743a000022069baf68",
    "type": "attribute",
    "key":
"/schemas/59ef87743800008e06881fe5/templates/Template1/ansp/App-1/epgs/Web-EPG/intraEpg",
    "event": "created",
    "container": {
      "type": "epg",
      "key":
"/schemas/59ef87743800008e06881fe5/templates/Template1/ansp/App-1/epgs/Web-EPG",
      "name": "Web-EPG"
    },
    "newValue": "unenforced",
    "user": {
      "userId": "0000ffff00000000000000020",
      "username": "admin",
      "firstName": "Admin",
      "lastName": "User",
      "domainId": "0000ffff00000000000000090",
      "domainName": "Local"
    },
    "timestamp": "2017-10-24T18:33:24.584"
  },
  {
    "id": "59ef87743a000022069baf69",
    "type": "attribute",
    "key":
"/schemas/59ef87743800008e06881fe5/templates/Template1/ansp/App-1/epgs/Web-EPG/proxyArp",
    "event": "created",
    "container": {
      "type": "epg",
      "key":
"/schemas/59ef87743800008e06881fe5/templates/Template1/ansp/App-1/epgs/Web-EPG",
      "name": "Web-EPG"
    },
    "newValue": "false",
    "user": {
      "userId": "0000ffff00000000000000020",
      "username": "admin",
      "firstName": "Admin",
      "lastName": "User",
      "domainId": "0000ffff00000000000000090",
      "domainName": "Local"
    },
    "timestamp": "2017-10-24T18:33:24.584"
  },
  {
    "id": "59ef87743a000022069baf6a",
    "type": "attribute",
    "key":
"/schemas/59ef87743800008e06881fe5/templates/Template1/ansp/App-1/epgs/Web-EPG/bdRef",
    "event": "associated",

```

```

        "container": {
            "type": "epg",
            "key":
"/schemas/59ef87743800008e06881fe5/templates/Template1/anps/App-1/epgs/Web-EPG",
            "name": "Web-EPG"
        },
        "newValue": "/schemas/59ef87743800008e06881fe5/templates/Template1/bds/BD1",
        "user": {
            "userId": "0000ffff00000000000000020",
            "username": "admin",
            "firstName": "Admin",
            "lastName": "User",
            "domainId": "0000ffff00000000000000090",
            "domainName": "Local"
        },
        "timestamp": "2017-10-24T18:33:24.584"
    },
    {
        "id": "59ef87743a000022069baf6b",
        "type": "epg",
        "key": "/schemas/59ef87743800008e06881fe5/templates/Template1/anps/App-1/epgs/DB-EPG",

        "name": "DB-EPG",
        "event": "created",
        "container": {
            "type": "template",
            "key": "schemas/59ef87743800008e06881fe5/templates/Template1",
            "name": "Template 1"
        },
        "user": {
            "userId": "0000ffff00000000000000020",
            "username": "admin",
            "firstName": "Admin",
            "lastName": "User",
            "domainId": "0000ffff00000000000000090",
            "domainName": "Local"
        },
        "timestamp": "2017-10-24T18:33:24.584"
    },
    {
        "id": "59ef87743a000022069baf6c",
        "type": "attribute",
        "key":
"/schemas/59ef87743800008e06881fe5/templates/Template1/anps/App-1/epgs/DB-EPG/displayName",

        "event": "created",
        "container": {
            "type": "epg",
            "key": "/schemas/59ef87743800008e06881fe5/templates/Template1/anps/App-1/epgs/DB-EPG",

            "name": "DB-EPG"
        },
        "newValue": "DB-EPG",
        "user": {
            "userId": "0000ffff00000000000000020",
            "username": "admin",
            "firstName": "Admin",
            "lastName": "User",
            "domainId": "0000ffff00000000000000090",
            "domainName": "Local"
        },
        "timestamp": "2017-10-24T18:33:24.584"
    },
    {

```

```

      "id": "59ef87743a000022069baf6d",
      "type": "attribute",
      "key":
"/schemas/59ef87743800008e06881fe5/templates/Template1/ans/App-1/epgs/DB-EPG/uSegEpg",
      "event": "created",
      "container": {
        "type": "epg",
        "key": "/schemas/59ef87743800008e06881fe5/templates/Template1/ans/App-1/epgs/DB-EPG",

        "name": "DB-EPG"
      },
      "newValue": "false",
      "user": {
        "userId": "0000ffff00000000000000020",
        "username": "admin",
        "firstName": "Admin",
        "lastName": "User",
        "domainId": "0000ffff00000000000000090",
        "domainName": "Local"
      },
      "timestamp": "2017-10-24T18:33:24.584"
    },
    {
      "id": "59ef87743a000022069baf6e",
      "type": "attribute",
      "key":
"/schemas/59ef87743800008e06881fe5/templates/Template1/ans/App-1/epgs/DB-EPG/intraEpg",
      "event": "created",
      "container": {
        "type": "epg",
        "key": "/schemas/59ef87743800008e06881fe5/templates/Template1/ans/App-1/epgs/DB-EPG",

        "name": "DB-EPG"
      },
      "newValue": "unenforced",
      "user": {
        "userId": "0000ffff00000000000000020",
        "username": "admin",
        "firstName": "Admin",
        "lastName": "User",
        "domainId": "0000ffff00000000000000090",
        "domainName": "Local"
      },
      "timestamp": "2017-10-24T18:33:24.584"
    },
    {
      "id": "59ef87743a000022069baf6f",
      "type": "attribute",
      "key":
"/schemas/59ef87743800008e06881fe5/templates/Template1/ans/App-1/epgs/DB-EPG/proxyArp",
      "event": "created",
      "container": {
        "type": "epg",
        "key": "/schemas/59ef87743800008e06881fe5/templates/Template1/ans/App-1/epgs/DB-EPG",

        "name": "DB-EPG"
      },
      "newValue": "false",
      "user": {
        "userId": "0000ffff00000000000000020",
        "username": "admin",
        "firstName": "Admin",
        "lastName": "User",
        "domainId": "0000ffff00000000000000090",

```

```

        "domainName": "Local"
    },
    "timestamp": "2017-10-24T18:33:24.584"
},
{
    "id": "59ef87743a000022069baf70",
    "type": "attribute",
    "key":
"/schemas/59ef87743800008e06881fe5/templates/Template1/ans/App-1/egps/DB-EPG/bdRef",
    "event": "associated",
    "container": {
        "type": "epg",
        "key": "/schemas/59ef87743800008e06881fe5/templates/Template1/ans/App-1/egps/DB-EPG",

        "name": "DB-EPG"
    },
    "newValue": "/schemas/59ef87743800008e06881fe5/templates/Template1/bds/BD1",
    "user": {
        "userId": "0000ffff00000000000000020",
        "username": "admin",
        "firstName": "Admin",
        "lastName": "User",
        "domainId": "0000ffff0000000000000090",
        "domainName": "Local"
    },
    "timestamp": "2017-10-24T18:33:24.584"
},
{
    "id": "59ef87743a000022069baf71",
    "type": "bd",
    "key": "/schemas/59ef87743800008e06881fe5/templates/Template1/bds/BD1",
    "name": "BD1",
    "event": "created",
    "container": {
        "type": "template",
        "key": "schemas/59ef87743800008e06881fe5/templates/Template1",
        "name": "Template 1"
    },
    "user": {
        "userId": "0000ffff00000000000000020",
        "username": "admin",
        "firstName": "Admin",
        "lastName": "User",
        "domainId": "0000ffff0000000000000090",
        "domainName": "Local"
    },
    "timestamp": "2017-10-24T18:33:24.584"
},
{
    "id": "59ef87743a000022069baf72",
    "type": "attribute",
    "key": "/schemas/59ef87743800008e06881fe5/templates/Template1/bds/BD1/displayName",
    "event": "created",
    "container": {
        "type": "bd",
        "key": "/schemas/59ef87743800008e06881fe5/templates/Template1/bds/BD1",
        "name": "BD1"
    },
    "newValue": "BD1",
    "user": {
        "userId": "0000ffff00000000000000020",
        "username": "admin",
        "firstName": "Admin",
        "lastName": "User",

```

```

        "domainId": "0000ffff000000000000000090",
        "domainName": "Local"
    },
    "timestamp": "2017-10-24T18:33:24.584"
},
{
    "id": "59ef87743a000022069baf73",
    "type": "attribute",
    "key": "/schemas/59ef87743800008e06881fe5/templates/Template1/bds/BD1/12UnknownUnicast",
    "event": "created",
    "container": {
        "type": "bd",
        "key": "/schemas/59ef87743800008e06881fe5/templates/Template1/bds/BD1",
        "name": "BD1"
    },
    "newValue": "proxy",
    "user": {
        "userId": "0000ffff000000000000000020",
        "username": "admin",
        "firstName": "Admin",
        "lastName": "User",
        "domainId": "0000ffff000000000000000090",
        "domainName": "Local"
    },
    "timestamp": "2017-10-24T18:33:24.584"
},
{
    "id": "59ef87743a000022069baf74",
    "type": "attribute",
    "key":
"/schemas/59ef87743800008e06881fe5/templates/Template1/bds/BD1/intersiteBumTrafficAllow",
    "event": "created",
    "container": {
        "type": "bd",
        "key": "/schemas/59ef87743800008e06881fe5/templates/Template1/bds/BD1",
        "name": "BD1"
    },
    "newValue": "true",
    "user": {
        "userId": "0000ffff000000000000000020",
        "username": "admin",
        "firstName": "Admin",
        "lastName": "User",
        "domainId": "0000ffff000000000000000090",
        "domainName": "Local"
    },
    "timestamp": "2017-10-24T18:33:24.584"
},
{
    "id": "59ef87743a000022069baf75",
    "type": "attribute",
    "key":
"/schemas/59ef87743800008e06881fe5/templates/Template1/bds/BD1/optimizeWanBandwidth",
    "event": "created",
    "container": {
        "type": "bd",
        "key": "/schemas/59ef87743800008e06881fe5/templates/Template1/bds/BD1",
        "name": "BD1"
    },
    "newValue": "true",
    "user": {
        "userId": "0000ffff000000000000000020",
        "username": "admin",

```

```

        "firstName": "Admin",
        "lastName": "User",
        "domainId": "0000ffff0000000000000090",
        "domainName": "Local"
    },
    "timestamp": "2017-10-24T18:33:24.584"
},
{
    "id": "59ef87743a000022069baf76",
    "type": "attribute",
    "key": "/schemas/59ef87743800008e06881fe5/templates/Template1/bds/BD1/12Stretch",
    "event": "created",
    "container": {
        "type": "bd",
        "key": "/schemas/59ef87743800008e06881fe5/templates/Template1/bds/BD1",
        "name": "BD1"
    },
    "newValue": "true",
    "user": {
        "userId": "0000ffff0000000000000020",
        "username": "admin",
        "firstName": "Admin",
        "lastName": "User",
        "domainId": "0000ffff0000000000000090",
        "domainName": "Local"
    },
    "timestamp": "2017-10-24T18:33:24.584"
},
{
    "id": "59ef87743a000022069baf77",
    "type": "attribute",
    "key": "/schemas/59ef87743800008e06881fe5/templates/Template1/bds/BD1/vrfRef",
    "event": "associated",
    "container": {
        "type": "bd",
        "key": "/schemas/59ef87743800008e06881fe5/templates/Template1/bds/BD1",
        "name": "BD1"
    },
    "newValue": "/schemas/59ef87743800008e06881fe5/templates/Template1/vrfs/VRf1",
    "user": {
        "userId": "0000ffff0000000000000020",
        "username": "admin",
        "firstName": "Admin",
        "lastName": "User",
        "domainId": "0000ffff0000000000000090",
        "domainName": "Local"
    },
    "timestamp": "2017-10-24T18:33:24.584"
},
{
    "id": "59ef87743a000022069baf78",
    "type": "attribute",
    "key": "/schemas/59ef87743800008e06881fe5/templates/Template1/bds/BD1/subnets",
    "event": "added",
    "container": {
        "type": "bd",
        "key": "/schemas/59ef87743800008e06881fe5/templates/Template1/bds/BD1",
        "name": "BD1"
    },
    "newValue": "10.0.0.1/24",
    "user": {
        "userId": "0000ffff0000000000000020",
        "username": "admin",
        "firstName": "Admin",

```



```

        "lastName": "User",
        "domainId": "0000ffff0000000000000090",
        "domainName": "Local"
    },
    "timestamp": "2017-10-24T18:33:24.584"
}
]
}

```

Find Users Associated with Schema Audit Records

The following example shows how to find all users associated with Schema audit records.

GET:

```
GET https://<host>:<port>/api/v1/schema/audit-records/users
```

```
GET https://<host>:<port>/api/v1/schema/:id/audit-records/users
```

Request:

<empty>

Response:

```

HTTP/1.1 200
{
  "auditUsers": [
    {
      "userId": "0000ffff0000000000000020",
      "username": "admin",
      "firstName": "Admin",
      "lastName": "User",
      "domainId": "0000ffff0000000000000090",
      "domainName": "Local"
    },
    {
      "userId": "59ed6de03d00003800175a2e",
      "username": "bob",
      "firstName": "Bob",
      "lastName": "Smith",
      "domainId": "0000ffff0000000000000090",
      "domainName": "Local"
    }
  ]
}

```

List or Filter Audit Records for Fabric-Connectivity

The following example shows how to list or filter audit records for fabric-connectivity. This example provides a high-level audit view of updates for fabric-connectivity.

**Note**

- Default limit and offset values apply.
- Multiple additional filters (such as username, event, timestamp, date range) may be applied. Filters may also be combined.

GET:

```
GET https://<host>:<port>/api/v1/sites/fabric-connectivity/audit-records
```

```
GET https://<host>:<port>/api/v1/sites/fabric-connectivity/audit-records?offset=0&limit=3
```

```
GET
```

```
https://<host>:<port>/api/v1/sites/fabric-connectivity/audit-records?username=admin&event=created,updated
```

Request:

Find Users Associated with Fabric-Connectivity Audit Records

The following example shows how to find all users associated with Fabric-Connectivity audit records.

GET:

```
GET /api/v1/sites/fabric-connectivity/audit-records/users
```

APIs for Backups

About Backup

Starting with Cisco ACI Multi-Site, release 1.1(1), the APIs for backup are supported. Therefore, the backup examples are applicable starting with release 1.1(1) only.

Create a New Backup

The following is an example of how to create a new backup.

POST:

```
POST https://<host>:<port>/api/v1/backups
```

Request:

```
{
  "name": "testFile3",
  "description": "this is optional key value"
```

Response:

```
HTTP/1.1 201
Location: /api/v1/backups/20171116221404
{
  "id": "20171116221404",
  "sizeKB": 18.44,
  "timestamp": "2017-11-16T22:14:04",
  "metadata": {
    "description": "test-description",
    "name": "Check_20171116221404"
  }
}
```

List Backups

The following is an example of how to list backups.

GET:

```
GET https://<host>:<port>/api/v1/backups
```

Request:

<empty>

Response:

```
HTTP/1.1 200
{
  "backups": [
    {
      "id": "20171116221404",
      "sizeKB": 18.44,
      "timestamp": "2017-11-16T22:14:04",
      "metadata": {
        "description": "test-description",
        "name": "Check_20171116221404"
      }
    },
    {
      "id": "20171116221721",
      "sizeKB": 18.55,
      "timestamp": "2017-11-16T22:17:21",
      "metadata": {
        "description": "test-description",
        "name": "Check_20171116221721"
      }
    },
    {
      "id": "20171116221724",
      "sizeKB": 19.11,
      "timestamp": "2017-11-16T22:17:24",
      "metadata": {
        "description": "test-description",
        "name": "Check_20171116221724"
      }
    },
    {
      "id": "20171116224425",
      "sizeKB": 18.99,
```

```

        "timestamp": "2017-11-16T22:44:25",
        "metadata": {
            "description": "test-description",
            "name": "Check_20171116224425"
        }
    },
    {
        "id": "20171127071624",
        "sizeKB": 24.72,
        "timestamp": "2017-11-27T07:16:24",
        "metadata": {
            "name": "testFile3_20171127071624"
        }
    },
    {
        "id": "20171127071655",
        "sizeKB": 28.08,
        "timestamp": "2017-11-27T07:16:55",
        "metadata": {
            "description": "this is optional key value",
            "name": "testFile3_20171127071655"
        }
    }
]
}

```

The following is an example of how to list a backup.

GET:

GET https://<host>:<port>/api/v1/backups/:id

Request:

<empty>

Response:

```

HTTP/1.1 200
{
    "id": "20170926231729",
    "sizeKB": 1.91,
    "dateTime": "2017-09-26T23:17:29",
    "metadata": {
        "description": "Right before zip review"
    }
}

```

Download a Backup

The following is an example of how to download a backup.

GET:

GET https://<host>:<port>/api/v1/backups/:id/download

Request:

```
<empty>
```

Response:

```
HTTP/1.1 200  
Binary output as it is a file
```

Upload a Backup

The following is an example of how to upload a backup.

GET:

```
GET https://<host>:<port>/api/v1/backups/:id/upload
```

Request:

```
form-data  
key: name  
value: valid tar.gz filename
```

Response:

```
HTTP/1.1 200  
{  
  "id": "20171127071655",  
  "sizeKB": 28.08,  
  "timestamp": "2017-11-27T07:16:55",  
  "metadata": {  
    "description": "this is optional key value",  
    "name": "testFile3_20171127071655"  
  }  
}
```

