

## **LAB GUIDE**

# **Voice-driven network operations for an integrated and programmable SDN**

## **DEVWKS-1908**

### **Speakers:**

**Arvind Chari**

**Ovesnel Mas Lara**

# Table of Contents

Table of Contents.....	2
Learning Objectives.....	4
Requirements.....	4
Components.....	4
Network Diagram.....	4
Lab Access .....	5
Prepare Lab environment .....	7
Step 1 Setup Postman environment .....	7
Step 2 Setup Postman collection .....	7
Step 3 Test connection to APIC.....	8
Step 4 Test connection to DNAC.....	8
Step 5 Open SDN controllers UI.....	9
Step 6 Provide credentials .....	10
LAB1: Network Health Check .....	11
Task 1: Check the status of the Software Defined Network .....	11
Step 1 Check network health on controllers UI .....	11
Step 2 Get the SDA network health via API .....	11
Step 3 Get the DC fabric health through API .....	12
LAB2: Macro Segmentation .....	13
Task 1: Create a Macro-segment in Cisco DNAC.....	13
Step 1 Check Virtual Networks on DNAC UI.....	13
Step 2 Create a new Virtual Network via DNAC API .....	13
Step 3 Check DNAC task status .....	14
Step 4 Confirm Virtual Network creation on DNAC UI .....	15
Task 2: Create a Macro-segment in Cisco ACI.....	15
Step 1 Check ACI tenants on APIC UI .....	15
Step 2 Create a new Tenant and VRF via APIC API .....	15
Step 3 Confirm tenant and VRF creation on APIC UI .....	16
Task 3: Delete a Macro-segment from Cisco DNAC.....	16
Step 1 Delete a Virtual Network from DNAC via API .....	17
Step 2 Confirm that the VM was deleted on DNAC UI.....	17

Task 4: Delete a Macro-segment from Cisco ACI.....	17
Step 1 Delete Tenant from ACI via API.....	17
Step 2 Confirm tenant deletion on APIC UI .....	18
LAB3: Micro Segmentation .....	19
Task 1: Enabling Micro-segmentation in Cisco ACI.....	19
Step 1 Open the IoT tenant on APIC GUI .....	20
Step 2 Verify the Application Profiles for IoT tenant .....	20
Step 3 Verify the Bridge Domains for IoT tenant.....	21
Step 4 Verify the Contracts for IoT tenant.....	21
Step 5 Create a new AP, EPG, BD and Contract for tenant IoT via APIC API.....	22
Step 6 Visualize the application topology on APIC UI.....	22
Step 7 Establish a contract between EPGs via APIC API .....	22
Step 8 Visualize the application topology on APIC UI.....	23
Appendix 1 Extra API calls.....	24
Check Virtual Networks in DNAC via API .....	24
Check ACI tenants via API .....	24
Delete AP, EPG, BD and Contract.....	24
Appendix 2 Alexa Robot Skill Intents .....	26

# Learning Objectives

Upon completion of this lab, you will be able to use APIs to orchestrate multi-domain:

- Health checks
- Macro-segmentation
- Micro-segmentation

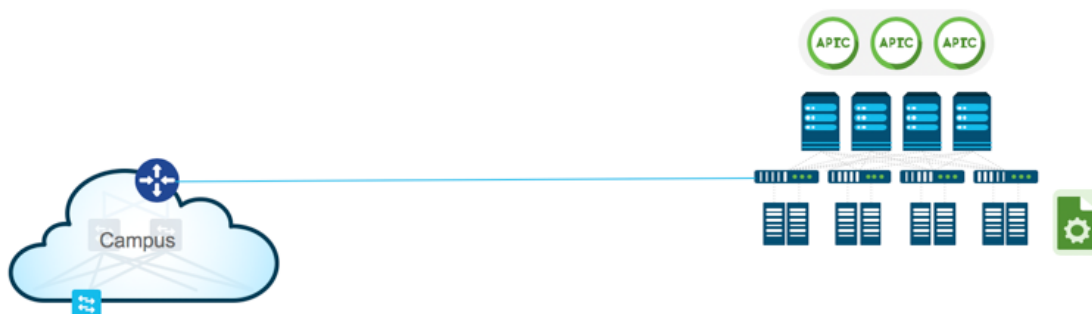
## Requirements

- Laptop with Google Chrome browser.
- Cisco AnyConnect Client.
- Postman
- A Postman collection and environment created for DEWKS-1908
- Lab Information Sheet with credentials, IPs and URLs of Lab components

## Components

- Cisco DNA Center 1.3.0
- Cisco ACI 4.0(3d)
- VMware vCenter 6.0.0

## Network Diagram

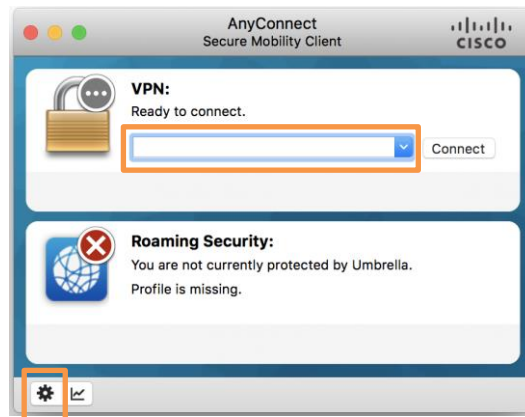


## Lab Access

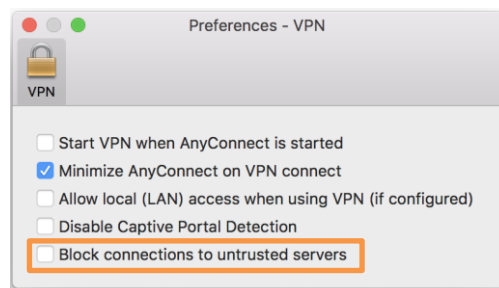
In order to access this Lab, you need to establish a VPN connection using the Cisco AnyConnect Client which is already installed on your laptop. Please, follow these steps to connect to the VPN:

*Note: The required VPN information is provided in the **Lab Information Sheet***

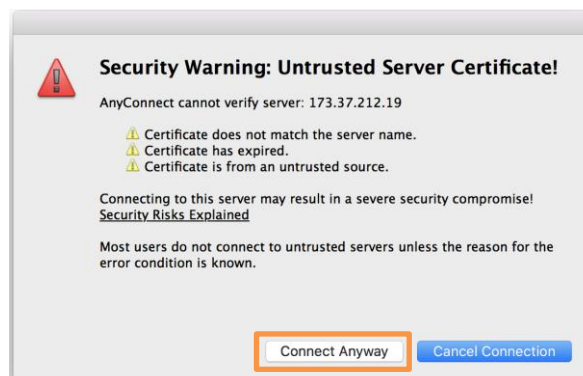
1. Open Cisco AnyConnect on your laptop.
2. Enter the IP address of the VPN server, click on the small **gear icon** located at the bottom left corner of the window:



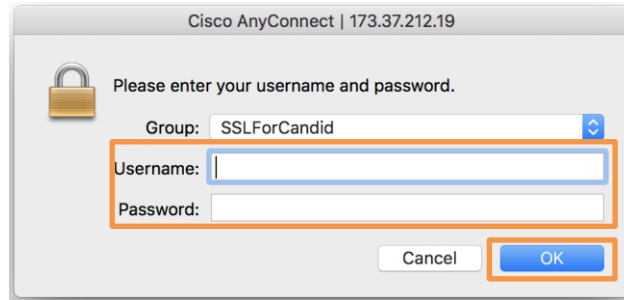
3. Un-check the **Block connections to untrusted servers** option:



4. Close Preferences window and click **Connect**.
5. If the following security warning appears, click **Connect Anyway**:



6. Enter your username and password and click **OK**:



The image shows a Cisco AnyConnect login window. The title bar reads "Cisco AnyConnect | 173.37.212.19". Inside the window, there is a padlock icon and the text "Please enter your username and password." Below this, there is a "Group:" dropdown menu with "SSLForCandid" selected. Underneath are two input fields: "Username:" and "Password:". At the bottom right, there are two buttons: "Cancel" and "OK". An orange rectangular box highlights the "Group:", "Username:", and "Password:" fields. Another orange rectangular box highlights the "OK" button.

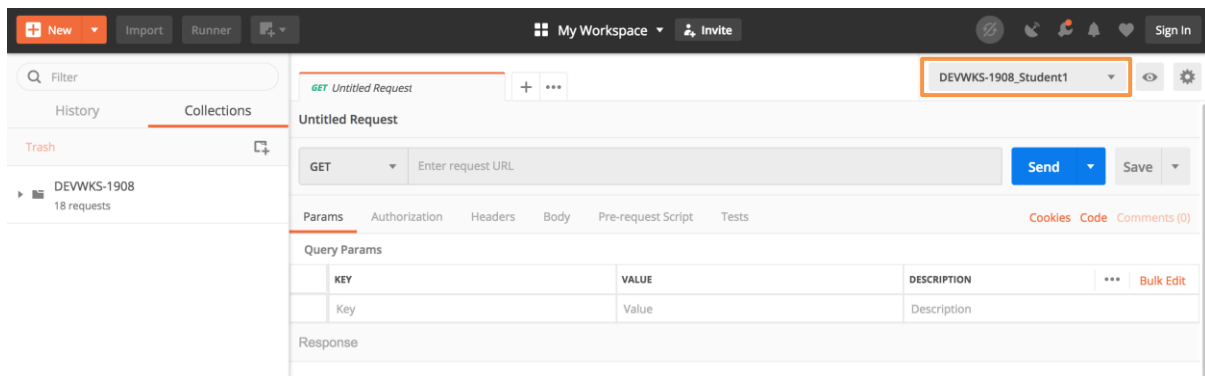
At this point you should be connected the Lab VPN.

# Prepare Lab environment

In this task you will setup Postman, login into SDN controllers UI, and test the connection to them via APIs.

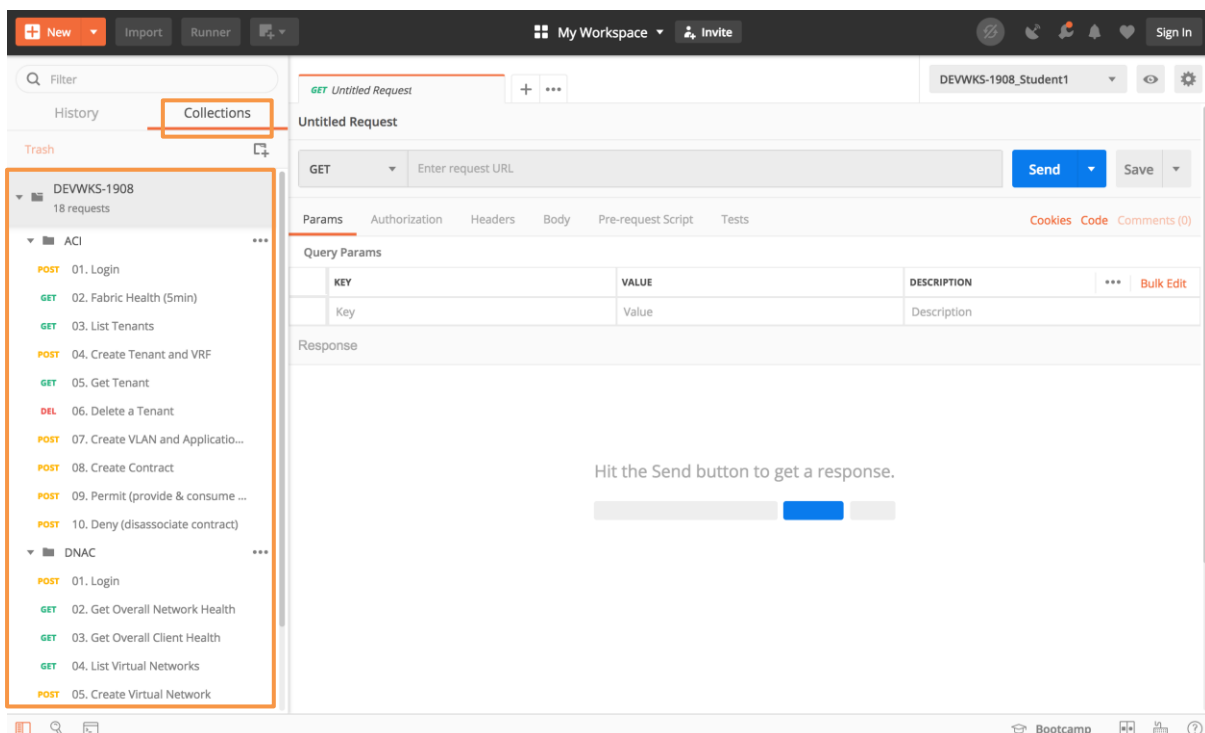
## Step 1 Setup Postman environment

Open Postman and select the **Environment** called “DEVWKS-1908\_StudentX”, where X is your assigned student number. This environment contains a set of key-value pairs that will be used in API requests and will save you time running the Lab.



## Step 2 Setup Postman collection

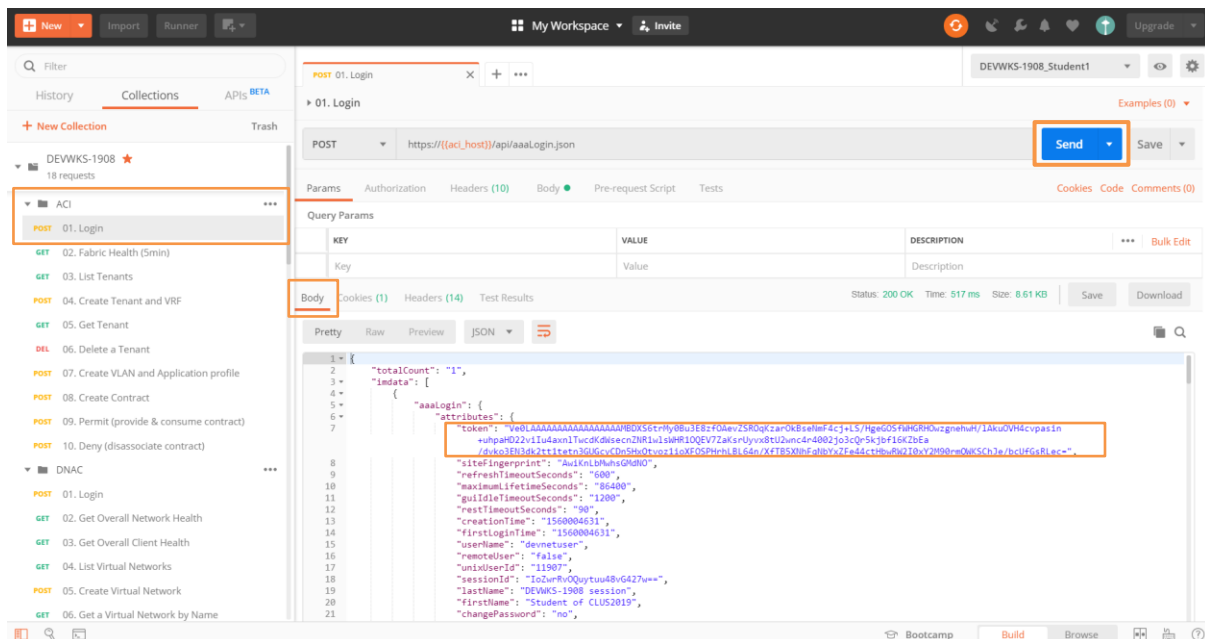
Click the **Collections** tab in the left panel and open the collection called “DEVWKS-1908”:



Inside each folder, the requests are sequential. **Don't** run them yet, we will run them one by one during the Lab and check on the UI for the controllers to see the effect of the requests.

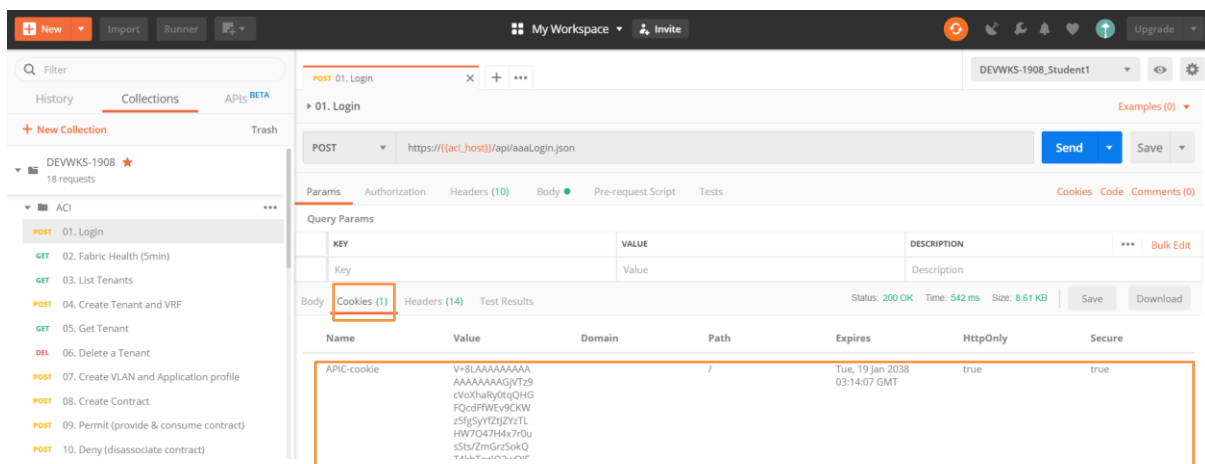
## Step 3 Test connection to APIC

Let's check the connection to APIC controller. Within the **ACI** folder, open the POST request called "**01. Login**" and click **Send**:



You just sent a POST request to authenticate with APIC. The body of the response should include an authentication **token** as shown above.

The response has also a **cookie** that Postman will automatically use to authenticate subsequent requests.

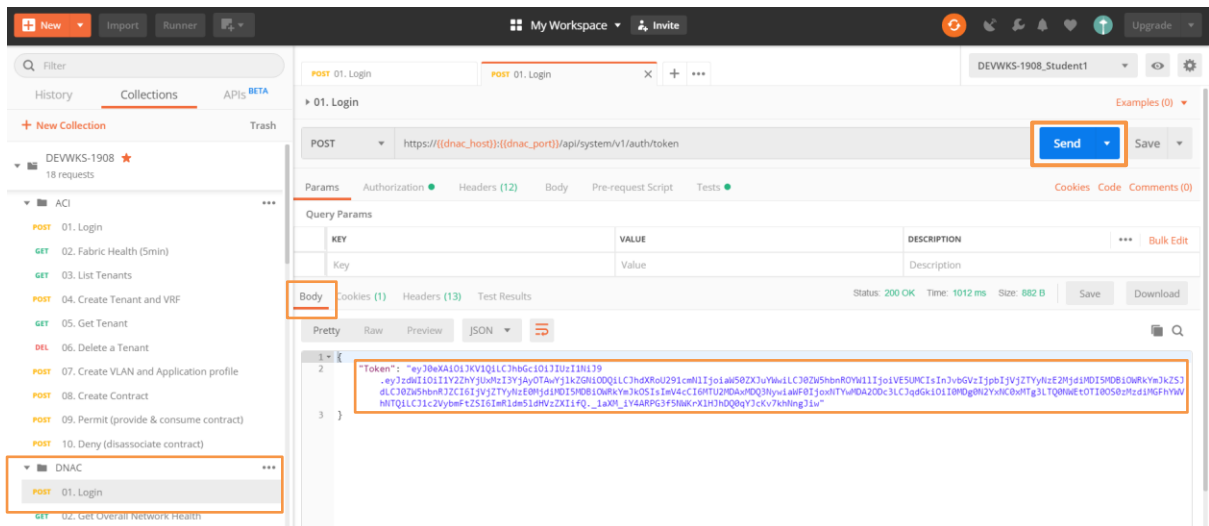


**NOTE:** If this request fails you may need to accept the certificate from the controller in your browser <http://blog.getpostman.com/2014/01/28/using-self-signed-certificates-with-postman/>

## Step 4 Test connection to DNAC

Let's check the connection to DNAC controller. Within the **DNAC** folder, open the POST request called "**01. Login**" and click **Send**:





The body of the response should include an authentication **token** as shown above. This token is stored in an environment variable and used in subsequent requests.

## Step 5 Open SDN controllers UI

Open Google Chrome browser and navigate to the APIC and DNAC URLs provided in the Lab Information Sheet.

If the following warning appears in your browser, click **Advanced**:



### Your connection is not private

Attackers might be trying to steal your information from **10.201.146.32** (for example, passwords, messages, or credit cards). [Learn more](#)

NET::ERR\_CERT\_AUTHORITY\_INVALID

☐ Help improve Safe Browsing by sending some [system information and page content](#) to Google. [Privacy policy](#)

Advanced

Back to safety

and then click on **Proceed to <IP> (unsafe)** link:

Hide advanced

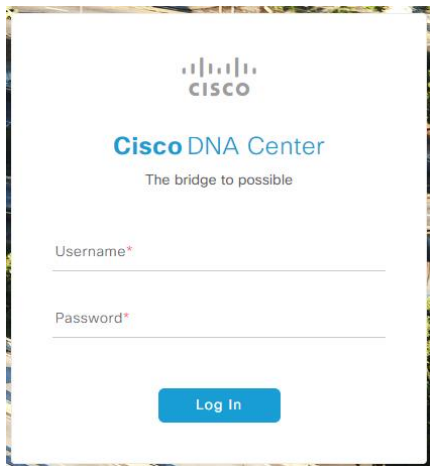
Back to safety

This server could not prove that it is **10.201.146.32**; its security certificate is not trusted by your computer's operating system. This may be caused by a misconfiguration or an attacker intercepting your connection.

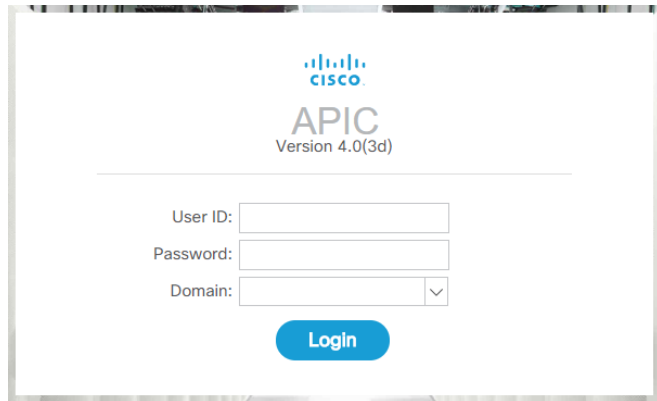
Proceed to 10.201.146.32 (unsafe)

## Step 6 Provide credentials

Enter your credentials in the login screens (you can leave APIC Domain in blank) and click **Login**:



The Cisco DNA Center login screen features the Cisco logo at the top, followed by the text "Cisco DNA Center" and the tagline "The bridge to possible". Below this, there are two input fields: "Username\*" and "Password\*", each with a red asterisk indicating a required field. A blue "Log In" button is positioned at the bottom center of the form.



The Cisco APIC login screen displays the Cisco logo, the text "APIC", and "Version 4.0(3d)". The login form includes three input fields: "User ID:", "Password:", and "Domain:". The "Domain:" field is a dropdown menu with a downward arrow icon. A blue "Login" button is located at the bottom center of the form.

You should the now be in the home page of Cisco APIC and Cisco DNAC controllers.

# LAB1: Network Health Check

## Task 1: Check the status of the Software Defined Network

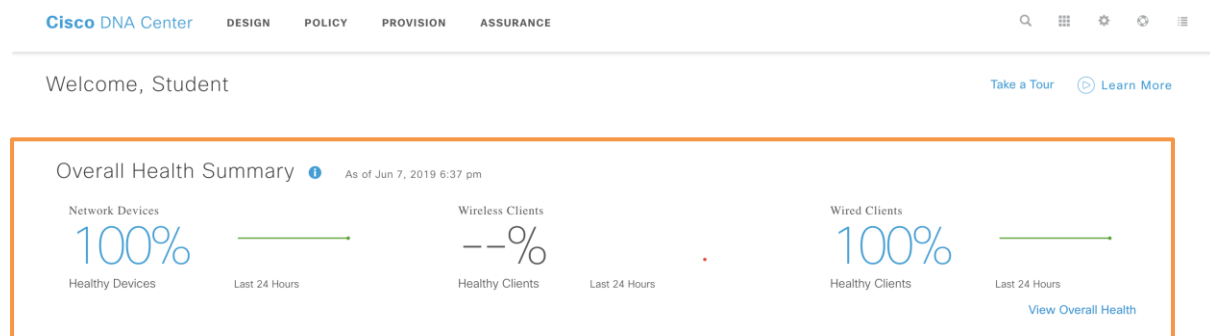
In this task you will check the health of the Software-defined Access (SDA) network and Software-defined DC Fabric in two ways:

1. Using the UI of Cisco APIC and Cisco DNAC controllers
2. Making RESTful API requests from Postman

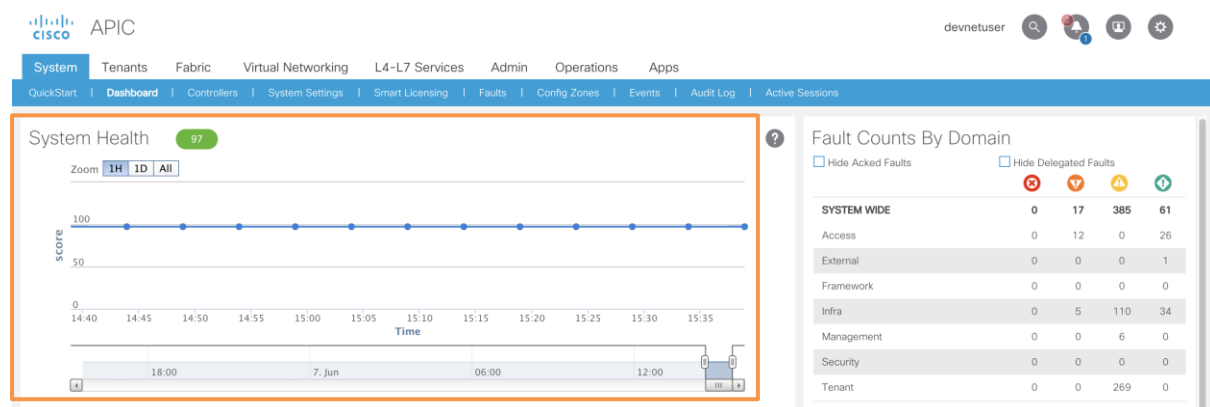
This was done by the Alexa Robot skill when we asked her **“What is the current status of the network?”**.

### Step 1 Check network health on controllers UI

Switch to the Cisco DNAC and Cisco APIC home pages you opened in Google Chrome in the previous task, check the Overall Health of DNAC:



And the System Health of ACI:



### Step 2 Get the SDA network health via API

Open Postman, within the folder **DNAC**, open the GET request called **“Get Overall Network Health”** and click **Send**:

The screenshot shows the Postman interface with a collection named 'DEVWKS-1908'. The request '02. Get Overall Network Health' is selected. The URL is `https://(dnac_host):(dnac_port)/dna/intent/api/v1/network-health/timestamp=((dnac_timestamp))`. The response is a JSON object with the following structure:

```

{
  "version": "1.0",
  "response": [ {
    "time": "2019-06-08T00:15:00.000+0000",
    "healthScore": 100,
    "totalCount": 2,
    "goodCount": 2,
    "unmonCount": 0,
    "failCount": 0,
    "badCount": 0,
    "entity": null,
    "timeInMillis": 1559952900000
  } ],
  "measuredBy": "global",
  "latestMeasuredByEntity": null,
  "latestHealthScore": 100,
  "monitoredDevices": 2,
  "monitoredHealthyDevices": 2,
  "monitoredUnhealthyDevices": 0,
  "unmonitoredDevices": 0,
  "healthDistribution": [ {

```

As shown above, the response includes the same overall network health value you just saw in DNAC UI.

Repeat the same with the request called **“Get Overall Client Health”**.

### Step 3 Get the DC fabric health through API

Open the folder **ACI**, and send the GET request called **“Fabric Health (5min)”**:

The screenshot shows the Postman interface with a collection named 'DEVWKS-1908'. The request '02. Fabric Health (5min)' is selected. The URL is `https://(aci_host)/api/node/mo/topology/HDfabricOverallHealth5min-0.json`. The response is a JSON object with the following structure:

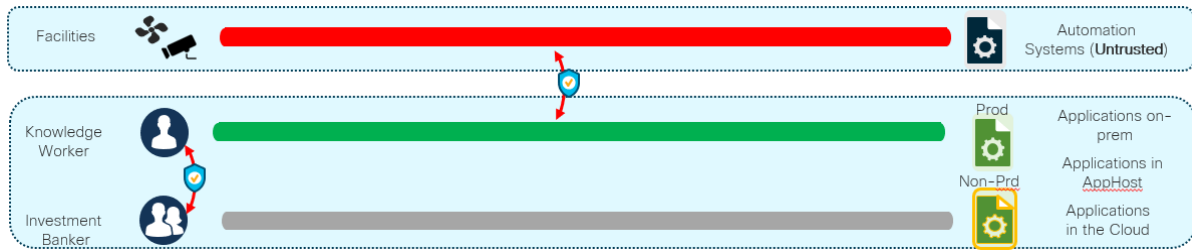
```

{
  "totalCount": "1",
  "imdata": [ {
    "fabricOverallHealthHist5min": {
      "attributes": {
        "childAction": "",
        "cnt": "30",
        "dn": "topology/HDfabricOverallHealth5min-0",
        "healthAvg": "97",
        "healthMax": "97",
        "healthMin": "97",
        "healthSpec": "0",
        "healthThr": "0",
        "healthTr": "0",
        "index": "0",
        "lastCollOffset": "300",
        "repIntvEnd": "2019-06-08T08:38:58.529-07:00",
        "repIntvStart": "2019-06-08T08:33:58.420-07:00",
        "status": ""
      }
    }
  } ]
}

```

As shown above, the response includes the fabric health values you just saw in the UI.

## LAB2: Macro Segmentation



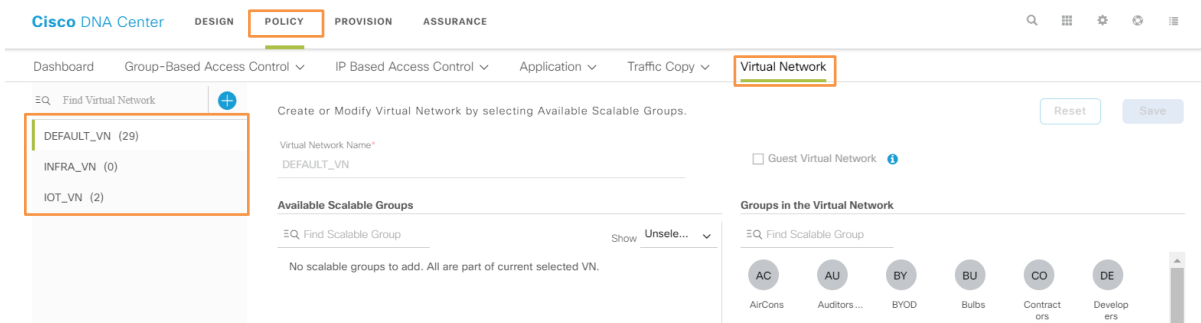
### Task 1: Create a Macro-segment in Cisco DNAC

In this task and the next one, you will isolate Untrusted IoT devices by creating a Virtual Network in Cisco DNAC and a Tenant in Cisco ACI. This was done by the Alexa Robot skill when we asked her “**Create a new segment called Untrusted**”.

You will perform these tasks by making API requests from Postman. The results will be verified on the UI of each controller.

#### Step 1 Check Virtual Networks on DNAC UI

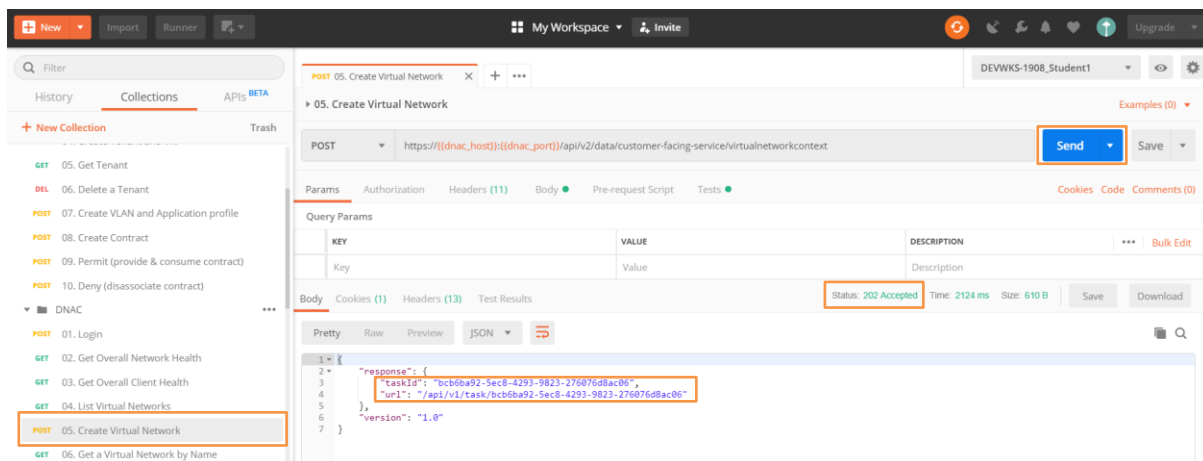
Open **DNAC UI**, go to **Policy > Virtual Network** and check the existing Virtual Networks:



In the following steps we will check, through the DNAC API, the same information and will create a new Virtual Network.

#### Step 2 Create a new Virtual Network via DNAC API

Let's now create a new Virtual Network (VN) called “**Sx\_Untrusted**” in DNAC (x = Student number). You don't need to set the VN name, there is already a variable with this value in your Postman environment. Send to **DNAC** the POST request called “**Create Virtual Network**”:

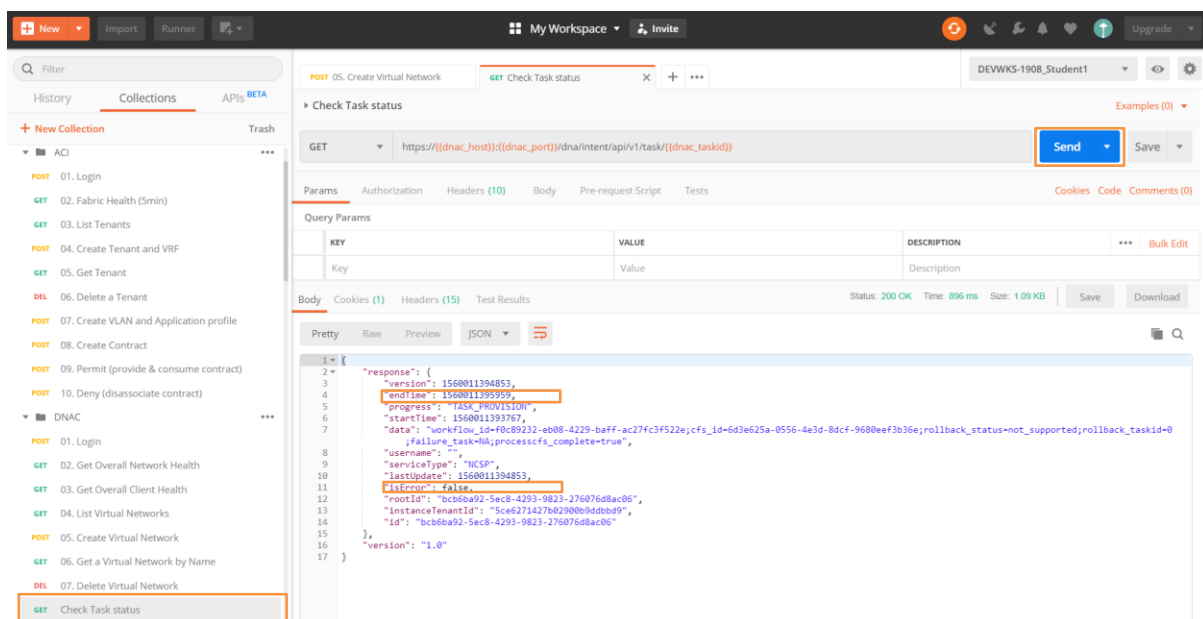


Most of the calls on DNAC controller are **asynchronous** (PUT/POST/DELETE). Accepted asynchronous requests will return the HTTP status code **202 Accepted**, and as shown above, a **taskId** is provided in the response.

You need to poll the task status to find out if the creation was successful. The next step shows how to do this.

### Step 3 Check DNAC task status

To check the status of the previous task, send to **DNAC** the request called “**Check Task Status**”:



The presence of **"endTime"** in the response means the task has finished. **"isError"** being false means the task was successful.

Notice there is a **{{dnac\_taskid}}** in the URL being called. Anything inside the **{{}}** is a variable. In this case **dnac\_taskid**. This was set behind the scenes in the previous API call.

Once the task finishes, you should be able to see the newly created Virtual Network **“Sx\_Untrusted”** by running from Postman GET requests **“Get a Virtual Network by Name”** or **“List Virtual Networks”** provided in the DEWKS-1908 > DNAC folder.

#### Step 4 Confirm Virtual Network creation on DNAC UI

Open **DNAC** UI, refresh the page **Policy > Virtual Network** and confirm that the Virtual Network was created.

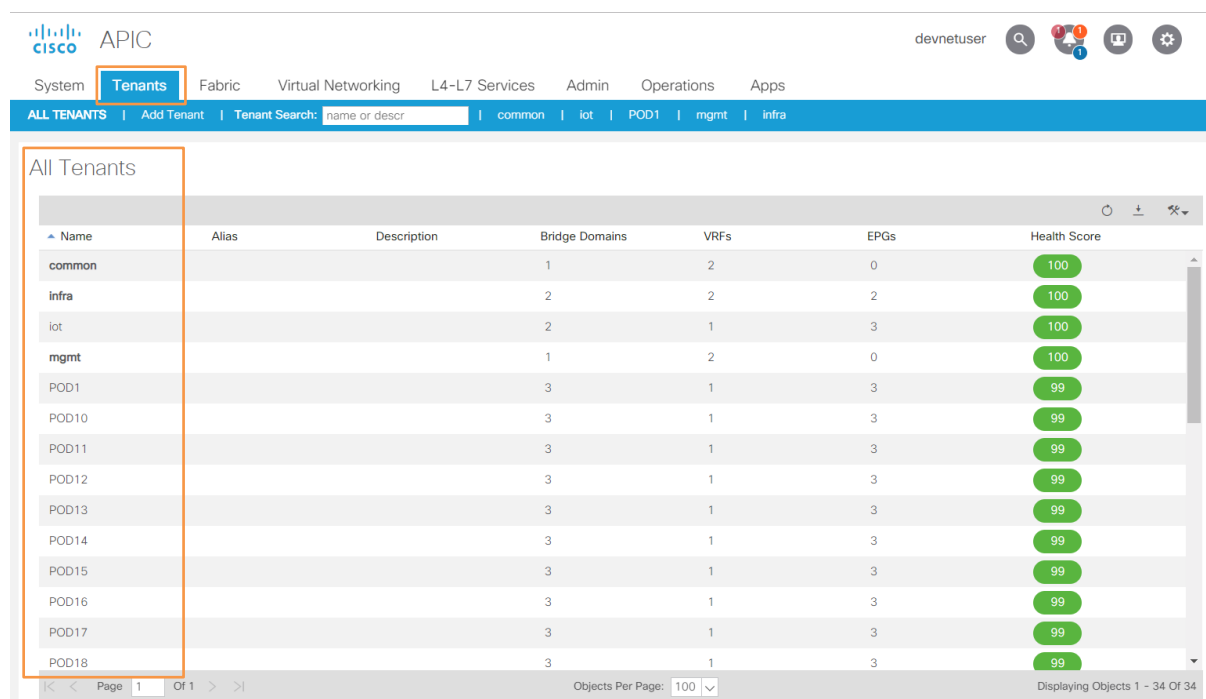
## Task 2: Create a Macro-segment in Cisco ACI

In this task, you will isolate Untrusted devices by creating a Tenant in Cisco ACI. This was done by the Alexa Robot skill when we asked **“Create a new segment called Untrusted”**.

You will perform these tasks making API requests from Postman. The results will be verified on the UI of APIC.

#### Step 1 Check ACI tenants on APIC UI

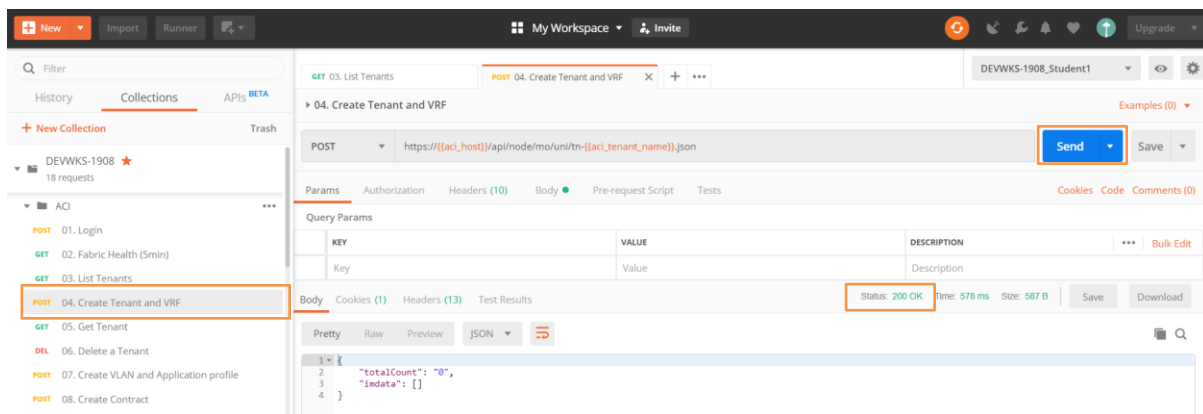
Open **APIC** UI, click on the **Tenants** tab and check the existing Tenants (you may need to scroll down or change the number of Objects Per Page to see the full list):



Name	Alias	Description	Bridge Domains	VRFs	EPGs	Health Score
common			1	2	0	100
infra			2	2	2	100
iot			2	1	3	100
mgmt			1	2	0	100
POD1			3	1	3	99
POD10			3	1	3	99
POD11			3	1	3	99
POD12			3	1	3	99
POD13			3	1	3	99
POD14			3	1	3	99
POD15			3	1	3	99
POD16			3	1	3	99
POD17			3	1	3	99
POD18			3	1	3	99

#### Step 2 Create a new Tenant and VRF via APIC API

Let's now create a new Tenant called **“Sx\_Untrusted”** in ACI (x = Student number) and within this tenant a VRF called **“Sx\_Untrusted-VRF”**. You don't need to set the tenant/VRF name, there are already variables with these values in your Postman environment. Send to **APIC** the POST request called **“Create Tenant and VRF”**:



If the HTTP response status is **200 OK**, you should now be able to see the new tenant and VRF by running from Postman the GET request called **“Get Tenant”**.

### Step 3 Confirm tenant and VRF creation on APIC UI

Open **APIC** UI, click on the **Tenants** tab and confirm that the new tenant and VRF were created:

The screenshot shows the Cisco APIC UI with the 'Tenants' tab selected. The table below lists the tenants and their associated VRFs.

Name	Alias	Description	Bridge Domains	VRFs	EPGs	Health Score
POD25			3	1	3	99
POD26			3	1	3	99
POD27			3	1	3	99
POD28			3	1	3	99
POD29			3	1	3	99
POD3			3	1	3	99
POD30			3	1	3	99
POD4			3	1	3	99
POD5			3	1	3	99
POD6			3	1	3	99
POD7			3	1	3	99
POD8			3	1	3	99
POD9			3	1	3	99
S1_Untrusted			0	1	0	100

### Task 3: Delete a Macro-segment from Cisco DNAC

In this task you will delete the previously created Macro-Segment. This was done by the Alexa Robot skill when we asked her **“Delete the segment called Untrusted”**.

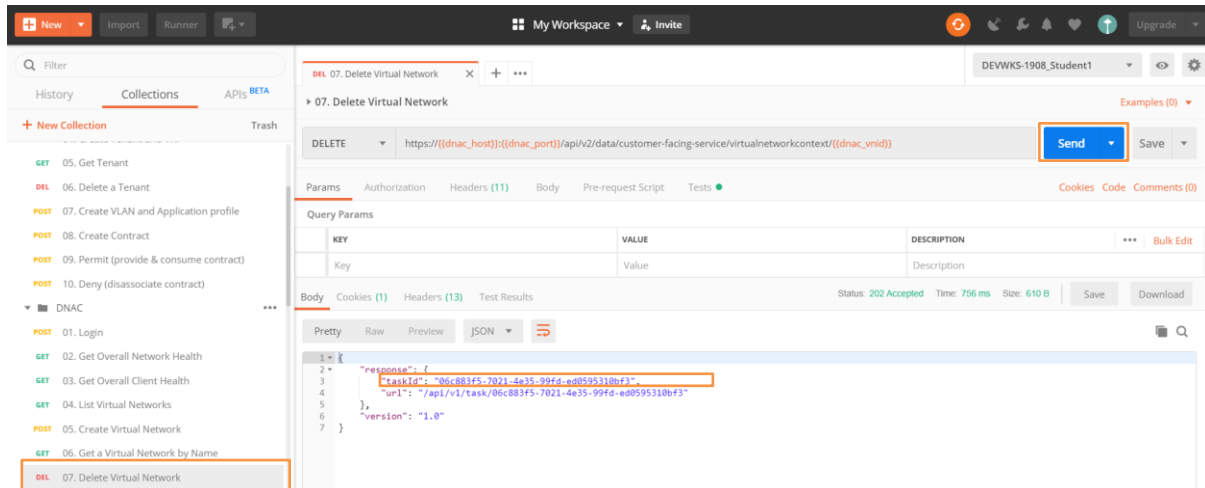
You will perform this task making API requests from Postman. The results will be verified on the DNAC UI.



## Step 1 Delete a Virtual Network from DNAC via API

Let's delete the previously created Virtual Network (VN). Open Postman and send to **DNAC** the GET request called "**Get Virtual Network by Name**". Behind the scenes, this will take the VN id from the response and assign it to an environment variable that will be used in the next API call.

Now send to **DNAC** the DELETE request called "**Delete Virtual Network**".



This is an **asynchronous** call. Check that the HTTP status code is **202 Accepted** as shown above. The **taskId** provided in the response will be used in the next step to poll the task status to find out if the deletion was successful.

## Step 2 Confirm that the VM was deleted on DNAC UI

Open **DNAC** UI, refresh the page **Policy > Virtual Network** and confirm that the Virtual Network was deleted.

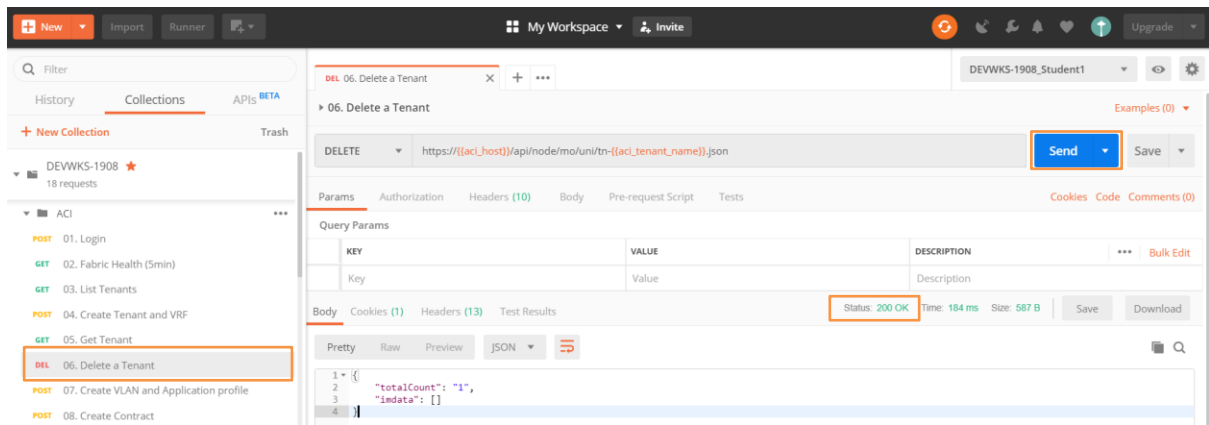
## Task 4: Delete a Macro-segment from Cisco ACI

In this task you will delete the previously created Macro-Segment. This was done by the Alexa Robot skill when we asked her "**Delete the segment called Untrusted**".

You will perform this task making API requests from Postman. The results will be verified on the APIC UI.

## Step 1 Delete Tenant from ACI via API

Let's delete the previously created Tenant. Open Postman and send to **APIC** the DELETE request called "**Delete Tenant**".



## Step 2 Confirm tenant deletion on APIC UI

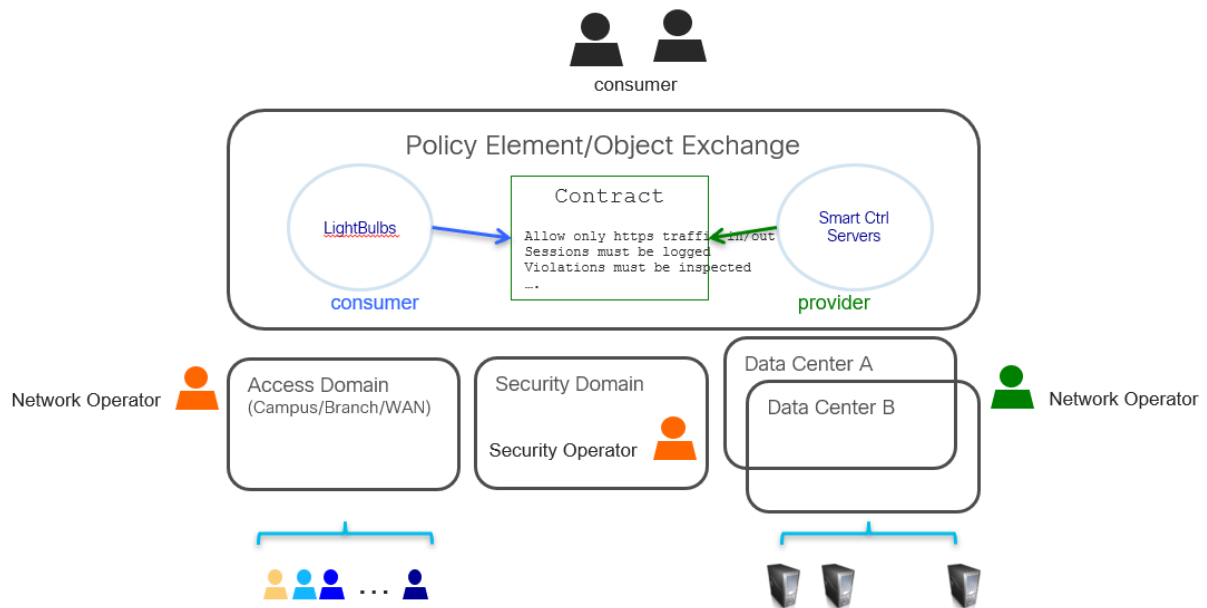
Open **APIC UI**, click on the **Tenants** tab and confirm that the tenant was deleted.

## LAB3: Micro Segmentation

### Task 1: Enabling Micro-segmentation in Cisco ACI

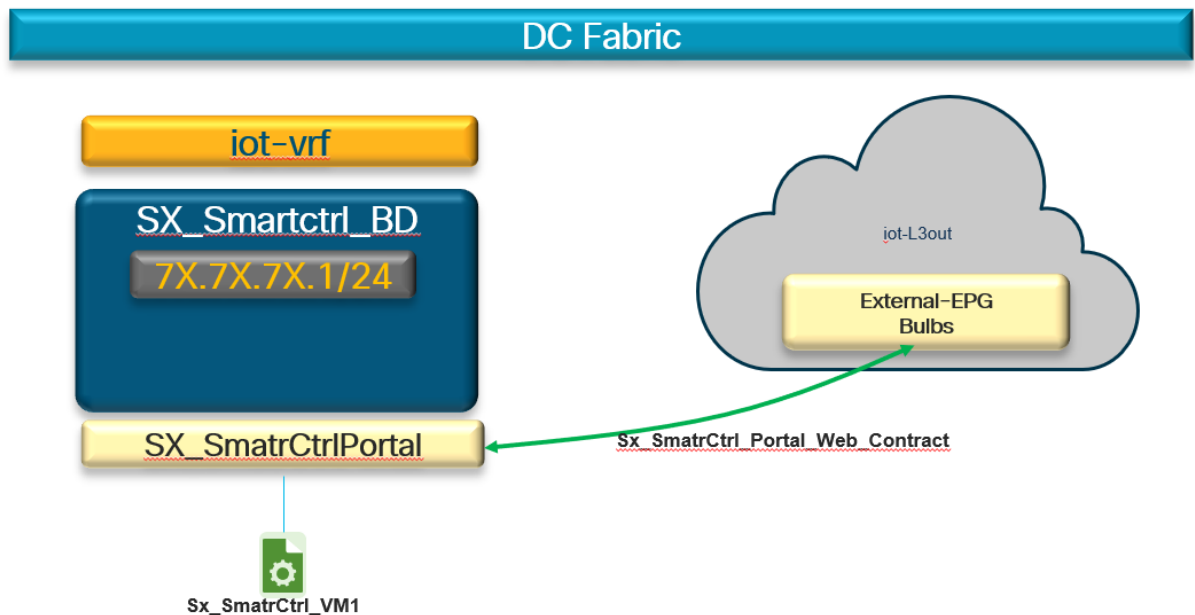
In this task we will create a series of ACI logical constructs in order to segment and secure the IoT application and then allow specific traffic from Smart Controller servers to light bulbs.

This was done by the Alexa Robot skill when we asked her “**Allow light bulbs to talk to smart controller servers**”.



The following logical constructs will be created within the IoT tenant:

- One **Application Profile** (AP) within the IoT tenant. An AP is a collection of different EPGs and policies needed to communicate between them.
- One application **Endpoint Group** (EPG) within the AP for the Web servers. An EPG is a collection of endpoints requiring the same policy treatment. i.e. app tiers, or services.
- One **Bridge Domain** (BD) and one **Subnet**. BDs are used to define a L2 boundary (Flood Domain) and impose additional constraints (such as no broadcast) within that L2 boundary. Not a VLAN, simply a container for subnets. EPGs can only be a member of a single BD. The BD also acts as the default gateway for the subnet.
- One **Contract** to permit HTTP and ICMP traffic. Contracts specify how an EPG communicates or interact with other EPGs using dynamically assigned ACLs. All Contracts consists of 1 or more Subjects, where each Subject contains 1 or more Filters, where each Filter contains 1 or more Entries. Each Entry is equivalent to a line in an ACL applied on the Leaf switch that the EP (within the EPG) is attached to.



### Step 1 Open the IoT tenant on APIC GUI

Open APIC GUI, click on **Tenants** tab and then double-click on “**iot**” tenant:

APIC

System **Tenants** Fabric Virtual Networking L4-L7 Services Admin Operations Apps

ALL TENANTS | Add Tenant | Tenant Search: name or descr | common | **iot** | POD1 | mgmt | infra

All Tenants

Name	Alias	Description	Bridge Domains	VRFs	EPGs	Health Score
common			1	2	0	100
infra			2	2	2	100
<b>iot</b>			2	1	3	100
mgmt			1	2	0	100
POD1			3	1	3	99
POD10			3	1	3	99
POD11			3	1	3	99
POD12			3	1	3	99
POD13			3	1	3	99
POD14			3	1	3	99
POD15			3	1	3	99
POD16			3	1	3	99
POD17			3	1	3	99
POD18			3	1	3	99

Page 1 Of 1 | Objects Per Page: 100 | Displaying Objects 1 - 34 Of 34

### Step 2 Verify the Application Profiles for IoT tenant

On the left panel menu, open the **Application Profiles** folder, you will see some Application Profiles (AP) there, each of them has one EPG.

The screenshot shows the Cisco APIC interface for the 'Tenant iot' tenant. The left sidebar contains a navigation menu with 'Application Profiles' highlighted. The main panel displays the 'Application Profiles' configuration, showing three profiles: HVAC-AP, Patch-AP, and SmartCtrl-AP. Each profile is connected to an EPG (x1).

### Step 3 Verify the Bridge Domains for IoT tenant

On the left panel menu, navigate to **Networking>Bridge Domains** to see the list of BDs:

The screenshot shows the Cisco APIC interface for the 'Tenant iot' tenant. The left sidebar contains a navigation menu with 'Bridge Domains' highlighted. The main panel displays the 'Networking - Bridge Domains' configuration, showing a table of bridge domains.

Name	Alias	Type	Segment	Multicast Address	Custom MAC Address	L2 Unknown Unicast	ARP Flooding	Unicast Routing	Subnet
HVAC-BD		regular	16121805	225.1.79...	00:22:BDc...	Hardware...	True	True	41.41.41.1/24
SmartCtrl...		regular	16482213	225.1.21...	00:22:BDc...	Hardware...	True	True	40.40.40.1/24

### Step 4 Verify the Contracts for IoT tenant

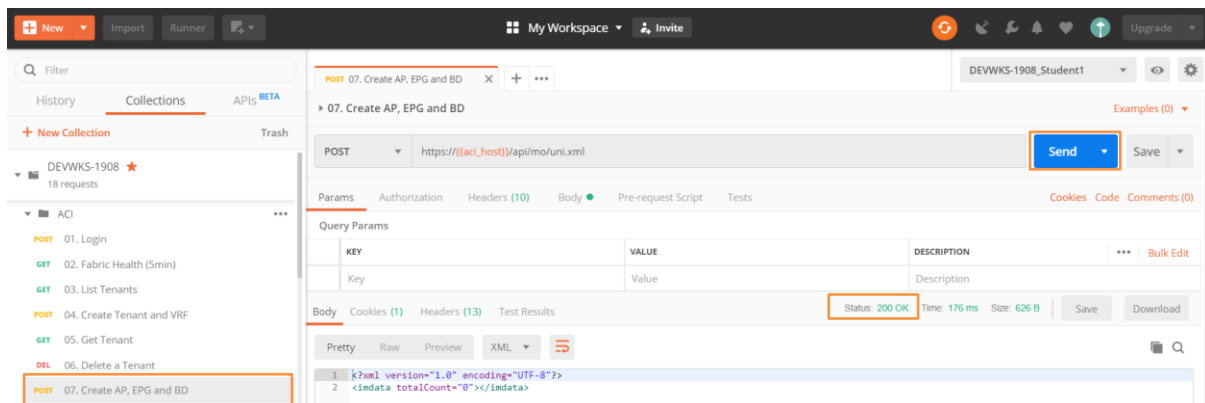
On the left panel menu, navigate to **Contracts>Standard** folder to see the list of contracts:

The screenshot shows the Cisco APIC interface for the 'Tenant iot' tenant. The left sidebar contains a navigation menu with 'Standard' highlighted. The main panel displays the 'Contracts - Standard' configuration, showing a table of standard contracts.

Name	Alias	Scope	QoS Class	Target DSCP	Subjects	Tags	Exported Tenants	Description
Allow_Web		VRF	Unspecif...	Unspecif...	HTTP, ICMP			
HVAC_Po...		VRF	Unspecif...	Unspecif...	HTTP_ICMP			
SmartCtrl...		VRF	Unspecif...	Unspecif...	HTTP_ICMP			

## Step 5 Create a new AP, EPG, BD and Contract for tenant IoT via APIC API

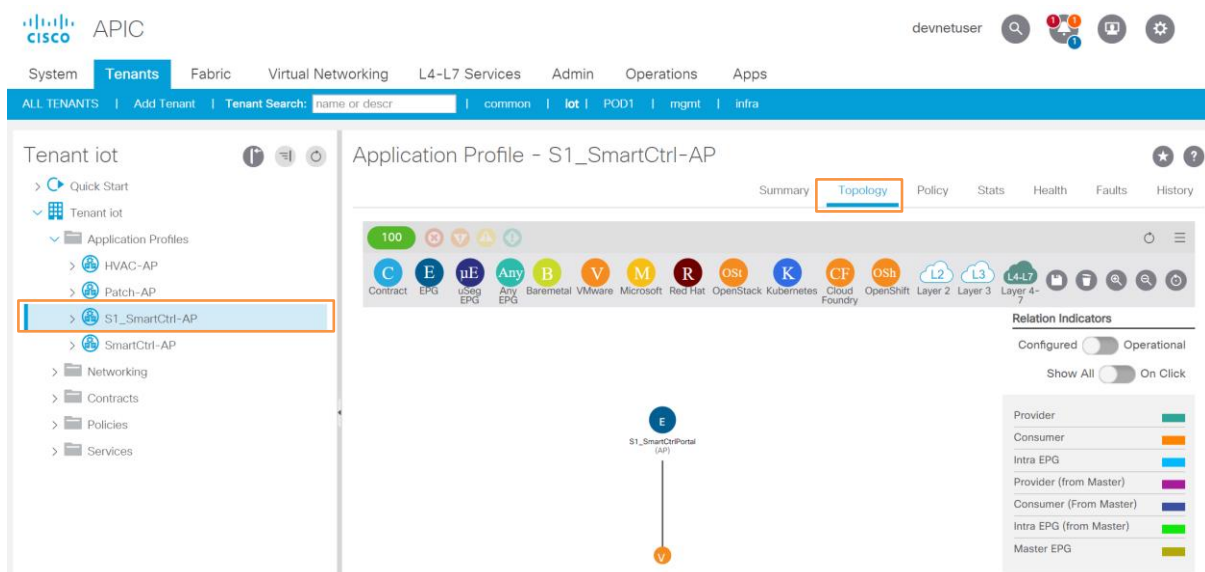
In this step you will create a new AP called “**Sx\_SmartCtrl-AP**”, an application EPG called “**Sx\_SmartCtrlPortal**”, a BD called “**Sx\_SmartCtrl-BD**” and a contract called “**Sx\_SmartCtrl\_Portal\_Web\_Contract**”. Open Postman and send to **APIC** the POST request called “**Create AP, EPG, BD and Contract**”:



Confirm that HTTP response status is **200 OK** and repeat steps 2, 3 and 4 of this task to confirm that the logical constructs were created.

## Step 6 Visualize the application topology on APIC UI

In this step you will see the relationship between different EPGs of your application. Open APIC UI, click on your new AP called “**Sx\_SmartCtrl-AP**”, then click on the **Topology** tab as shown below:

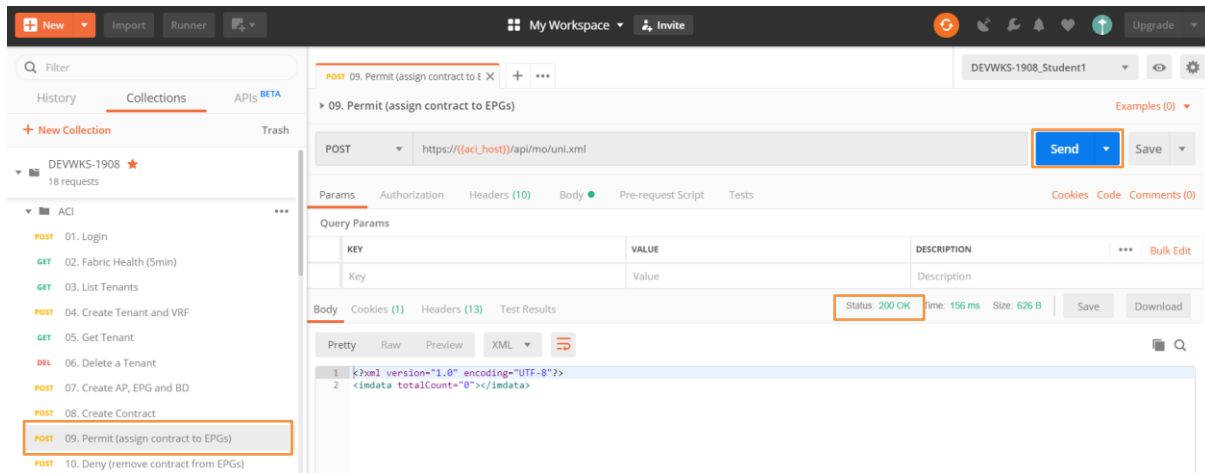


In the next steps we will provide a contract from this EPG and consume the contract from the Light Bulbs external EPG.

## Step 7 Establish a contract between EPGs via APIC API

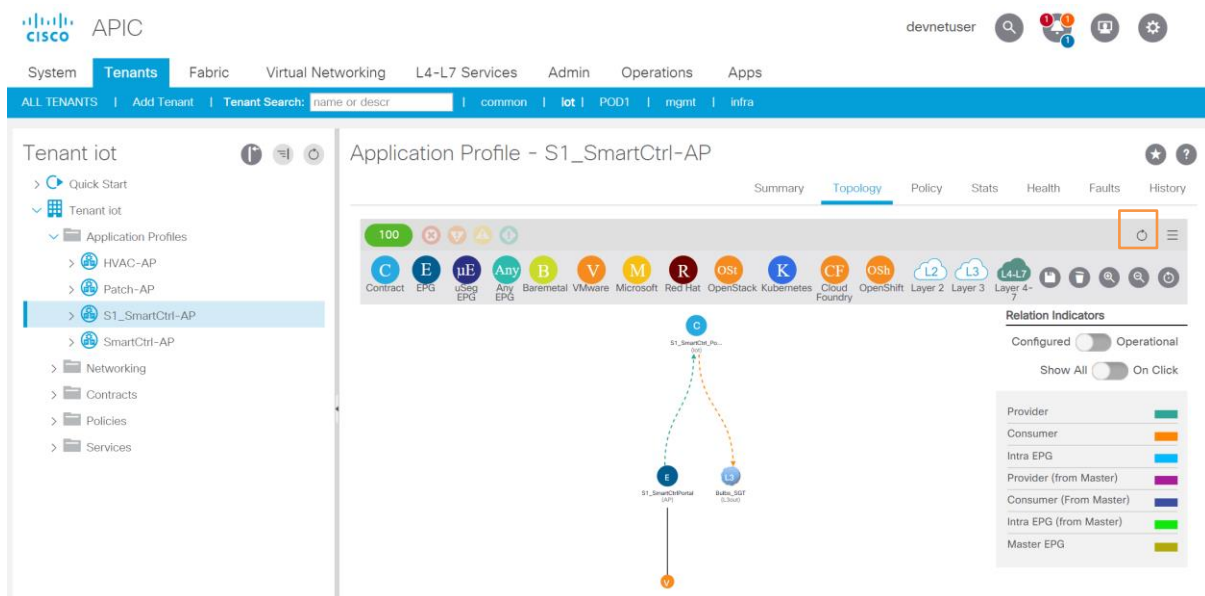
In ACI, by default, endpoints within different EPGs can't talk to each other. In order to allow HTTP and ICMP traffic from Smart Controller servers to light bulbs, we have to provide the contract called “**Sx\_SmartCtrl\_Portal\_Web\_Contract**” from the application EPG called “**Sx\_SmartCtrl-AP**” and consume it from the external EPG called

**“Bulbs\_SGT”**. Open Postman and send to **APIC** the POST request called **“Permit (assign contract to EPGs)”**:



## Step 8 Visualize the application topology on APIC UI

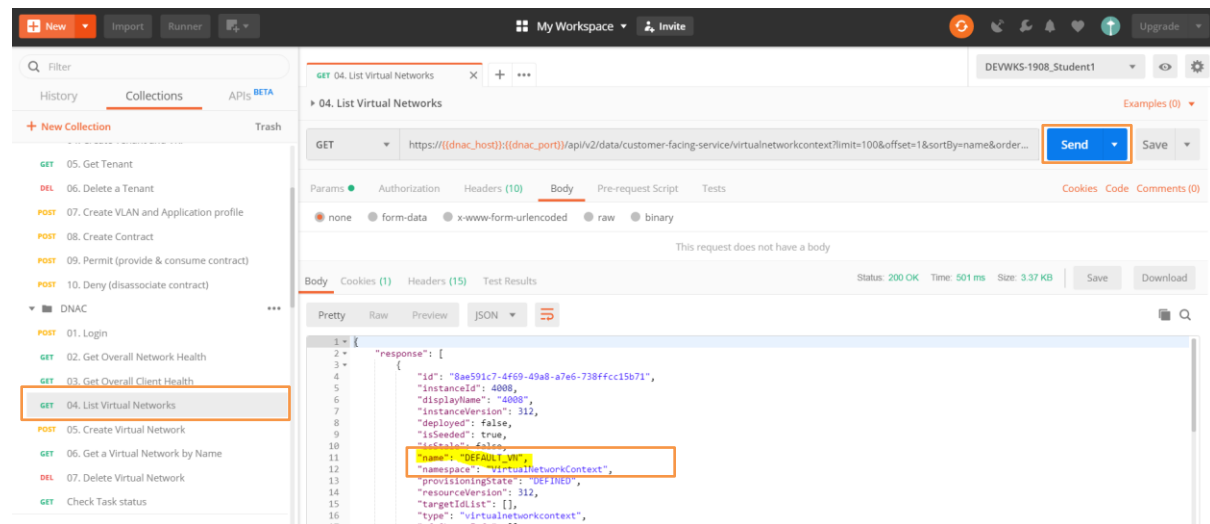
If you go back to the APIC UI, and click the **refresh** icon the topology menu bar, you will see a relationship between the application EPG (Smart Controller servers) and the external EPG (Bulbs), where the 1<sup>st</sup> provides the contract and the 2<sup>nd</sup> consumes it.



# Appendix 1 Extra API calls

## Check Virtual Networks in DNAC via API

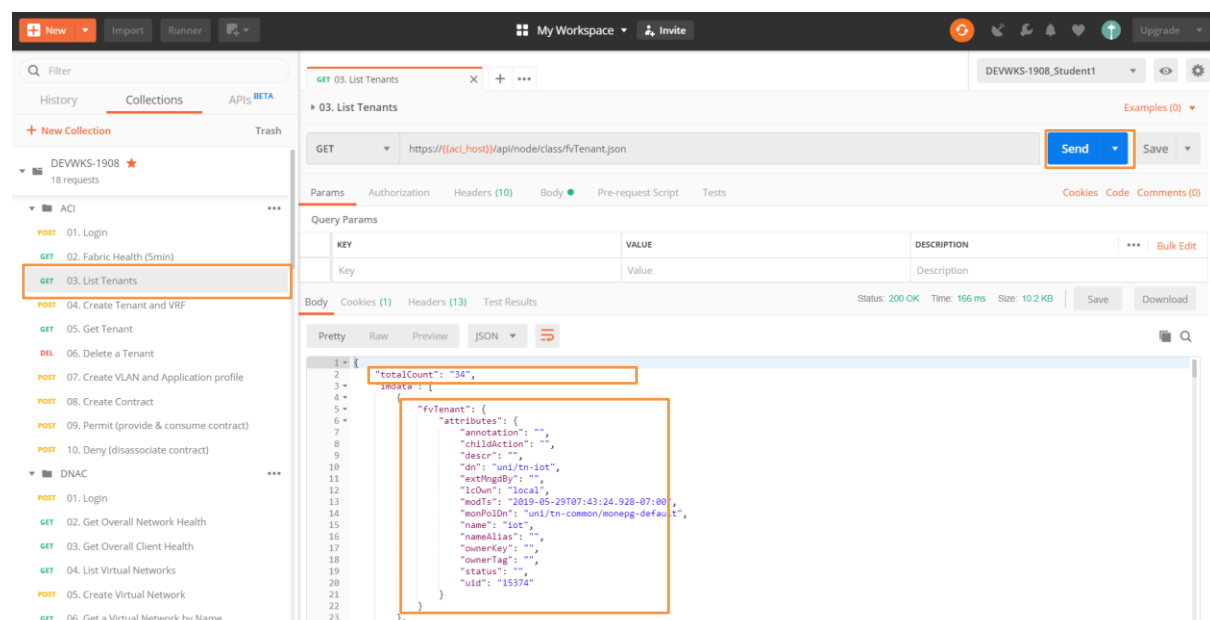
To get the list of virtual networks via APIs, open Postman and send to **DNAC** the GET request called “**List Virtual Networks**”:



A full list of Virtual Networks is received in the response.

## Check ACI tenants via API

To get the list of tenants via API, open Postman and send to **APIC** the GET request called “**List Tenants**”:



## Delete AP, EPG, BD and Contract

Open Postman and send to **APIC** the POST request called “**Delete AP, EPG, BD and Contract**”:



New

Import

Runner

My Workspace

Invite

Upgrade

Filter

History

Collections

APIs BETA

New Collection

Trash

DEVWKS-1908 18 requests

ACI

POST 01. Login

GET 02. Fabric Health (5min)

GET 03. List Tenants

POST 04. Create Tenant and VRF

GET 05. Get Tenant

DEL 06. Delete a Tenant

POST 07. Create AP, EPG, BD and Contract

POST 08. Permit (assign contract to EPGs)

POST 09. Deny (remove contract from EPGs)

POST 10. Delete AP, EPG, BD and Contract

POST 07. Create AP, EPG, BD and Contract

POST 01. Login

POST 10. Delete AP, EPG, BD and Contract

DEVWKS-1908\_Student1

10. Delete AP, EPG, BD and Contract

Examples (0)

POST

https://{{aci\_host}}/api/mo/uni.xml

Send

Save

Params

Authorization

Headers (10)

Body

Pre-request Script

Tests

Cookies

Code

Comments (0)

Query Params

KEY	VALUE	DESCRIPTION		Bulk Edit
Key	Value	Description		

Body

Cookies (1)

Headers (13)

Test Results

Status: 200 OK

Time: 95 ms

Size: 626 B

Save

Download

Pretty

Raw

Preview

XML

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <imdata totalCount="0"></imdata>

```

## Appendix 2 Alexa Robot Skill Intents

- Alexa, ask Robot **what is the status of my network?**
- Alexa, ask Robot to **create a new segment**  
(Accepted segments names are: trusted, untrusted, production)
- Alexa, ask Robot to **delete the segment called { }**  
(Accepted segments names are: trusted, untrusted, production)
- Alexa, ask Robot to **allow light bulbs to talk to smart controllers**
- Alexa, ask Robot to **reset communications for light bulbs and smart controllers**
- Alexa, ask Robot **where is my smart controller server?**