



Testing and Configuration Guide for Cisco Platform Exchange Grid (pxGrid) 2.0

Author: John Eppich



### **Table of Contents**

About this Document	4
Technical Overview	5
Illustrating Cisco pxGrid Client Flow using Chrome Postman	7
Enabling pxGrid	7
Exporting the ISE Identity Certificate into your Browser	8
Creating the pxGrid client Account	9
Activating the Account	10
Performing Services Lookup	12
Obtaining the Access Secret	13
Run GetSessionByIPAddress WebSockets REST API Script	14
Java Code Examples	16
Generating pxGrid client Certificates from ISE Internal CA	16
Converting Certificates to JKS format	17
Downloading Java Examples	19
SessionSubscribe Coding Example	21
Rest of Java Code Examples	25
Sample Configuration	25
pxGrid Control	29
SampleHelper	31
StompFrame	
StompSubscription	
StompPubSubClientEndpoint	
SessionQueryAll	40
SessionQueryByIP	41
CustomServiceProvider	42
CustomServiceConsumer	44
Adaptive Network Control (ANC) Examples	46
ANCSubscribe	46
ANCGetPolicies	47
ANCGetPoliciesByName	48
ANCGetEndpoints	49
ANCGetEndpointsByMAC	50
ANCApplyByIP	51
ANCClearByIP	52

# cisco.

ANCCreatePolicy	53
ANCDeletePolicy	54
ANCGetByOperationID	55
Cisco pxGrid Context-In	56
Enabling pxGrid as Subscriber for Profiling	57
Running API_Simulator	58
Viewing Asset Device in Context Visibility Screen	61
Creating Profiling Policy Based on Asset Attributes	63
Creating Authorization Policy Based on Asset's Logical Profile	67
Verifying Asset as Defined by the Logical Profile	71
Creating Profiling Policy based on Asset Custom Attributes	73
Enabling Custom Attribute Value	74
Creating Authorization Policy Based on Asset's Logical Profile for Custom Attributes	82
Verifying Asset as Defined by the Logical Profile for Custom Attributes	84
Editing Script Values	87
Editing/Adding Customer Values	94
API_Simulator Context-In Code	102
SampleConfiguration	102
PxgridControl	106
PublisherController	108
Custom Publisher	109
CustomSubscriber	113
Device List	115
Devices	115
Console.java	117
Sample Helper. Java	117
Stompframe.java	119
StompSubscription	122
StompPubSubClientEndpoint	122
Endpoint Asset Configuration	126

## About this Document

This document is for Cisco Ecosystem Partners, Cisco Security Technical Alliances (CSTA) Partners and Cisco Internal Solutions implementing Cisco Platform Exchange Grid (pxGrid 2.0). This document accompanies the Cisco Devnet WebSockets & REST API Cisco pxGrid (2.0) development document. https://developer.cisco.com/docs/pxgrid-api/

Cisco pxGrid 2.0 does not rely on "java" and "c" SDK's for development as was the case with Cisco pxGrid 1.0. Instead Cisco pxGrid 2.0 relies on WebSockets & REST API over the STOMP messaging protocol, which alleviates the dependency on SDK's for development. The developer may now use python for pxGrid application development.

This document explains how WebSockets & REST APIs are used in Cisco pxGrid 2.0, and STOMP is the underlying messaging protocol. Code examples are provided to help in code development.

Cisco pxGrid Context-In is a new feature in ISE 2.4, where Internet of Technology (IOT) solutions can provide their asset data for Cisco Identity Services Engine (ISE) to consume and incorporate with the ISE Profiling Policies. An IOT organization' security policy can then be written by assigning these ISE Profiling Policies to ISE Authorization Profiles.

An API\_Simulator.JAR file is included to learn the details of Cisco pxGrid Context-In. This simulates a pxGrid client publishing asset information and ISE taking this context in via pxGrid. Coding examples for this JAR file are also provided.

Chrome POSTMAN is a web-based REST Client that allows you to enter and monitor HTTP requests and responses. This will be used to explain the concepts of WebSockets & REST APIs.

# **Technical Overview**

Cisco Platform Exchange Grid (pxGrid) 2.0 is based on a REpresentational State Transfer (REST) and WebSockets Model, over the Simple Text Oriented Protocol (STOMP) messaging protocol. WebSockets provide quick and scalable bi-directional data transfer, while REST provides quick extensible querying mechanisms – all over the same interface. pxGrid utilizes WebSockets for pubsub components while all one-shot queries (for both control and service data) is done via REST. All message bodies are formatted in JSON.

Cisco pxGrid will use port 8910 on ISE for pxGrid-related REST and WebSocket communications. Messages over Websockets will be sent and received in binary format, these messages should conform to the STOMP messaging protocol. For more information on supported commands on pxGrid please see: https://github.com/cisco-pxgrid/pxgrid-rest-ws/wiki

The following represents the typical pxGrid 2.0 client flow:



#### Create Certificate SSL Context / Account Create

All clients must authenticate to the ISE pxGrid controller either via certificate-based SSL authentication or username-password authentication.

#### Account Activate

All clients request to activate their accounts on the pxGrid server which is handled by the REST API.

#### **Register/Unregister Service**

Service providers will use these APIs to provide and update the necessary information (i.e. resource URLs) from which their services are accessible for other pxGrid clients.

#### Service Lookup

All clients can use this API to dynamically discover all available provider services and their locations.

#### Access Secret

For every service returned that interests a particular client, that client must also query the pxGrid controller for an access secret in order to obtain the information provided by the service.

#### Service Query / Subscribe

With the access secret and service location information in hand, client can then perform REST-based queries or build WebSocket connections to receive information.

We will go through an example of authenticating and registering a pxGrid client using Chrome POSTMAN as the REST client. An account will be created and activated. The pxGrid client will request the session topic or service from the ISE pxGrid node via a service lookup. The service lookup will return the WebSockets URL (WsURL) resource and the ISE node that publishes this session topic. The pxGrid client will then obtain an access secret and retrieve the session information based on the WsURL resource.

The following is a visual representation with the REST and WebSocket architecture:



This representation shows the pxGrid client provider or publisher publishing notifications or topics to the pxGrid controller via WebSockets and another pxGrid client consumer subscribing to these notification or topics.

Cisco pxGrid Context-In follows the same pxGrid client flow and will be discussed in detail in the *Cisco pxGrid Context-in* Section.

# Illustrating Cisco pxGrid Client Flow using Chrome Postman

Chrome POSTMAN will be used to illustrate Cisco pxGrid Client Flow using REST API. First Cisco pxGrid is enabled. Chrome POSTMAN can be downloaded from: https://chrome.google.com/webstore/detail/postman/fhbjgbiflinjbdggehcddcbncdddomop?hl=en

The following steps are as follows:

- Enabling pxGrid on the ISE Node
- Import the ISE identity certificates into your browser
- Creating the pxGrid client Account
- Enabling the Account
- Activating the Account
- Performing Services Lookup
- Obtaining the Access Secret
- Run GetSessionByIPAddress WebSockets REST API script

Please note that the pxGridcontol.java code sample uses the same logic. This is a good way to check your coding as well.

### Enabling pxGrid

 Step 1
 Enable pxGrid

 Select Administration->System Deployment->Edit Node->Enable pxGrid

✓ Ac	Iministration		
✓ <b>▼</b> Mc	onitoring		
	Role	PRIMARY	
	Other Monitoring Node		
V v Po	licy Service		
$\checkmark$	<ul> <li>Enable Session Service</li> </ul>	rices (i)	
	Include Node in	None None	
$\checkmark$	Enable Profiling Serv	rice (i)	
	Enable Threat Centri	c NAC Service (i)	
	Enable SXP Service	1	
	Enable Device Admi	n Service (i)	
	Enable Passive Iden	tity Service (i)	



#### Step 2 Select Save

**Step 3** Verify that the ISE published nodes appear Select **Administration->pxGrid Services** 

diule Identity Services Engine H	orne	Derations   Policy   Admi	inistration		License Warr	ning 🔺 🔍 🥹 🔿
System Identity Management N	letwork Resources	Management pxGrid Services	Feed Service     Threat Centrie	C NAC	Click here to do wireless setup and visit	bility setup Do not show this again.
All Clients Web Clients Capabilities	s Live Log Settings Certi	icates Permissions				- 1
		total i chang approvatory :			1 - 6 of 6 Show	v 25 v per page Page 1 v
Client Name	Client Description	Capabilities	Status	Client Group(s)	Auth Method	Log
ise-mnt-ise24prod		Capabilities(2 Pub, 1 Sub)	Online (XMPP)	Internal	Certificate	View
□ ▶ ise-admin-ise24prod		Capabilities(4 Pub, 2 Sub)	Online (XMPP)	Internal	Certificate	View
□ ► ise-fanout-ise24prod		Capabilities(0 Pub, 0 Sub)	Online (XMPP)	Internal	Certificate	View
☐ ▶ ise-pubsub-ise24prod		Capabilities(0 Pub, 0 Sub)	Online (XMPP)	Internal	Certificate	View
□ ► ise-bridge-ise24prod		Capabilities(0 Pub, 4 Sub)	Online (XMPP)	Internal	Certificate	View

and you have connectivity

Connected to pxGrid ise24prod.lab10.com

### Exporting the ISE Identity Certificate into your Browser

Export the ISE identity certificate, into the PC's truststore. A MacBook Pro was used in this example.

References: https://github.com/cisco-pxgrid/pxgrid-rest-ws/wiki

**Step 1** Import ISE identity certificate into Chrome browser and set to "always trust"

🗰 Keychain Access Fi	e Edit	View	Window	Help
• •				
Certificate Certi	<b>:om</b> rity :h 31, 2019 e is not trus	at 6:50:0: ted	3 PM Eastern	Daylight Time
▼ Trust				
When using this certificat	e: Always	Trust	≎ ?	
Secure Sockets Layer (SS	L) Always	Trust	۵	
Secure Mail (S/MIM	E) Always	Trust	۵	
Extensible Authentication (EA	) Always	Trust	٥	
IP Security (IPse	c) Always	Trust	٥	
Code Signin	g Always	Trust	٥	
Time Stampir	g Always	Trust	۵	
X.509 Basic Polic	y Always	Trust	٥	

- Step 2
   Ensure that you have "Allowed password based account creation" enabled under Administration->pxGrid

   Services->Settings->pxGrid Settings
- **Step 3** Close out the browser and log back in again

### **Creating the pxGrid client Account**

The initial pxGrid client account will be created and returned username and password will be used to access other WebSocket REST calls. For example, this authentication will be used later to determine the ISE peer node for a service lookup that contains the session topic.

**Step 1** Create an account and obtain the username and password that will be used for basic authentication for the other WebSocket REST API calls. In this exercise we will be using password-based authentication.

post $$	https://ise24fc2.lab10.com:8910/pxgrid/control/AccountCreate
---------	--

#### **Step 2** Define the following Headers:

Auth	orization	Headers (2)	Body 鱼	Pre-request Scrip	ot Tests
	Key				Value
~	Accept-Lar	nguage			application/json
<b>~</b>	Content-T	уре			application/json

#### **Step 3** The body should read:

Note: The nodeName provides the pxGrid client name. Please note that the nodename must be Fully Qualified Domain (FQDN) resolvable.

Authorization	Headers (2)	Body 🔵	Pre-req	uest Script	Tests
form-data	x-www-form-	urlencoded	🖲 raw	binary	JSON (application/json) 💉
1 - {					
2 "	nodeName":"ise24	fc2"			
3 }					

**Step 4** You will receive the username and password that will be used for other WebSockets REST calls. The node name represents the pxGrid client node.



### **Activating the Account**

The account needs to be activated by the ISE or pxGrid admin before the pxGrid client can register to the ISE pxGrid node.

**Step 1** Activate the account

POST V https://ise24fc2.lab10.com:8910/pxgrid/control/AccountActivate

**Step 2** Add the following authorization settings:

```
<u>Note</u>: The username and password are from step 6 above, or the results of the account creation.
```

Authorization	Headers (3)	Body 🔵	Pre-request Script	Tests	
Туре			Basic Auth		~
Username			ise24fc2		
Password			Ym9ieeaJVEwbCojU		
		ŀ	<ul> <li>Show Password</li> </ul>		

**Step 3** Add the following headers, the authorization header will appear after the username and password have been configured.

Auth	orization • Headers (3)	Body  Pre-request So	ript Tests
	Key		Value
~	Accept-Language		application/json
~	Content-Type		application/json
~	Authorization		Basic aXNIMjRwcm9kOkFnV2VzNDRNQmNCbWtCUDM=

**Step 4** You will be in the pending state until the ISE admin approves the account

Body	Cookies	Headers (11)	Test R	lesults
Pretty	Raw	Preview	JSON 🗸	₽
1 - { 2 3 4 }	"accour "versio	ntState": "PE on": "2.0.0.1	NDING", .3"	

#### **Step 5** In ISE, you will see pxGrid client name (nodename) in the pending state

Note: Regardless of enabled password-based authentication for client settings, the admin will still have to approve the pending client request.

dentity Services Engine	Home	Operations      Policy      Admit	nistration Vork Centers		License Warning 🔺	୍ ଡ ତ ନ
System Identity Management	Network Resources     Device P	ortal Management pxGrid Services	Feed Service     Threat Centric	NAC		
All Clients Web Clients Capabilities Live Log Settings Certificates Permissions						
√ Enable Ø Disable Ø Approve	😝 Group 🛛 👎 Decline 🛛 🐼 Delete 👻	Refresh Total Pending Approval(1)	•		1 - 8 of 8 Show 25 💌 per	r page Page 1 🗘
Client Name	Client Description	Capabilities	Status	Client Group(s)	Auth Method	Log
Ise-pubsub-ise24fc2		Capabilities(0 Pub, 0 Sub)	Online (XMPP)	Internal	Certificate	View
■ ise-bridge-ise24fc2		Capabilities(0 Pub, 4 Sub)	Online (XMPP)	Internal	Certificate	View
ise-mnt-ise24fc2		Capabilities(2 Pub, 1 Sub)	Online (XMPP)	Internal	Certificate	View
Ise-admin-ise24fc2		Capabilities(6 Pub, 2 Sub)	Online (XMPP)	Internal	Certificate	View
▶ ise-fanout-ise24fc2		Capabilities(0 Pub, 0 Sub)	Online (XMPP)	Internal	Certificate	View
■ ise24fc2	Websokets_App	Capabilities(0 Pub, 0 Sub)	Pending		UserName/Password	View

- Step 6 Select "ise24fc2" and "Approve"
- **Step 7** You will see the following message:

dentity Services Engine	Home	ty ▶ Operations ▶ Policy - Adr	ninistration Vork Cen	ters	License Warning 🔺	
System Identity Management	Network Resources					
All Clients Web Clients Capat	bilities Live Log Si	Are you sure you want to Approve the sele	ected client(s)?			
Senable Obisable Approve	😁 Group 🛛 🗬 Decline 🌔				tem 1 - 8 of 8 Show 25	🔹 per page 🛛 Page 🚺 🌲
Client Name	Client Description			No Yes	Auth Method	Log
□ ► ise-pubsub-ise24fc2	_	capaonicios(o rao, o oco)	Omme (Amme)	anternor	Certificate	View
□ ► ise-bridge-ise24fc2		Capabilities(0 Pub, 4 Sub)	Online (XMPP)	Internal	Certificate	View
□ ► ise-mnt-ise24fc2		Capabilities(2 Pub, 1 Sub)	Online (XMPP)	Internal	Certificate	View
□ ► ise-admin-ise24fc2		Capabilities(6 Pub, 2 Sub)	Online (XMPP)	Internal	Certificate	View
□ ► ise-fanout-ise24fc2		Capabilities(0 Pub, 0 Sub)	Online (XMPP)	Internal	Certificate	View
✓ ► ise24fc2	Websokets_App	Capabilities(0 Pub, 0 Sub)	Pending		UserName/Password	View

#### Step 8 Select "Yes"

#### **Step 9** You should see:

dentity Services Engine	Home   Context Visibility	Operations     Policy     Admin	histration + Work Centers		License Warning 🔺 🔍	. 0
System     Identity Management	Network Resources     Device	e Portal Management pxGrid Services	Feed Service     Threat Centric N	IAC		
All Clients Web Clients Capab	ilities Live Log Settings	Certificates Permissions				
√ Enable Ø Disable Ø Approve	🖯 Group 👎 Decline 🛛 🐼 Delete	<ul> <li>Refresh Total Pending Approval(0)</li> </ul>	•		1 - 8 of 8 Show 25 🔻 per	bage Page
Client Name	Client Description	Capabilities	Status	Client Group(s)	Auth Method	Log
Ise-pubsub-ise24fc2		Capabilities(0 Pub, 0 Sub)	Online (XMPP)	Internal	Certificate	View
Ise-bridge-ise24fc2		Capabilities(0 Pub, 4 Sub)	Online (XMPP)	Internal	Certificate	View
Ise-mnt-ise24fc2		Capabilities(2 Pub, 1 Sub)	Online (XMPP)	Internal	Certificate	View
Ise-admin-ise24fc2		Capabilities(6 Pub, 2 Sub)	Online (XMPP)	Internal	Certificate	View
ise-fanout-ise24fc2		Capabilities(0 Pub, 0 Sub)	Online (XMPP)	Internal	Certificate	View
ise24fc2	Websokets_App	Capabilities(0 Pub, 0 Sub)	Offline (XMPP)		UserName/Password	View

Step 10 If you re-run REST client request again, the account will be enabled

Body	Cookies	Headers (1	1) Test F	Results
Pretty	Raw	Preview	json $\vee$	1
1 - {				
2 3	"accou "versi	ntState": "  on": "2.0.0	ENABLED", .13"	
4 }				

### **Performing Services Lookup**

ServicesLookup is used to find a service and its properties. It determines what topics or services are available on the ISE peer node that publishes these topics. Examples of these topics are Session Directory, RADIUS failures, MDM, Profiler Configuration, System Health, TrustSec, TrustSec Configuration and TrustSec SXP. For more information, please see: <u>https://github.com/cisco-pxgrid/pxgrid-rest-ws/wiki/pxGrid-Consumer</u>

In the case of publishing Internet of Things (IOT) asset devices, this will return the services of the pxGrid client publishing the asset topic.

In this example, we are interested in seeing which node publishes the session information.

**Step 1** Run Service Lookup, to see what services are available on all the ISE nodes

POST 🗸

https://ise24fc2.lab10.com:8910/pxgrid/control/ServiceLookup

#### **Step 2** The Headers should read:

```
Note: The Authorization header should be the username and password from step 8, when creating the account
```

Aut	norization • Headers (3)	Body • Pre-request So	cript Tests
	Кеу		Value
~	Accept-Language		application/json
~	Content-Type		application/json
~	Authorization		Basic aXNIMjRwcm9kOkFnV2VzNDRNQmNCbWtCUDM=

**Step 3** We request session topic which can be found here: <u>https://github.com/cisco-pxgrid/pxgrid-rest-ws/wiki/Session-Directory</u>



**Step 4** Note the nodeName or the ISE node that publishes the session information and the properties JSON example:

Body	Cookies Headers (11) Test Res	ılts
Pretty	Raw Preview JSON V	<u> </u>
1 -	£	
2 -	"services": [	
3 -	{	
4	"name": "com.cisco.is	e.session",
5	"nodeName": "ise-mnt-	seZ4fcZ",
6 -	"properties": {	
7	"sessionTopic": "	<pre>/topic/com.cisco.ise.session",</pre>
8	"groupTopic": "/to	<pre>&gt;pic/com.cisco.ise.session.group",</pre>
9	"wsPubsubService"	"com.cisco.ise.pubsub",
10	"restBaseURL": "h	:tps://ise24fc2.lab10.com:8910/pxgrid/mnt/sd",
11	"restBaseUrl": "h	:tps://ise24fc2.lab10.com:8910/pxgrid/mnt/sd"
12	}	
13	}	
14	]	
15	}	

The **sessionTopic** "/topic/com.cisco.com.ise.session" contains the session attributes of the authenticated endpoint. The **restBaseUrl** <u>https://ise24fc2.lab10.com:8910/pxgrid/mnt/sd</u> will be used later on in the REST Client request to get all authenticated sessions.

The wsPubsubService "com.cisco.ise.pubsub" contains the WebSockets published service.

### **Obtaining the Access Secret**

The Access Secret provides HTTP Basic Auth access to the ISE node that will publish the session information. This will be used in the REST Client API to obtain session information when querying based on an IP address.

#### **Step 1** Obtain access secret



#### **Step 2** The Headers are as follows

Auth	orization 🔵	Headers (3)	Body 🔵	Pre-request Sc	ript Tests
	Key				Value
~	Accept				application/json
~	Content-Typ	e			application/json
~	Authorizatio	n			Basic aXNIMjRwcm9kOkFnV2VzNDRNQmNCbWtCUDM=

**Step 3** Specify the ISE node that will publish the session information as provided by the returned service lookup results.



**Step 4** The access secret is returned below:

Body	Cookies	Headers (1	1) Test	Results
Pretty	Raw	Preview	json $\vee$	Ð
1 - { 2 3 }	"secret	t": "Pe874I	dYp4RsNpmv'	

### Run GetSessionByIPAddress WebSockets REST API Script

The Get Session By IP RESTAPI returns available session information from an IP address query. For more information please see: <u>https://github.com/cisco-pxgrid/pxgrid-rest-ws/wiki/Session-Directory</u>

**Step 1** Run the getSessionByIpAddress REST API script below:

```
POST V https://ise24fc2.lab10.com:8910/pxgrid/mnt/sd/getSessionBylpAddress
```

Note that the restBaseUrl https://ise24fc2.lab10.com:8910/pxgrid/mnt/sd is used in the script

#### **Step 2** Please use the following header information

<u>Note</u> : p	The au	ithori d	zation heade	r contains the u	sername from	n step 8, crea	ting the initial account, and the password contains the	he access secret
		Auth	orization •	Headers (3)	Body	Pre-request Sc	rint Tests	
		/ water in		fiedders (5)	body •	ine request se		
			Key				Value	
		~	Accept-Langu	Jage			application/json	
		~	Content-Type	2			application/json	
		~	Authorization	ı			Basic aXNIMjRwcm9kOjZadTVJcXdXcEhMWDZ1OG8=	

**Step 3** Submit the IP address you want to perform a query action on

Authorizat	ion 鱼	Headers (3)	Body 🔵	Pre-r	equest Script	Tests
form-d	lata	x-www-form-ur	encoded	raw	binary	JSON (application/json) 💉
1 - {						
2	"ip/	Address": 192.10	58.1.15"			
3 }						

**Step 4** Below are the returned session attributes for the endpoint IP address of 192.168.1.15

Pretty	Raw Preview JSON V 📮	Save Response					
1 -	4						
2	"timestamp": "2018-04-10T14:10:25.899Z".						
3	"state": "STARTED".						
4	"userName": "jeppich".						
5	"callingStationId": "10:DD:B1:C9:3C:39",						
6	"calledStationId": "50:3D:E5:C4:05:8C",						
7	"auditSessionId": "0A000010000003A021FC0F4",						
8 -	"ipAddresses": [						
9	"192.168.1.15"						
10	],						
11	"macAddress": "10:DD:B1:C9:3C:39",						
12	"nasIpAddress": "192.168.1.3",						
13	"nasPortId": "GigabitEthernet1/0/12",						
14	"nasPortType": "Ethernet",						
15	"endpointProfile": "Apple-Device",						
16	"endpointOperatingSystem": "Apple Mac OS X 10.7.0 (Lion) - 10.10 (Yosemite) or iOS 4.1 - 8.3 (Darwin 10.0.0 - 14.5.0)",						
17	"adNormalizedUser": "jeppich",						
18	"adUserDomainName": "lab10.com",						
19	"adUserNetBiosName": "LAB10",						
20	"adUserResolvedIdentities": "jeppich@lab10.com",						
21	"adUserResolvedDns": "CN=John Eppich,CN=Users,DC=lab10,DC=com",						
22 -	"providers": [						
23	"None"						
24	],						
25	"endpointCheckResult": "none",						
26	"identitySourcePortStart": 0,						
27	"identitySourcePortEnd": 0,						
28	"identitySourcePortFirst": 0,						
29	"isMachineAuthentication": "false",						
30	"serviceType": "Framed",						
31	"networkDeviceProfileName": "Cisco",						
32	"radius-lowlype": "Wireaw2_1x",						
33	"ssid": "50-30-E5-C4-05-8C",						
34	"mdmRegistered": false,						
35	"mdmCompliant": false,						
36	"mdmDiskEncrypted": false,						
37	"mdmJallBroken": false,						
38	"mdmPinLocked": false						
39	H						



## Java Code Examples

You can download Java code examples once registered to Cisco Devnet <u>www.cisco.com/go/pxgridpartner</u> from <u>https://github.com/cisco-pxgrid/pxgrid-rest-ws</u>

The code samples are as follows:

🛴 CustomServiceConsumer.java
🛄 CustomServiceProvider.java
model
PxgridControl.java
📙 SampleConfiguration.java
📙 SampleHelper.java
🔜 SessionQueryAll.java
🗓 SessionQueryByIP.java
🛼 SessionSubscribe.java
📜 StompFrame.java
🔝 StompPubsubClientEndpoint.java
🛄 StompSubscription.java

For this example, we will run the SessionSubscribe.java code, to see the available contextual information from an authenticated user session.

You can also develop code to subscribe and obtain contextual information from other topics: Radius Failure, these topics are Session Directory, RADIUS failures, MDM, Profiler Configuration, System Health, TrustSec, TrustSec Configuration and TrustSec SXP. For more information, please see: <u>https://github.com/cisco-pxgrid/pxgrid-rest-ws/wiki/pxGrid-Consumer</u>

First, we will need to create certificates your system, which will be the pxGrid client. The pxGrid client will authenticate to the ISE pxGrid node using certificates or pre-shared keys, in this example we will use certificates generated from the ISE internal Certificate Authority (CA).

### Generating pxGrid client Certificates from ISE Internal CA

In this example, we will create the pxGrid client certificate in PKCS12 format. You will also want to test in PEM format, when you go for your pxGrid certification.

#### Step 1 Select Administration->pxGrid Services->Certificates-> provide the following information

Note:	CN name should be Fully	Qualified Domain N	ame (FQDN) resolvable.	PKCS12 format is not supported using Python libraries.
-------	-------------------------	--------------------	------------------------	--

ten Iden	tity Services Engine	Home	Context	Visibility	<ul> <li>Operations</li> </ul>	Polloy	- Administration	Work Centers	
<ul> <li>System</li> </ul>	Identity Manageme	nt 🕨 Networ	k Resources	Device	Portal Manager	nent pxGrid S	Services + Feed S	ervice + Threat Centric N.	AC
All Clients	Web Clients Ca	apabilities	Live Log S	ettings	Certificates	Permissions			
Genera	ate pxGrid Certif	icates							
	1	want to *	Generate a sing	gle certifica	ate (without a cer	tificate signing r	equest)		•
	Common Nam	ie (CN) • je	ohns-macbook-	pro.lab10.	com				
	De	scription p	xGrid						
	Certificate 1	Femplate Px	Grid_Certificate	_Template	0				
-	Subject Alternative Nam	ne (SAN)		john	-macbook-pro.la	b10 —			
	Certificate Download	Format •	PKCS12 formal	(including	certificate chain	; one file for bot	h the certificate chain	and key)	•
	Certificate Pa	ssword •							
	Confirm Pa	ssword •							
								Reset	Create
									_

- Step 2 Select Create
- **Step 3** Download the zipped file
  - You should see:



#### **Step 4** Note, if using PEM format, when you unzip the file you will see the following:

<u>Note:</u> Please refer to <u>https://communities.cisco.com/docs/DOC-71928</u>, using iSE 2.2 Internal CA to deploy to pxGrid clients (java keystores), for Productional ISE deployments, please refer to: <u>https://communities.cisco.com/docs/DOC-68284</u>

- CertificateServicesEndpointSubCA-ise24fc3\_.cer
- CertificateServicesNodeCA-ise24fc3\_.cer
- CertificateServicesRootCA-ise24fc3\_.cer
- ise24fc3.lab10.com\_.cer
- johns-macbook-pro.lab10.com\_john-macbook-pro.lab10.com.cer
- johns-macbook-pro.lab10.com\_john-macbook-pro.lab10.com.key

### **Converting Certificates to JKS format**

The PKCS12 certificate is converted into the keystore filename and truststore filename.

```
openssl pkcs12 -export -out session.p12 -inkey Johns-Macbook-Pro.lab10.com Johns-Macook-Pro.lab10.com.key -in
Johns-Macbook-Pro.lab10.com_Johns-Macook-Pro.lab10.com.cer -chain -CAfile CA1.cer
Enter pass phrase for Johns-Macbook-Pro.lab10.com_Johns-Macook-Pro.lab10.com.key:
Enter Export Password: Cisco123
Verifying - Enter Export Password: Ciscol23
keytool -importkeystore -srckeystore session.p12 -destkeystore session1.jks -srcstoretype PKCS12
Enter destination keystore password: Cisco123
Re-enter new password: Cisco123
Enter source keystore password:
                                 Cisco123
Entry for alias 1 successfully imported.
Import command completed: 1 entries successfully imported, 0 entries failed or cancelled
openssl x509 -outform der -in CA1.cer -out CA1.der
keytool -import -alias session1 -keystore rootsession.jks -file CA1.der
Enter keystore password: Cisco123
Re-enter new password: Cisco123
Owner: CN=Certificate Services Endpoint Sub CA - ise24fc3
Issuer: CN=Certificate Services Node CA - ise24fc3
Serial number: 589713fe8d1d4c99b580aae99e862c4f
Valid from: Thu Apr 12 22:42:16 EDT 2018 until: Thu Apr 13 22:42:14 EDT 2028
Certificate fingerprints:
        MD5: 8E:B3:9F:92:B8:E4:80:51:64:68:4C:72:44:51:15:3F
        SHA1: 5D:EF:20:E1:9C:CA:5D:F7:15:28:FA:1D:4D:4F:A9:79:CD:E5:A6:FC
        SHA256:
FB:C5:84:4B:30:D3:8E:95:B9:FE:28:54:EC:60:A7:E4:4B:A7:6D:1C:8D:8C:0C:15:C0:4B:2C:37:4A:43:8F:0C
        Signature algorithm name: SHA256withRSA
        Version: 3
```

cisco.

```
Extensions:
#1: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [
KeyIdentifier [
0010: A4 6D 33 CE
                                                        .m3.
[CN=Certificate Services Root CA - ise24fc3]
SerialNumber: [ 6f56a636 30094fa4 b4b85ac9 4f5def5b]
1
#2: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
 CA:true
  PathLen: 2147483647
1
#3: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [
 Key_CertSign
1
#4: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 33 47 E4 40 4B 5E 0C 08 77 DE A2 77 30 50 E9 3C 3G.@K^..w..w0P.<
0010: 12 78 92 39
                                                       .x.9
1
1
Trust this certificate? [no]: yes
Certificate was added to keystore
keytool -import -alias session2 -keystore session1.jks -file Johns-Macbook-Pro.lab10.com_Johns-Macook-
Pro.lab10.com.cer
Enter keystore password: Cisco123
Certificate already exists in keystore under alias <1>
Do you still want to add it? [no]: yes
Certificate was added to keystore
keytool -import -alias session3 -keystore rootsession.jks -file CA1.cer
Enter keystore password: Cisco123
Certificate already exists in keystore under alias <session1>
Do you still want to add it? [no]: yes
Certificate was added to keystore
keytool -import -alias session4 -keystore rootsession.jks -file CertificateServicesRootCA-ise24fc3_.cer
Enter keystore password: Cisco123
Owner: CN=Certificate Services Root CA - ise24fc3
Issuer: CN=Certificate Services Root CA - ise24fc3
Serial number: 23471fb4679a4836b6023da18e312e3e
Valid from: Thu Apr 12 22:42:14 EDT 2018 until: Thu Apr 13 22:42:14 EDT 2028
Certificate fingerprints:
        MD5: 94:EA:6F:D5:E6:D6:A4:53:D2:69:7E:C6:6F:02:AB:2D
        SHA1: F8:8A:36:C8:45:F0:A5:01:32:32:E0:8D:59:E4:F9:A2:24:A6:71:47
        SHA256:
19:2F:41:EC:93:C7:EE:BB:CC:22:AB:44:24:FF:95:AF:E0:5F:5F:30:F9:D4:7C:84:43:91:93:A6:47:1C:67:97
        Signature algorithm name: SHA256withRSA
        Version: 3
Extensions:
#1: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
 CA:true
 PathLen:2147483647
1
#2: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [
 Key_CertSign
```

```
]
#3: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: AE 42 CE AB 57 30 7C 75 F8 10 94 25 0E DC DF FA .B..W0.u...%....
0010: 9E 7F 3A 57 ...:W
]
Trust this certificate? [no]: yes
Certificate was added to keystore
```

### **Downloading Java Examples**

In this section, java code examples are downloaded and imported into an Eclipse project

- Step 1 Register for devnet: <u>www.cisco.com/go/pxgridpartner</u>
- Step 2 Open browser to <u>https://github.com/cisco-pxgrid/pxgrid-rest-ws</u>

$\left( \leftarrow  ight)  ightarrow$ C $rac{1}{2}$	🛈 🔒 GitHub, Inc. (US) 🛛 https://github	.com/cisco-pxgrid/pxgrid-rest-ws	V 🚥 💟 🏠 🤍 Searci	1
	This repository Search	Pull requests Issues Mark	ketplace Explore	♦ +• 🕃•
	Cisco-pxgrid / pxgrid-rest-ws			Star 3 % Fork 2
	<> Code ① Issues 0 1 Pull re	equests 0 III Projects 0 III Wiki	Insights	
	Cisco pxGrid with REST and WebSoc	ket		
	T 44 commits	1 branch 🛇 0 releases	🚨 1 contributor	화 Apache-2.0
	Branch: master - New pull request		Create new file Upload files Find	file Clone or download
	🔡 alei121 Remove pointer		Latest	commit f56eea7 7 days ago
	<b>g</b> olang	Remove pointer		7 days ago
	iava java	cleanup		12 days ago
	python	Arguments and code cleanup		9 days ago
	.gitignore	Arguments and code cleanup		9 days ago
		Initial commit		11 months ago
	README.md	Update README.md		11 months ago
	README.md			

**Step 3** Download the zipped file and save locally

T 44 commits	پ <b>1</b> branch	🛇 0 releases	<b>22</b> 10	contributor		⊉ Apache-2.0
anch: master - New pull red	quest		Create new file	Upload files	Find file	Clone or download -
alei121 Remove pointer			Clo	ne with HTT	'PS (?)	Use SSH
golang	Remove pointer		Use	Git or checkou	t with SVN	using the web URL.
java	cleanup		htt	tps://github.c	om/cisco-p	xgrid/pxgrid-r 👔
python	Arguments and	code cleanup				
gitignore	Arguments and	code cleanup		pen in Deskto	Opening (	Download ZIP pxgrid-rest-ws-master.zip
LICENSE	Initial commit			You have cho	sen to ope	in:
README.md	Update READM	E.md		h pxgrid-rest	-ws-maste	er.zip
README md				which is: ZI from: https	P file ://codeloa	d.github.com
				What should	Firefox do	with this file?
				Open wit	h Archive	Utility (default)
pxGrid				Save File		
pyGrid is a protocol fre	mowork that defines the	control machanisms to f	cilitato mar	Do this a	utomaticall	y for files like this from now o
prono is a protocol inc	anework that defines the	Control meetidilishis to h	remare mar			
Benefits of using pxGr	id:					Cancel

**Step 4** Unzip the file

You should see:



- Step 5 Select File->Import->Maven->Existing Maven Projects
- **Step 6** Select Next->enter root directory where the POM file is located

#### Note: The POM.xml file will be enabled

	Import Maven Projects	
Maven Project Select Maven p	s rojects	
Root Directory:	/Applications/master_rest_samples/pxgrid-rest-ws-master/java	Browse
Projects:		
/pom.xm	com.cisco.pxgrid:pxgrid-rest-ws-samples:2.0.0-SNAPSHOT:jar	Select All
		Deselect All
		Select Tree
		Refresh
Add project	s) to working set	
Advanced		
2	Cancel	
(!)		Fillian

**Step 7** Highlight the snapshot.jar file

00	Import Maven Projects		
Maven Project Select Maven p	s rojects		
			Drawna
Root Directory:	/Applications/master_rest_samples/pxgrid-rest-ws-master/java	<b>Ľ</b>	Browse
Projects:			
✓ /pom.xml	I com.cisco.pxgrid:pxgrid-rest-ws-samples:2.0.0-SNAPSHOT:jar		Select All
			Deselect All
			Select Tree
			Deselect Tree
			Refresh
Add project	s) to working set		
			~
Advanced			
?	< Back Next >	Cancel	Finish

#### Step 8 Select Finish

**Step 9** You should see the imported jar file and the following samples:

Note: If you do not see the scripts, select Run->Run As->Maven Clean



### SessionSubscribe Coding Example

This is a common example of subscribing to the session directory service and receiving incoming session notifications in real-time

 Step 1
 Under com.cisco.pxgrid.samples.ise->SessionSubscribe, Right-Click on "mainStringID:void->Run

 As->Run Configurations

You will see

• •	Run Configurations	
Create, manage, and run confi Run a Java application	igurations	
Apache Tomcat     Apache Tomcat     Eclipse Application     Eclipse Application     Eclipse Data Tools     Generic Server     Generic Server	Name:       SessionSubscribe         Image: SessionSubscribe       Image: SessionSubscribe         Project:       pxgrid-rest-ws-samples         Main class:       Image: SessionSubscribe         com.clsco.pxgrid.samples.ise.SessionSubscribe       Sea         Include system libraries when searching for a main class       Include inherited mains when searching for a main class         Stop in main       Stop in main	rse
?	Close	Run

#### **Step 2** Select (**x**) **Arguments**, and enter the following:

Note: The keystorefile name, keystorepassword, truststore filename and truststore password are from the earlier step in generating certificates.

```
-a ise24fc3.lab10.com -u mac03 -k /Applications/master_rest_samples/session1.jks -p Cisco123 -t
/Applications/master_rest_samples/rootsession.jks -q Cisco123
```

where the argument usage is:

usage: SessionSubscribe							
-a,hostname <arg></arg>	Host name (multiple accepted)						
<pre>-d,description <arg></arg></pre>	Description (optional)						
<pre>-k,keystorefilename <arg></arg></pre>	Keystore .jks filename (not required if password is specified)						
-p,keystorepassword <arg></arg>	Keystore password (not required if password is specified)						
-q,truststorepassword <arg></arg>	Truststore password						
-t,truststorefilename <arg></arg>	Truststore .jks filename						
-u,nodename <arg></arg>	Node name						
-w,password <arg></arg>	Password (not required if keystore is specified)						

- Step 3 Select Apply
- Step 4 Select Run
- **Step 5** If you have active user authenticated sessions coming in, you will see the following

----- config -----

Page 23

CISCO

```
hostname = ise24fc3.lab10.com
   nodename = mac03
   password = (not specified)
   description = (not specified)
  keystorefilename = /Applications/master rest samples/session1.jks
   keystorepassword = Cisco123
  truststorefilename = /Applications/master_rest_samples/rootsession.jks
   truststorepassword = Cisco123
22:29:27.354 [main] INFO com.cisco.pxgrid.samples.ise.PxgridControl - AccountActivate request={}
22:29:32.701 [main] INFO com.cisco.pxgrid.samples.ise.PxgridControl - AccountActivate
response={"accountState":"ENABLED", "version":"2.0.0.13"}
22:29:32.701 [main] INFO com.cisco.pxgrid.samples.ise.SessionSubscribe - pxGrid controller version=2.0.0.13
22:29:32.703 [main] INFO com.cisco.pxgrid.samples.ise.PxgridControl - ServiceLookup
request={"name":"com.cisco.ise.session"}
22:29:32.722 [main] INFO com.cisco.pxgrid.samples.ise.PxgridControl - ServiceLookup
response={"services":[{"name":"com.cisco.ise.session","nodeName":"ise-mnt-
ise24fc3","properties":{"sessionTopic":"/topic/com.cisco.ise.session","groupTopic":"/topic/com.cisco.ise.sess
ion.group","wsPubsubService":"com.cisco.ise.pubsub","restBaseURL":"https://ise24fc3.lab10.com:8910/pxgrid/mnt
/sd","restBaseUrl":"https://ise24fc3.lab10.com:8910/pxgrid/mnt/sd"}}]}
22:29:32.722 [main] INFO com.cisco.pxgrid.samples.ise.SessionSubscribe
wsPubsubServiceName=com.cisco.ise.pubsub sessionTopic=/topic/com.cisco.ise.session
22:29:32.722 [main] INFO com.cisco.pxgrid.samples.ise.PxgridControl - ServiceLookup
request={"name":"com.cisco.ise.pubsub"}
22:29:32.733 [main] INFO com.cisco.pxgrid.samples.ise.PxgridControl - ServiceLookup
response={"services":[{"name":"com.cisco.ise.pubsub","nodeName":"ise-pubsub-
ise24fc3", "properties":{"wsUrl":"wss://ise24fc3.lab10.com:8910/pxgrid/ise/pubsub"}}]}
22:29:32.733 [main] INFO com.cisco.pxgrid.samples.ise.SessionSubscribe -
wsUrl=wss://ise24fc3.lab10.com:8910/pxgrid/ise/pubsub
22:29:32.735 [main] INFO com.cisco.pxgrid.samples.ise.PxgridControl - AccessSecret
request={"peerNodeName":"ise-pubsub-ise24fc3"}
22:29:32.895 [main] INFO com.cisco.pxgrid.samples.ise.PxgridControl - AccessSecret
response={"secret":"36QiBjNQdWnR6lPf"}
22:29:34.805 [Grizzly(1)] INFO com.cisco.pxgrid.samples.ise.StompPubsubClientEndpoint - WS onOpen
22:29:34.833 [main] INFO com.cisco.pxgrid.samples.ise.StompPubsubClientEndpoint - STOMP CONNECT
host=ise24fc3.lab10.com
22:29:34.840 [main] INFO com.cisco.pxgrid.samples.ise.StompPubsubClientEndpoint - STOMP SUBSCRIBE
topic=/topic/com.cisco.ise.session
22:29:34.842 [Grizzly(2)] INFO com.cisco.pxgrid.samples.ise.StompPubsubClientEndpoint - STOMP CONNECTED
version=1.2
press <enter> to disconnect...
22:31:57.678 [Grizzly(1)] INFO com.cisco.pxgrid.samples.ise.SessionSubscribe -
Content={"sessions":[{"timestamp":"2018-04-
16T02:31:53.599Z", "state": "STARTED", "userName": "jeppich", "callingStationId": "10:DD:B1:C9:3C:39", "calledStatio
nId":"50:3D:E5:C4:05:8C", "auditSessionId":"0A000001000002D02D9EFBF", "ipAddresses":["192.168.1.136"], "macAddr
ess":"10:DD:B1:C9:3C:39", "nasIpAddress":"192.168.1.3", "nasPortId":"GigabitEthernet1/0/12", "nasPortType":"Ethe
rnet", "endpointProfile":"Apple-Device", "endpointOperatingSystem":"Apple Mac OS X 10.7.0 (Lion) - 10.10
(Yosemite) or iOS 4.1 - 8.3 (Darwin 10.0.0 -
14.5.0)", "adNormalizedUser": "jeppich", "adUserDomainName": "lab10.com", "adUserNetBiosName": "LAB10", "adUserResol vedIdentities": "jeppich@lab10.com", "adUserResolvedDns": "CN\u003dJohn
Eppich.CN\u003dUsers,DC\u003dlab10,DC\u003dcom", "providers":["None"], "endpointCheckResult":"none", "identitySo
urcePortStart":0, "identitySourcePortEnd":0, "identitySourcePortFirst":0, "isMachineAuthentication":"false", "ser
viceType":"Framed", "networkDeviceProfileName":"Cisco", "radiusFlowType":"Wired802_1x", "ssid":"50-3D-E5-C4-05-
8C", "mdmRegistered": false, "mdmCompliant": false, "mdmDiskEncrypted": false, "mdmJailBroken": false, "mdmPinLocked":
false}]}
22:32:21.684 [Grizzly(1)] INFO com.cisco.pxgrid.samples.ise.SessionSubscribe - Content={"sessions":[{"timestamp":"2018-04-
16T02:32:17.261Z", "state": "STARTED", "userName": "LAB10\\pxgrid5", "callingStationId": "00:0C:29:01:5D:E8", "calle
dStationId":"50:3D:E5:C4:05:96","auditSessionId":"0A000001000000190003320D","ipAddresses":["192.168.1.9"],"ma
cAddress":"00:0C:29:01:5D:E8", "nasIpAddress":"192.168.1.3", "nasPortId":"GigabitEthernet1/0/22", "nasPortType":
"Ethernet", "endpointProfile":"Windows7-Workstation", "endpointOperatingSystem":"Windows 7
Professional", "adNormalizedUser": "pxgrid5", "adUserDomainName": "lab10.com", "adUserNetBiosName": "LAB10", "adUser
ResolvedIdentities": "pxgrid5@labl0.com", "adUserResolvedDns": "CN\u003dpxgrid5, CN\u003dUsers, DC\u003dlab10, DC\u
003dcom", "providers": ["None"], "endpointCheckResult": "none", "identitySourcePortStart": 0, "identitySourcePortEnd
":0, "identitySourcePortFirst": 0, "isMachineAuthentication": "false", "serviceType": "Framed", "networkDeviceProfil
eName": "Cisco", "radiusFlowType": "Wired802_1x", "sil: "50-3D-E5-C4-05-
96", "mdmRegistered":false, "mdmCompliant":false, "mdmDiskEncrypted":false, "mdmJailBroken":false, "mdmPinLocked":
false } ] }
22:36:17.720 [Grizzly(2)] INFO com.cisco.pxgrid.samples.ise.SessionSubscribe -
Content={"sessions":[{"timestamp":"2018-04-
16T02:36:12.99Z","state":"STARTED","userName":"LAB10\\pxgrid5","callingStationId":"00:0C:29:01:5D:E8","called
StationId":"50:3D:E5:C4:05:96","auditSessionId":"0A000001000000190003320D","ipAddresses":["192.168.1.9"],"mac
Address":"00:0C:29:01:5D:E8","nasIpAddress":"192.168.1.3","nasPortId":"GigabitEthernet1/0/22","nasPortType":"
```

Ethernet", "endpointProfile": "Windows7-Workstation", "endpointOperatingSystem": "Windows 7 Professional", "adNormalizedUser": "pxgrid5", "adUserDomainName": "labl0.com", "adUserNetBiosName": "LAB10", "adUser ResolvedIdentities": "pxgrid5@labl0.com", "adUserResolvedDns": "CN\u003dpxgrid5, CN\u003dUsers, DC\u003dlab10, DC\u 003dcom", "providers": ["None"], "endpointCheckResult": "none", "identitySourcePortStart": 0, "identitySourcePortEnd ":0, "identitySourcePortFirst": 0, "isMachineAuthentication": "false", "serviceType": "Framed", "networkDeviceProfil eName": "Cisco", "radiusFlowType": "Wired802_1x", "ssid": "50-3D-E5-C4-05- 96", "mdmRegistered": false, "mdmCompliant": false, "mdmDiskEncrypted": false, "mdmJailBroken": false, "mdmPinLocked":
12:49:17 586 [Grizzly(2)] INFO com cisco pygrid samples ise SessionSubscribe _
Content={"sesions": {{"timestamp": "2018-04-
<pre>16T02:47:32.481Z", "state": "DISCONNECTED", "userName": "jeppich", "callingStationId": "10:DD:B1:C9:3C:39", "calledS tationId": "50:3D:E5:C4:05:8C", "auditSessionId": "0A0000010000002D02D9EFBF", "ipAddresses": ["192.168.1.136"], "ma cAddress": "10:DD:B1:C9:3C:39", "nasIpAddress": "192.168.1.3", "nasPortId": "GigabitEthernet1/0/12", "nasPortType": "Ethernet", "endpointProfile": "Apple-Device", "endpointOperatingSystem": "Apple Mac OS X 10.7.0 (Lion) - 10.10 (Yosemite) or iOS 4.1 - 8.3 (Darwin 10.0.0 -</pre>
14.5.0)", "adNormalizedUser": "jeppich", "adUserDomainName": "lab10.com", "adUserNetBiosName": "LAB10", "adUserResol
Eppich.CN\u003dUsers,DC\u003dlab10,DC\u003dcom", "providers":["None"], "endpointCheckResult":"none", "identitySo urcePortStart":0, "identitySourcePortEnd":0, "identitySourcePortFirst":0, "isMachineAuthentication":"false", "ser viceType":"Framed", "networkDeviceProfileName":"Cisco", "radiusFlowType":"Wired802_1x", "ssid":"50-3D-E5-C4-05- 8C", "mdmRegistered":false, "mdmCompliant":false, "mdmDiskEncrypted":false, "mdmJailBroken":false, "mdmPinLocked":
false}]}
<pre>/22:49:17.588 [Grizziy(2)] INFO com.clsco.pxgrid.samples.ise.sessionsubscribe - Contonte_["graciong", 1. ["timochamp", "2019.04</pre>
<pre>Content-{ Sessions :[{ timestamp : 2010-04= 16T02:48:13.485Z","state":"STARTED","userName":"jeppich","callingStationId":"10:DD:B1:C9:3C:39","calledStatio nId":"50:3D:E5:C4:05:8C","auditSessionId":"0A0000010000002F02F84944","ipAddresses":["192.168.1.136"],"macAddr ess":"10:DD:B1:C9:3C:39","nasIpAddress":"192.168.1.3","nasPortId":"GigabitEthernet1/0/12","nasPortType":"Ethe rnet","endpointProfile":"Apple-Device","endpointOperatingSystem":"Apple Mac OS X 10.7.0 (Lion) - 10.10 (Yosemite) or iOS 4.1 - 8.3 (Darwin 10.0.0 -</pre>
14.5.0)", "adNormalizedUser": "jeppich", "adUserDomainName": "lab10.com", "adUserNetBiosName": "LAB10", "adUserResol
<pre>vedIdentities":"jeppich@lab10.com","adUserResolvedDns":"CN\u003dJohn Eppich,CN\u003dUsers,DC\u003dlab10,DC\u003dcom","providers":["None"],"endpointCheckResult":"none","identitySo urcePortStart":0,"identitySourcePortEnd":0,"identitySourcePortFirst":0,"isMachineAuthentication":"false","ser viceType":"Framed","networkDeviceProfileName":"Cisco","radiusFlowType":"Wired802_1x","ssid":"50-3D-E5-C4-05- 8C","mdmRegistered":false,"mdmCompliant":false,"mdmDiskEncrypted":false,"mdmJailBroken":false,"mdmPinLocked": false}]}</pre>
22:49:17.618 [Grizzly(2)] INFO com.cisco.pxgrid.samples.ise.StompPubsubClientEndpoint - WS onClose

closeReason code=VIOLATED\_POLICY phrase=Did not receive a pong: too slow ...

#### **Step 6** Select **Operations->RADIUS->Live Logs**, to see the authenticated endpoints in ISE.

dhaha cisco	Ident	ity Services Engine	Home	Contex	d Visibility	- Operations	▶ Policy	Administration	Work Centers				1 License Wa	ning 🔺	۹ 😐	• •
₹R/	ADIUS	Threat-Centric NAC Live	Logs ▶1	TACACS	Troublesh	oot + Adaptive	Network Control	Reports			[	lick here to do a	wireless setur, and vis	hility setup D	o not show th	is again X
Live	Logs	Live Sessions									Ľ		moless setup and his	binty actup D	o not onow an	o uguni.
	Misconfigured Supplicants (9)			cants 🕄	Misco	nfigured Networl	k Devices 🕄	RADIU	S Drops 🕲	Client Stop	oped Respondi	ng 🔁	Repeat Count	er 🔁		
			0			0			0		0		4			
										Refresh	Never	- Show	Latest 20 records	- Within	Last 3 hour	.s •
S	Refresh	Reset Repeat Count	ts 💆 E	xport To 🕶											<b>T</b> Filter	- ¢-
	Time		Status		Details	Repeat	Identity	Endp	oint ID	Endpoint P	Authentica	t Authoriz	ati Authorizati	IP Ad	dress	Networ
×				•			Identity	End	point ID	Endpoint Profi	Authenticat	ior Authoriz	ation Authorizatio	IP Ad	dress	• Networ
	Apr 1	6, 2018 02:48:13.485 AM		0	<u>o</u>	3	jeppich	10:DI	D:B1:C9:3C:39	Apple-Device	Default >> D	) Default >	B PermitAcces	s 192.16	8.1.136	
	Apr 1	6, 2018 02:36:12.990 AM		0	-0	1	LAB10\pxgrid5	00:00	:29:01:5D:E8	Windows7	Default >> D	) Default >	> B PermitAcces	s 192.16	8.1.9	
	Apr 1	6, 2018 02:36:12.375 AM		<b>~</b>	Q			00:00	:29:01:5D:E8							Switch
	Apr 1	6, 2018 02:32:16.659 AM		<b>~</b>	0		LAB10\pxgrid5	00:00	2:29:01:5D:E8	Windows7	Default >> D	) Default >	> B PermitAcces	s 192.16	8.1.9	Switch
	Apr 1	6, 2018 02:32:16.498 AM		~	0			00:00	:29:01:5D:E8							Switch
	Apr 1	6, 2018 02:31:53.138 AM		~	9			10:DI	D:B1:C9:3C:39							Switch
	Apr 1	6, 2018 02:07:04.897 AM		<b>~</b>	Q		jeppich	10:DI	D:B1:C9:3C:39	Apple-Device	Default >> D	) Default >	> B PermitAcces	s 192.16	8.1.136	Switch
	Apr 1	6, 2018 01:49:09.962 AM		~	9		10:DD:B1:C9:3	C:39 10:DI	D:B1:C9:3C:39	Apple-Device	Default >> N	I Default >	> B PermitAcces	s 192.16	8.1.136	Switch
	Apr 1	5, 2018 10:13:27.331 PM		~	Q		jeppich	10:DI	D:B1:C9:3C:39	Apple-Device	Default >> D	) Default >	> B PermitAcces	s 192.16	8.1.136	Switch
	Apr 1	5, 2018 09:31:08.394 PM		~	0		10:DD:B1:C9:3	C:39 10:DI	D:B1:C9:3C:39	Apple-Device	Default >> N	1 Default >	> B PermitAcces	s 192.16	8.1.136	Switch

## **Rest of Java Code Examples**

These rest of the java code examples are explained

### **Sample Configuration**

This code configures the –D PXGRID Hostname, -D PXGRID Username, -D PXGRID PASSWORD, -D PXGRID GROUP, -D PXGRID Description, -D PXGRID Keystore\_Filename, -D PXGRID Keystore\_Psssword, -D PXGRID Truststore Filename, and –D Truststore Password values. This also sets up the keystores.

```
package com.cisco.pxgrid.samples.ise.http;
import java.io.FileInputStream;
import java.io.IOException;
import java.net.Authenticator;
import java.net.PasswordAuthentication;
import java.net.Socket;
import java.security.GeneralSecurityException;
import java.security.KeyStore;
import java.security.Principal;
import java.security.PrivateKey;
import java.security.cert.Certificate;
import java.security.cert.CertificateException;
import java.security.cert.CertificateFactory;
import java.security.cert.X509Certificate;
import java.util.Collection;
import java.util.Enumeration;
import javax.net.ssl.HttpsURLConnection;
import javax.net.ssl.KeyManager;
import javax.net.ssl.KeyManagerFactory;
import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManager;
import javax.net.ssl.TrustManagerFactory;
import javax.net.ssl.X509KeyManager;
import javax.net.ssl.X509TrustManager;
public class SampleConfiguration {
    protected final static String PROP HOSTNAMES="PXGRID HOSTNAMES";
    protected final static String PROP_USERNAME="PXGRID_USERNAME";
    protected final static String PROP_PASSWORD="PXGRID_PASSWORD";
    protected final static String PROP GROUP="PXGRID GROUP";
    protected final static String PROP_DESCRIPTION="PXGRID_DESCRIPTION";
    protected final static String PROP_KEYSTORE_FILENAME="PXGRID_KEYSTORE_FILENAME";
    protected final static String PROP_KEYSTORE_PASSWORD="PXGRID_KEYSTORE PASSWORD";
    protected final static String PROP_TRUSTSTORE_FILENAME="PXGRID_TRUSTSTORE_FILENAME";
    protected final static String PROP_TRUSTSTORE_PASSWORD="PXGRID_TRUSTSTORE_PASSWORD";
    private String[] hostnames;
    private String username;
    private String password;
    private String[] groups;
    private String description;
    private SSLContext sslContext;
    private String keystoreFilename;
    private String keystorePassword;
    private String truststoreFilename;
    private String truststorePassword;
        public SampleConfiguration() throws GeneralSecurityException, IOException {
               load();
               print();
        }
```

```
cisco.
```

```
public String getUserName() {
               return username;
        }
    public void setUsername(String username) {
               this.username = username;
        public String[] getGroups() {
               return groups;
        }
       public String getDescription() {
               return description;
        }
    public SSLContext getSSLContext() {
        return sslContext;
    public String getPassword() {
               return password;
        }
    public String[] getHostnames() {
               return hostnames:
        }
    private void load() throws GeneralSecurityException, IOException {
        String hostnameProperty = System.getProperty(PROP_HOSTNAMES);
        username = System.getProperty(PROP USERNAME);
        password = System.getProperty(PROP_PASSWORD);
        String group_property = System.getProperty(PROP_GROUP);
        description = System.getProperty(PROP_DESCRIPTION);
        keystoreFilename = System.getProperty(PROP_KEYSTORE_FILENAME);
        keystorePassword = System.getProperty(PROP KEYSTORE PASSWORD);
        truststoreFilename = System.getProperty(PROP_TRUSTSTORE_FILENAME);
        truststorePassword = System.getProperty(PROP_TRUSTSTORE_PASSWORD);
        if (hostnameProperty == null || hostnameProperty.isEmpty()) throw new
IllegalArgumentException("Missing " + PROP HOSTNAMES);
        if (username == null || username.isEmpty()) throw new IllegalArgumentException("Missing " +
PROP_USERNAME);
        if (truststoreFilename == null || truststoreFilename.isEmpty()) throw new
IllegalArgumentException("Missing " + PROP_TRUSTSTORE_FILENAME);
    if (truststorePassword == null || truststorePassword.isEmpty()) throw new
IllegalArgumentException("Missing " + PROP_TRUSTSTORE_PASSWORD);
        hostnames = hostnameProperty.split(",");
        if (group_property != null && !group_property.isEmpty()) {
                       groups = group_property.split(",");
        }
        if (description != null) {
                if (description.isEmpty()) description = null;
                else description = description.trim();
        }
        sslContext = SSLContext.getInstance("TLSv1.2");
        sslContext.init(getKeyManagers(), getTrustManagers(), null);
    }
    public void setupAuth(HttpsURLConnection https) throws GeneralSecurityException, IOException {
               Authenticator.setDefault(new MyAuthenticator());
    }
    private class MyAuthenticator extends Authenticator {
        public PasswordAuthentication getPasswordAuthentication() {
            return (new PasswordAuthentication(username, password.toCharArray()));
        }
```



```
}
       private KeyManager[] getKeyManagers() throws IOException, GeneralSecurityException {
               if (keystoreFilename == null || keystoreFilename.isEmpty())
                       return null;
               KeyStore ks = keystoreFilename.endsWith(".p12") ? KeyStore.getInstance("pkcs12") :
KeyStore.getInstance("JKS");
               FileInputStream in = new FileInputStream(keystoreFilename);
               ks.load(in, keystorePassword.toCharArray());
               in.close();
               KeyManagerFactory kmf =
KeyManagerFactory.getInstance(KeyManagerFactory.getDefaultAlgorithm());
               kmf.init(ks, keystorePassword.toCharArray());
               KeyManager[] mngrs = kmf.getKeyManagers();
               if (mngrs == null || mngrs.length == 0) {
                       throw new GeneralSecurityException("no key managers found");
               }
               if (mngrs[0] instanceof X509KeyManager == false) {
                       throw new GeneralSecurityException("key manager is not for X509");
               }
               return new KeyManager[] { new SampleX509KeyManager((X509KeyManager) mngrs[0]) };
       private TrustManager[] getTrustManagers() throws IOException, GeneralSecurityException {
               FileInputStream in = new FileInputStream(truststoreFilename);
               KeyStore ks = null;
               if(truststoreFilename.endsWith(".pem")) {
                       ks = KeyStore.getInstance("JKS");
                       ks.load(null, null);
                       CertificateFactory certFac = CertificateFactory.getInstance("X.509");
                       Collection<? extends Certificate> certs = certFac.generateCertificates(in);
                       int i = 0;
                       for(Certificate c : certs) {
                              ks.setCertificateEntry("trust-" + i, c);
               } else if(truststoreFilename.endsWith(".p12")) {
                       ks = KeyStore.getInstance("pkcs12");
                       ks.load(in, truststorePassword.toCharArray());
               } else {
                       ks = KeyStore.getInstance("JKS");
                       ks.load(in, truststorePassword.toCharArray());
               }
               in.close();
               Enumeration<String> e = ks.aliases();
               boolean hasCertEntries = false;
               while (e.hasMoreElements()) {
                       String alias = e.nextElement();
                       if (ks.isCertificateEntry(alias)) {
                              hasCertEntries = true;
                       }
               }
               if (hasCertEntries == false) {
                       e = ks.aliases();
                       while (e.hasMoreElements()) {
                              String alias = e.nextElement();
                               if (ks.isKeyEntry(alias)) {
                                      Certificate[] chain = ks.getCertificateChain(alias);
                                      for (int i = 0; i < chain.length; ++i) {
    ks.setCertificateEntry(alias + "." + i, chain[i]);</pre>
                                      }
```



```
}
                      }
               }
               TrustManagerFactory tmf =
TrustManagerFactory.getInstance(TrustManagerFactory.getDefaultAlgorithm());
               tmf.init(ks);
               TrustManager[] tms = tmf.getTrustManagers();
               if (tms == null || tms.length == 0) {
                      throw new GeneralSecurityException("no trust managers found");
               }
               if (tms[0] instanceof X509TrustManager == false) {
                      throw new GeneralSecurityException("trust manager is not for X509");
               }
               return new TrustManager[] { new SampleX509TrustManager((X509TrustManager) tms[0]) };
       3
       private static class SampleX509KeyManager implements X509KeyManager {
               private X509KeyManager mngr;
               public SampleX509KeyManager(X509KeyManager mngr) {
                      this.mngr = mngr;
               }
               @Override
               public String chooseClientAlias(String[] arg0, Principal[] arg1, Socket arg2) {
                      String alias = mngr.chooseClientAlias(arg0, arg1, arg2);
                      if (alias == null) {
                              alias = mngr.chooseClientAlias(arg0, null, arg2);
                              if (alias == null) {
                                      throw new RuntimeException("no client certificate found ...");
                              }
                      }
                      return alias;
               }
               @Override
               public String chooseServerAlias(String arg0, Principal[] arg1, Socket arg2) {
                      throw new RuntimeException("Not implemented");
               }
               @Override
               public X509Certificate[] getCertificateChain(String arg0) {
                      return mngr.getCertificateChain(arg0);
               }
               @Override
               public String[] getClientAliases(String arg0, Principal[] arg1) {
                      return mngr.getClientAliases(arg0, null);
               }
               @Override
               public PrivateKey getPrivateKey(String arg0) {
                      return mngr.getPrivateKey(arg0);
               }
               @Override
               public String[] getServerAliases(String arg0, Principal[] arg1) {
                      throw new RuntimeException("Not implemented");
               }
       }
       private static class SampleX509TrustManager implements X509TrustManager {
               private X509TrustManager mngr;
```

# cisco.

```
public SampleX509TrustManager(X509TrustManager mngr) {
                      this.mngr = mngr;
              }
               @Override
              public void checkClientTrusted(X509Certificate[] arg0, String arg1) throws
CertificateException {
                      throw new RuntimeException("not implemented");
              }
               @Override
              public void checkServerTrusted(X509Certificate[] arg0, String arg1) throws
CertificateException {
                      try {
                             mngr.checkServerTrusted(arg0, arg1);
                      } catch (CertificateException e) {
                              throw new CertificateException("Server certificate is not trusted:" +
arg0[0].getSubjectX500Principal(),
                                             e);
                      }
              }
               @Override
              public X509Certificate[] getAcceptedIssuers() {
                      return mngr.getAcceptedIssuers();
              }
       }
   private void print() {
       System.out.println("----- properties ------");
        System.out.print(" hostnames=");
        for (String hostname : hostnames) System.out.print(hostname + " ");
       System.out.println();
       System.out.println(
                             username=" + username);
       System.out.println("
                             password=" + password);
       System.out.print(" groups=");
        for (String group : groups) System.out.print(group + " ");
       System.out.println();
                             description=" + description);
       System.out.println(
       System.out.println("
                             keystoreFilename=" + keystoreFilename);
       System.out.println("
                             keystorePassword=" + keystorePassword);
       System.out.println("
                             truststoreFilename=" + truststoreFilename);
       System.out.println(" truststorePassword=" + truststorePassword);
       System.out.println("------");
   }
```

### pxGrid Control

This code provides the pxGrid client with account creation on the ISE pxGrid node and service lookup request and access secret to the peer node, publishing the topic information. In the example, using API\_Simulator, the peer node would reflect the pxGrid client.

```
package com.cisco.pxgrid.samples.ise.http;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.URL;
import java.util.Base64;
import java.util.Map;
import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.HttpsURLConnection;
import javax.net.ssl.SSLSession;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
```

cisco.

```
import com.cisco.pxgrid.model.AccessSecretRequest;
import com.cisco.pxgrid.model.AccessSecretResponse;
import com.cisco.pxgrid.model.AccountActivateRequest;
import com.cisco.pxgrid.model.AccountActivateResponse;
import com.cisco.pxgrid.model.AccountCreateRequest;
import com.cisco.pxgrid.model.AccountCreateResponse;
import com.cisco.pxgrid.model.AccountState;
import com.cisco.pxgrid.model.Authorization;
import com.cisco.pxgrid.model.AuthorizationRequest;
import com.cisco.pxgrid.model.AuthorizationResponse;
import com.cisco.pxgrid.model.Service;
import com.cisco.pxgrid.model.ServiceLookupRequest;
import com.cisco.pxgrid.model.ServiceLookupResponse;
import com.cisco.pxgrid.model.ServiceRegisterRequest;
import com.cisco.pxgrid.model.ServiceRegisterResponse;
import com.google.gson.Gson;
/**
* Using HTTPS for pxGrid control
*/
public class PxgridControl {
       private static Logger logger = LoggerFactory.getLogger(PxgridControl.class);
       private SampleConfiguration config;
    private String controllerVersion;
    public PxgridControl(SampleConfiguration config) {
       this.config = config;
       }
       private <T> T sendRequest(HttpsURLConnection https, Object request, Class<T> responseClass) throws
IOException {
               https.setRequestProperty("Content-Type", "application/json");
https.setRequestProperty("Accept", "application/json");
               Gson gson = new Gson();
               OutputStreamWriter out = new OutputStreamWriter(https.getOutputStream());
               logger.info("Request={}", gson.toJson(request));
               gson.toJson(request, out);
               out.flush();
       InputStreamReader in = new InputStreamReader(https.getInputStream());
       T response = gson.fromJson(in, responseClass);
               logger.info("Response={}", gson.toJson(response));
       return response;
       }
   private HttpsURLConnection getHttpsURLConnection(String urlSuffix) throws IOException {
               String url = "https://" + config.getHostnames()[0] + ":8910/pxgrid/control/" + urlSuffix;
               URL conn = new URL(url);
               HttpsURLConnection https = (HttpsURLConnection) conn.openConnection();
               // SSL and Auth
               https.setSSLSocketFactory(config.getSSLContext().getSocketFactory());
               https.setRequestMethod("POST");
               String userPassword = config.getUserName() + ":" + config.getPassword();
               String encoded = Base64.getEncoder().encodeToString(userPassword.getBytes());
               https.setRequestProperty("Authorization", "Basic " + encoded);
               https.setHostnameVerifier(new HostnameVerifier() {
                       @Override
                       public boolean verify(String hostname, SSLSession session) {
                               return true;
                       }
               });
               https.setDoInput(true);
               https.setDoOutput(true);
               return https;
    3
```



```
/**
     * Create new account
     * @return password
   public String accountCreate() throws IOException {
       HttpsURLConnection https = getHttpsURLConnection("AccountCreate");
               AccountCreateRequest request = new AccountCreateRequest();
               request.setNodeName(config.getUserName());
               AccountCreateResponse response = sendRequest(https, request, AccountCreateResponse.class);
               return response.getPassword();
   }
   public AccountState accountActivate() throws IOException {
               HttpsURLConnection https = getHttpsURLConnection("AccountActivate");
               AccountActivateRequest request = new AccountActivateRequest();
               request.setDescription(config.getDescription());
               AccountActivateResponse response = sendRequest(https, request, AccountActivateResponse.class);
               controllerVersion = response.getVersion();
               return response.getAccountState();
   }
       public void registerService(String name, Map<String, String> properties) throws IOException {
               HttpsURLConnection https = getHttpsURLConnection("ServiceRegister");
               ServiceRegisterRequest request = new ServiceRegisterRequest();
               request.setName(name);
               request.setProperties(properties);
               sendRequest(https, request, ServiceRegisterResponse.class);
       }
       public Service[] lookupService(String name) throws IOException {
               HttpsURLConnection https = getHttpsURLConnection("ServiceLookup");
               ServiceLookupRequest request = new ServiceLookupRequest();
               request.setName(name);
               ServiceLookupResponse response = sendRequest(https, request, ServiceLookupResponse.class);
               return response.getServices();
       }
       public String getAccessSecret(String peerNodeName) throws IOException
               HttpsURLConnection https = getHttpsURLConnection("AccessSecret");
               AccessSecretRequest request = new AccessSecretRequest();
               request.setPeerNodeName(peerNodeName);
               AccessSecretResponse response = sendRequest(https, request, AccessSecretResponse.class);
               return response.getSecret();
       }
       public boolean isAuthorized(String requestNodeName, String serviceName, String operation) throws
IOException {
               HttpsURLConnection https = getHttpsURLConnection("Authorization");
               AuthorizationRequest request = new AuthorizationRequest();
               request.setRequestNodeName(requestNodeName);
               request.setServiceName(serviceName);
               request.setServiceOperation(operation);
               AuthorizationResponse response = sendRequest(https, request, AuthorizationResponse.class);
               return (response.getAuthorization() == Authorization.PERMIT);
       }
       public String getControllerVersion() {
               return controllerVersion;
       }
```

### **SampleHelper**

This code provides the initial WebSockets connection.

cisco.

```
package com.cisco.pxgrid.samples.ise;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;
import java.nio.charset.StandardCharsets;
import java.time.OffsetDateTime;
import java.time.format.DateTimeFormatter;
import java.util.Base64;
import java.util.Scanner;
import javax.net.ssl.HttpsURLConnection;
import javax.net.ssl.SSLSocketFactory;
import org.apache.commons.io.IOUtils;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
import com.google.gson.TypeAdapter;
import com.google.gson.stream.JsonReader;
import com.google.gson.stream.JsonToken;
import com.google.gson.stream.JsonWriter;
public class SampleHelper {
       private static Logger logger = LoggerFactory.getLogger(SampleHelper.class);
       public static HttpsURLConnection createHttpsURLConnection(String url, String user, String password,
                      SSLSocketFactory sslSocketFactory) throws IOException {
               URL conn = new URL(url);
               HttpsURLConnection https = (HttpsURLConnection) conn.openConnection();
               https.setSSLSocketFactory(sslSocketFactory);
               String userPassword = user + ":" + password;
               String encoded = Base64.getEncoder().encodeToString(userPassword.getBytes());
               https.setRequestProperty("Authorization", "Basic " + encoded);
               return https;
       }
       public static String prompt(String msg) {
               System.out.println(msg);
               @SuppressWarnings("resource")
               Scanner scanner = new Scanner(System.in);
               String value = scanner.nextLine();
                     '.equals(value))
               if ("'
                      return null;
               return value;
       }
       public static OffsetDateTime promptDate(String msg) {
               String value = prompt(msg);
               if (value == null) return null;
               return OffsetDateTime.parse(value);
       }
       public static void postObjectAndPrint(String url, String user, String password, SSLSocketFactory ssl,
                      Object postObject) throws IOException {
               Gson gson = new GsonBuilder().registerTypeAdapter(OffsetDateTime.class, new
OffsetDateTimeAdapter()).create();
               postStringAndPrint(url, user, password, ssl, gson.toJson(postObject));
       }
       public static void postStringAndPrint(String url, String user, String password, SSLSocketFactory ssl,
                      String postData) throws IOException {
               logger.info("postData={}", postData);
               HttpsURLConnection httpsConn = SampleHelper.createHttpsURLConnection(url, user, password,
ssl);
               httpsConn.setRequestMethod("POST");
               httpsConn.setRequestProperty("Content-Type", "application/json");
               httpsConn.setRequestProperty("Accept", "application/json");
```

# cisco.

```
httpsConn.setDoInput(true);
       httpsConn.setDoOutput(true);
       OutputStreamWriter osw = new OutputStreamWriter(httpsConn.getOutputStream());
       osw.write(postData);
       osw.flush();
       int status = httpsConn.getResponseCode();
       logger.info("Response status={}", status);
       if (status < HttpURLConnection.HTTP_BAD_REQUEST) {</pre>
               try (InputStream in = httpsConn.getInputStream()) {
                       String content = IOUtils.toString(in, StandardCharsets.UTF_8);
                       System.out.println("Content: " + content);
       } else
                   (InputStream in = httpsConn.getErrorStream()) {
                       String content = IOUtils.toString(in, StandardCharsets.UTF_8);
System.out.println("Content: " + content);
               }
       }
}
private static class OffsetDateTimeAdapter extends TypeAdapter<OffsetDateTime> {
       DateTimeFormatter formatter = DateTimeFormatter.ISO_OFFSET_DATE_TIME;
       @Override
       public void write(JsonWriter out, OffsetDateTime value) throws IOException {
               if (value == null) {
                       out.nullValue();
                       return;
               }
               out.value(formatter.format(value));
       }
       @Override
       public OffsetDateTime read(JsonReader in) throws IOException {
               if (in.peek() == JsonToken.NULL) {
                       in.nextNull();
                       return null;
               }
               return formatter.parse(in.nextString(), OffsetDateTime::from);
       }
}
```

### **StompFrame**

This code represents the parsing of STOMP frames.

```
package com.cisco.pxgrid.samples.ise;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.text.ParseException;
import java.util.HashMap;
import java.util.Map;
/**
\ast This follows STOMP 1.2 specification to parse and generate STOMP frames:
*
    https://stomp.github.io/stomp-specification-1.2.html
* This single class is self-sufficient handle all STOMP frames.
 * Note for WebSocket:
* If input comes as WebSocket text type, (WS RFC says Text is UTF-8)
 *
  server side handling code like Spring TextMessage may convert the bytes to String as UTF-8
```

```
cisco.
```

```
\ast which maybe the wrong encoding as STOMP frame itself can use other encoding.
    e.g. A particular encoding may have bytes: FF FF... 10, that is completely out
of range for Unicode.
* Unless STOMP body is also UTF-8, STOMP frame must be sent as binary
* @author Alan Lei
*/
public class StompFrame {
       public enum Command {
               CONNECT, STOMP, CONNECTED, SEND, SUBSCRIBE, UNSUBSCRIBE, ACK, NACK,
               BEGIN, COMMIT, ABORT, DISCONNECT, MESSAGE, RECEIPT, ERROR;
               private static Map<String, Command> mapOfStringToCommand = new HashMap<>();
               static {
                       for (Command command : Command.values()) {
                              mapOfStringToCommand.put(command.name(), command);
                       }
               }
               public static Command get(String value) {
                       return mapOfStringToCommand.get(value);
               }
       }
       private Command command;
       private Map<String, String> headers = new HashMap<>();
       private byte[] content;
       private final static int MAX_BUFFER_SIZE = 1024;
       public Command getCommand() {
               return command;
       }
       public void setCommand(Command command) {
               this.command = command;
       }
       public String getHeader(String name) {
               return headers.get(name);
       }
       public void setHeader(String name, String value) {
               headers.put(name, value);
        }
       public Map<String, String> getHeaders() {
               return headers;
       }
       public byte[] getContent() {
               return content;
       }
       public void setContent(byte[] content) {
               this.content = content;
        }
       public void write(OutputStream out) throws IOException {
               out.write(command.name().getBytes());
               out.write('\n');
               for (String name : headers.keySet()) {
                       out.write(name.getBytes());
                       out.write(':');
                       out.write(headers.get(name).getBytes());
                       out.write('\n');
               }
               out.write('\n');
               if (content != null) {
                       out.write(content);
               }
               out.write(0);
       }
```

# cisco.

```
private static String readLine(InputStream in) throws IOException, ParseException {
       byte[] line = new byte[MAX_BUFFER_SIZE];
       int index = 0;
       while (index < MAX_BUFFER_SIZE) {</pre>
               int b = in.read();
               if (b != -1) {
                       if (b == '\n') {
                              return new String(line, 0, index);
                       }
                       if (b != '\r') {
                              line[index] = (byte)b;
                              index++;
                       }
               }
               else {
                       // No line found
                       return null;
               }
       throw new ParseException("Line too long", MAX BUFFER SIZE);
}
/*
* Using InputStream instead of Reader because
* content-length is octet count instead of character count
*/
public static StompFrame parse(InputStream reader) throws IOException, ParseException {
       StompFrame stomp = new StompFrame();
       // Read Command
       String line = readLine(reader);
       Command command = Command.get(line);
       if (command == null) throw new ParseException("Unknown command: " + line, 0);
       stomp.setCommand(command);
       // Read Headers
       int contentLength = -1;
       while ((line = readLine(reader)) != null) {
               if (line.equals("")) break;
               int colon = line.indexOf(':');
               String name = line.substring(0, colon);
               String value = line.substring(colon + 1);
               stomp.setHeader(name, value);
               if (name.equals("content-length")) {
                       contentLength = Integer.parseInt(value);
               }
       }
       // Read Content
       if (contentLength != -1) {
               // content-length is in octets
               byte[] content = new byte[contentLength];
               reader.read(content);
               stomp.setContent(content);
               if (reader.read() != 0) {
                       throw new ParseException("Byte after STOMP Body not NULL", -1);
               }
       }
       else {
               // No content-length. Look for ending NULL byte.
               byte[] buffer = new byte[MAX BUFFER SIZE];
               int length = 0;
               while (length < MAX_BUFFER_SIZE) {</pre>
                       int b = reader.read();
                       if (b == -1) {
                              throw new ParseException("Premature end of stream", -1);
                       }
                       if (b == 0) {
                              if (length > 0) {
                                      byte[] content = new byte[length];
                                      System.arraycopy(buffer, 0, content, 0, length);
                                      stomp.setContent(content);
                              }
```

```
// More EOLs may follow, but ignored.
                                    return stomp;
                           }
                           buffer[length] = (byte)b;
                           length++;
                  }
                  throw new ParseException("Frame too long", -1);
         }
         return stomp;
}
@Override
public String toString() {
         StringBuilder sb = new StringBuilder();
        sb.append("command=" + command);
sb.append(", headers={");
         for (String name : headers.keySet()) {
                  sb.append("'" + name + "':");
sb.append("'" + headers.get(name) + "',");
         3
        sb.append("}");
sb.append(", content.length=" + content.length);
         return sb.toString();
}
```

### **StompSubscription**

This code provides service or topic subscription using the STOMP messaging protocol

```
package com.cisco.pxgrid.samples.ise;
import java.util.concurrent.atomic.AtomicInteger;
public class StompSubscription {
       public static interface Handler {
                void handle(StompFrame message);
        }
       private static AtomicInteger currentSubscriptionID = new AtomicInteger();
       private String id = Integer.toString(currentSubscriptionID.getAndIncrement());
       private String topic;
       private Handler handler;
       public StompSubscription(String topic, Handler handler) {
               this.topic = topic;
this.handler = handler;
        }
       public String getId() {
               return id;
        }
       public String getTopic() {
               return topic;
        }
       public Handler getHandler() {
                return handler;
        }
        public StompFrame getSubscribeMessage() {
                StompFrame message = new StompFrame();
                message.setCommand(StompFrame.Command.SUBSCRIBE);
               message.setHeader("destination", topic);
message.setHeader("id", id);
                return message;
        }
```

cisco.
**StompPubSubClientEndpoint** 

This code provides service or topic subscription using the STOMP messaging protocol for the client endpoint or asset device.

```
package com.cisco.pxgrid.samples.ise;
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.nio.ByteBuffer;
import java.text.ParseException;
import java.util.Map;
import java.util.concurrent.ConcurrentHashMap;
import javax.websocket.ClientEndpoint;
import javax.websocket.CloseReason;
import javax.websocket.Endpoint;
import javax.websocket.EndpointConfig;
import javax.websocket.MessageHandler;
import javax.websocket.Session;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import com.cisco.pxgrid.samples.ise.StompSubscription.Handler;
@ClientEndpoint
public class StompPubsubClientEndpoint extends Endpoint {
       private static Logger logger = LoggerFactory.getLogger(StompPubsubClientEndpoint.class);
       private volatile Session session;
       private Map<String, StompSubscription> mapOfIdToSubscription = new ConcurrentHashMap<>();
        public void connect(String hostname) throws IOException {
               logger.info("STOMP CONNECT host={}", hostname);
               StompFrame message = new StompFrame();
               message.setCommand(StompFrame.Command.CONNECT);
               message.setHeader("accept-version", "1.2");
message.setHeader("host", hostname);
               send(message);
        }
       public void disconnect(String receipt) throws IOException {
               logger.info("STOMP DISCONNECT receipt={}", receipt);
               StompFrame message = new StompFrame();
               message.setCommand(StompFrame.Command.DISCONNECT);
               if (receipt != null) {
                       message.setHeader("receipt", receipt);
               }
               send(message);
        }
       public void subscribe(StompSubscription subscription) throws IOException {
               logger.info("STOMP SUBSCRIBE topic={}", subscription.getTopic());
               mapOfIdToSubscription.put(subscription.getId(), subscription);
               if (session != null) {
                       StompFrame message = subscription.getSubscribeMessage();
                       send(message);
               }
        }
        public void publish(String topic, byte[] content) throws IOException {
               logger.info("STOMP SEND topic={}", topic);
               StompFrame message = new StompFrame();
```



```
message.setCommand(StompFrame.Command.SEND);
               message.setHeader("destination", topic);
               message.setHeader("content-length", Integer.toString(content.length));
               message.setContent(content);
               send(message);
       }
       private void send(StompFrame message) throws IOException {
               if (session != null) {
                      ByteArrayOutputStream baos = new ByteArrayOutputStream();
                       message.write(baos);
                       // Send as binary
                       session.getBasicRemote().sendBinary(ByteBuffer.wrap(baos.toByteArray()));
               }
       }
       public void waitForOpen() throws InterruptedException {
               synchronized (this) {
                      while (session == null) {
                              this.wait();
                      }
               }
       }
       private void onStompMessage(StompFrame stomp) {
               switch (stomp.getCommand()) {
               case CONNECTED:
                      String version = stomp.getHeader("version");
                       logger.info("STOMP CONNECTED version={}", version);
                      break:
               case RECEIPT:
                      String receiptId = stomp.getHeader("receipt-id");
                       logger.info("STOMP RECEIPT id={}", receiptId);
                      break;
               case MESSAGE:
                      String id = stomp.getHeader("subscription");
                       StompSubscription subscription = mapOfIdToSubscription.get(id);
                       Handler handler = subscription.getHandler();
                       if (handler != null) {
                              handler.handle(stomp);
                       }
                      break;
               case ERROR:
                       // Server will close connect on ERROR according to STOMP specification
                      logger.info("STOMP ERROR stomp={}", stomp);
                      break:
               default:
                       // Ignore others
                      break;
               }
       }
       private class TextHandler implements MessageHandler.Whole<String> {
               @Override
               public void onMessage(String message) {
                      try {
                              StompFrame stomp = StompFrame.parse(new
ByteArrayInputStream(message.getBytes()));
                              onStompMessage(stomp);
                       } catch (IOException | ParseException e) {
                              logger.error("onMessage", e);
                      }
               }
       }
       private class BinaryHandler implements MessageHandler.Whole<InputStream> {
               @Override
               public void onMessage(InputStream in) {
                       try {
                              StompFrame stomp = StompFrame.parse(in);
                              onStompMessage(stomp);
                       } catch (IOException | ParseException e) {
                              logger.error("onMessage", e);
```



```
}
      }
}
@Override
public void onOpen(Session session, EndpointConfig cfg) {
      logger.info("WS onOpen");
      this.session = session;
      try {
             session.addMessageHandler(new TextHandler());
             session.addMessageHandler(new BinaryHandler());
             for (StompSubscription subscription : mapOfIdToSubscription.values()) {
                    StompFrame message = subscription.getSubscribeMessage();
                    send(message);
             }
      } catch (IOException e) {
             logger.error("onOpen", e);
      }
      synchronized (this) {
             this.notifyAll();
      }
}
@Override
closeReason.getReasonPhrase());
      this.session = null;
}
@Override
public void onError(Session session, Throwable thr) {
      logger.info("WS onError thr={}", thr.getMessage());
      this.session = null;
}
```

### **SessionQueryAll**

This code returns all available sessions

```
package com.cisco.pxgrid.samples.ise;
import java.time.OffsetDateTime;
import org.apache.commons.cli.ParseException;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import com.cisco.pxgrid.samples.ise.model.AccountState;
import com.cisco.pxgrid.samples.ise.model.Service;
/**
* Demonstrates how to query all sessions from ISE Session Directory service
*/
public class SessionQueryAll {
       private static Logger logger = LoggerFactory.getLogger(SessionQueryAll.class);
       private static class SessionQueryRequest {
               OffsetDateTime startTimestamp;
       }
       private static void downloadUsingAccessSecret(SampleConfiguration config) throws Exception {
               OffsetDateTime startTimestamp = SampleHelper.promptDate("Enter start time (ex. '2015-01-
31T13:00:00-07:00' or <enter> for no start time): ");
               PxgridControl https = new PxgridControl(config);
               // pxGrid ServiceLookup for session service
               Service[] services = https.serviceLookup("com.cisco.ise.session");
               if (services == null || services.length == 0) {
                      logger.warn("Service unavailabe");
                       return;
               }
               // Use first service
               Service service = services[0];
               String url = service.getProperties().get("restBaseUrl") + "/getSessions";
               logger.info("url={}", url);
               // pxGrid AccesssSecret for the node
               String secret = https.getAccessSecret(service.getNodeName());
               SessionQueryRequest request = new SessionQueryRequest();
               request.startTimestamp = startTimestamp;
               SampleHelper.postObjectAndPrint(url, config.getNodeName(), secret,
config.getSSLContext().getSocketFactory(), request);
       }
       public static void main(String [] args) throws Exception {
               // Parse arguments
               SampleConfiguration config = new SampleConfiguration();
               try {
                      config.parse(args);
               } catch (ParseException e) {
                      config.printHelp("SessionQueryAll");
                      System.exit(1);
               }
               // AccountActivate
               PxgridControl control = new PxgridControl(config);
               while (control.accountActivate() != AccountState.ENABLED)
                      Thread.sleep(60000);
               logger.info("pxGrid controller version={}", control.getControllerVersion());
               downloadUsingAccessSecret(config);
```

### }



### **SessionQueryByIP**

This code returns available session attributes for a queried IP Address

```
package com.cisco.pxgrid.samples.ise;
import java.io.IOException;
import org.apache.commons.cli.ParseException;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import com.cisco.pxgrid.samples.ise.model.AccountState;
import com.cisco.pxgrid.samples.ise.model.Service;
/**
* Demonstrates how to query session by IP from ISE Session Directory service
*/
public class SessionQueryByIP {
       private static Logger logger = LoggerFactory.getLogger(SessionQueryByIP.class);
       private static void query(SampleConfiguration config, String ip) throws IOException {
               PxgridControl pxgrid = new PxgridControl(config);
               // pxGrid ServiceLookup for session service
               Service[] services = pxgrid.serviceLookup("com.cisco.ise.session");
               if (services == null || services.length == 0) {
                       System.out.println("Service unavailabe");
                       return:
               }
               // Use first service
               Service service = services[0];
               String url = service.getProperties().get("restBaseUrl") + "/getSessionByIpAddress";
               logger.info("url={}", url);
               // pxGrid AccessSecret for the node
               String secret = pxgrid.getAccessSecret(service.getNodeName());
               String postData = "{\"ipAddress\":\"" + ip + "\"}";
SampleHelper.postStringAndPrint(url, config.getNodeName(), secret,
config.getSSLContext().getSocketFactory(), postData);
       ł
       public static void main(String [] args) throws Exception {
               // Parse arguments
               SampleConfiguration config = new SampleConfiguration();
               try {
                       config.parse(args);
               } catch (ParseException e) {
                       config.printHelp("SessionQueryByIP");
                       System.exit(1);
               }
               // AccountActivate
               PxgridControl pxgrid = new PxgridControl(config);
               while (pxgrid.accountActivate() != AccountState.ENABLED)
                       Thread.sleep(60000);
               logger.info("pxGrid controller version={}", pxgrid.getControllerVersion());
               while (true) {
                       String ip = SampleHelper.prompt("IP address (or <enter> to disconnect): ");
                       if (ip == null)
                                              break;
                       query(config, ip);
               }
```

#### uluili cisco

}

### **CustomServiceProvider**

This code provides the pxGrid client to a publish a topic and the CustomServiceConsumer code will subscribe to this topic or service.

```
package com.cisco.pxgrid.samples.ise;
import java.io.IOException;
import java.net.URI;
import java.util.HashMap;
import java.util.Map;
import java.util.concurrent.Executors;
import java.util.concurrent.ScheduledExecutorService;
import java.util.concurrent.TimeUnit;
import javax.websocket.Session;
import org.apache.commons.cli.ParseException;
import org.glassfish.tyrus.client.ClientManager;
import org.glassfish.tyrus.client.ClientProperties;
import org.glassfish.tyrus.client.SslEngineConfigurator;
import org.glassfish.tyrus.client.auth.Credentials;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import com.cisco.pxgrid.samples.ise.model.AccountState;
import com.cisco.pxgrid.samples.ise.model.Service;
import com.cisco.pxgrid.samples.ise.model.ServiceRegisterResponse;
/**
 * Demonstrate how to create a custom service that publishes data
* The flow of the application is as follows:
 * 1. Parse arguments for configurations
 * 2. Activate Account. This will then require ISE Admin to approve this new node.
 * 3. pxGrid ServiceRegister to register the new custom service
 * 4. Schedule periodic pxGrid ServiceReregister to signify the service is still alive
 * 5. pxGrid ServiceLookup for ISE pubsub service
 * 6. pxGrid get AccessSecret for the ISE pubsub node
 * 7. Establish WebSocket connection with the ISE pubsub node
 * 8. Establish STOMP connection for pubsub messaging
* 9. Schedule periodic publish of data
 * 10. Wait for keyboard input for stopping the application
*/
public class CustomServiceProvider {
       private static Logger logger = LoggerFactory.getLogger(CustomServiceProvider.class);
       public static void main(String[] args) throws Exception {
               ScheduledExecutorService executor = Executors.newSingleThreadScheduledExecutor();
               // Parse arguments
               SampleConfiguration config = new SampleConfiguration();
               try {
                       config.parse(args);
               } catch (ParseException e) {
                       config.printHelp("CustomServiceProvider");
                       System.exit(1);
               3
               // AccountActivate
               PxgridControl control = new PxgridControl(config);
               while (control.accountActivate() != AccountState.ENABLED) {
                       Thread.sleep(60000);
               }
```

# cisco.

```
logger.info("pxGrid controller version={}", control.getControllerVersion());
               // pxGrid ServiceRegister
               Map<String, String> sessionProperties = new HashMap<>();
               sessionProperties.put("wsPubsubService", "com.cisco.ise.pubsub");
               sessionProperties.put("customTopic", "/topic/com.example.custom");
               ServiceRegisterResponse response = control.serviceRegister("com.example.custom",
sessionProperties);
               String registrationId = response.getId();
               long reregisterTimeMillis = response.getReregisterTimeMillis();
               // Schedule pxGrid ServiceReregister
               executor.scheduleWithFixedDelay(() -> {
                      try {
                              control.serviceReregister(registrationId);
                      } catch (IOException e) {
                              logger.error("Reregister failure");
                      }
               }, reregisterTimeMillis, reregisterTimeMillis, TimeUnit.MILLISECONDS);
               // pxGrid ServiceLookup for pubsub service
               Service[] services = control.serviceLookup("com.cisco.ise.pubsub");
               if (services.length == 0) {
                      logger.info("Pubsub service unavailabe");
                      return;
               }
// Use first service
               Service wsPubsubService = services[0];
               String wsURL = wsPubsubService.getProperties().get("wsUrl");
               logger.info("wsUrl={}", wsURL);
               // pxGrid AccessSecret
               String secret = control.getAccessSecret(wsPubsubService.getNodeName());
               // Setup WebSocket client
               ClientManager client = ClientManager.createClient();
               SslEngineConfigurator sslEngineConfigurator = new
SslEngineConfigurator(config.getSSLContext());
               client.getProperties().put(ClientProperties.SSL_ENGINE_CONFIGURATOR, sslEngineConfigurator);
               client.getProperties().put(ClientProperties.CREDENTIALS,
                              new Credentials(config.getNodeName(), secret.getBytes()));
               // WebSocket connect
               StompPubsubClientEndpoint endpoint = new StompPubsubClientEndpoint();
               URI uri = new URI(wsURL);
               Session session = client.connectToServer(endpoint, uri);
               // STOMP connect
               endpoint.connect(uri.getHost());
               // STOMP send periodically
               executor.scheduleWithFixedDelay(() -> {
                      try {
                              endpoint.publish("/topic/com.example.custom", "custom data".getBytes());
                      } catch (IOException e) {
                              logger.error("Publish failure");
               }, 0, 5, TimeUnit.SECONDS);
               SampleHelper.prompt("press <enter> to disconnect...");
               // pxGrid ServerUnregister
               control.unregisterService(registrationId);
               // Stop executor
               executor.shutdown();
               executor.awaitTermination(5, TimeUnit.SECONDS);
               // STOMP disconnect
               endpoint.disconnect("ID-123");
               // Wait for disconnect receipt
               Thread.sleep(3000);
```

```
}
```

// Websocket close
session.close();

### CustomServiceConsumer

This sample code provides consumer or pxGrid client subscription to a customer service or published topic

```
package com.cisco.pxgrid.samples.ise;
import java.net.URI;
import javax.net.ssl.SSLSession;
import javax.websocket.Session;
import org.apache.commons.cli.ParseException;
import org.glassfish.tyrus.client.ClientManager;
import org.glassfish.tyrus.client.ClientProperties;
import org.glassfish.tyrus.client.SslEngineConfigurator;
import org.glassfish.tyrus.client.auth.Credentials;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import com.cisco.pxgrid.samples.ise.model.AccountState;
import com.cisco.pxgrid.samples.ise.model.Service;
/**
* Demonstrates how to subscribe a topic from a custom service
* The flow of the application is as follows:
 * 1. Parse arguments for configurations
 * 2. Activate Account. This will then require ISE Admin to approve this new node.
 * 3. pxGrid ServiceLookup for the custom service
 * 4. pxGrid ServiceLookup for ISE pubsub service
 * 5. pxGrid get AccessSecret for the ISE pubsub node
 * 6. Establish WebSocket connection with the ISE pubsub node
 * 7. Establish STOMP connection for pubsub messaging
 * 8. Subscribe to the topic in the custom service
 *
  9. Wait for keyboard input for stopping the application
*/
public class CustomServiceConsumer {
       private static Logger logger = LoggerFactory.getLogger(CustomServiceProvider.class);
       // Subscribe handler class
       private static class MessageHandler implements StompSubscription.Handler {
               @Override
               public void handle(StompFrame message) {
                      System.out.println(new String(message.getContent()));
               }
       }
       public static void main(String [] args) throws Exception {
               // Parse arguments
               SampleConfiguration config = new SampleConfiguration();
               try {
                       config.parse(args);
               } catch (ParseException e) {
                       config.printHelp("CustomServiceConsumer");
                       System.exit(1):
               }
               // AccountActivate
               PxgridControl control = new PxgridControl(config);
               while (control.accountActivate() != AccountState.ENABLED) {
                       Thread.sleep(60000);
               }
               logger.info("pxGrid controller version={}", control.getControllerVersion());
```



```
// pxGrid ServiceLookup for custom service
               Service[] services = control.serviceLookup("com.example.custom");
               if (services.length == 0) {
                      logger.info("Service unavailabe");
                      return;
               }
               // Use first service. Note that ServiceLookup randomize ordering of services
               Service customService = services[0];
               String wsPubsubServiceName = customService.getProperties().get("wsPubsubService");
               String customTopic = customService.getProperties().get("customTopic");
               logger.info("wsPubsubServiceName={} sessionTopic={}", wsPubsubServiceName, customTopic);
               // pxGrid ServiceLookup for pubsub service
               services = control.serviceLookup(wsPubsubServiceName);
               if (services.length == 0) {
                      logger.info("Pubsub service unavailabe");
                      return:
               }
               // Use first service
               Service wsPubsubService = services[0];
               String wsURL = wsPubsubService.getProperties().get("wsUrl");
               logger.info("wsUrl={}", wsURL);
               // pxGrid AccessSecret for the pubsub node
               String secret = control.getAccessSecret(wsPubsubService.getNodeName());
               // WebSocket config
               ClientManager client = ClientManager.createClient();
               SslEngineConfigurator sslEngineConfigurator = new
SslEngineConfigurator(config.getSSLContext());
               client.getProperties().put(ClientProperties.SSL_ENGINE_CONFIGURATOR, sslEngineConfigurator);
               client.getProperties().put(ClientProperties.CREDENTIALS,
                              new Credentials(config.getNodeName(), secret.getBytes()));
               // WebSocket connect
               StompPubsubClientEndpoint endpoint = new StompPubsubClientEndpoint();
               URI uri = new URI(wsURL);
               Session session = client.connectToServer(endpoint, uri);
               // STOMP connect
               endpoint.connect(uri.getHost());
               // Subscribe
               StompSubscription subscription = new StompSubscription(customTopic, new MessageHandler());
               endpoint.subscribe(subscription);
               SampleHelper.prompt("press <enter> to disconnect...");
               // STOMP disconnect
               endpoint.disconnect("ID-123");
               // Wait for disconnect receipt
               Thread.sleep(3000);
               session.close();
       }
```

# **Adaptive Network Control (ANC) Examples**

The following coding examples represent the ANC operation that can be taken by the pxGrid client. The complete ANC configuration topic can be found: <u>https://github.com/cisco-pxgrid/pxgrid-rest-ws/wiki/ANC-configuration</u>

### ANCSubscribe

The consumer or pxGrid client subscribes to the /topic/com.cisco.ise.config.anc.status topic or service

```
package com.cisco.pxgrid.samples.ise.anc;
import java.net.URI;
import org.glassfish.grizzly.ssl.SSLEngineConfigurator;
import org.glassfish.tyrus.client.ClientManager;
import org.glassfish.tyrus.client.ClientProperties;
import org.glassfish.tyrus.client.auth.Credentials;
import com.cisco.pxgrid.model.AccountState;
import com.cisco.pxgrid.model.Service;
import com.cisco.pxgrid.samples.ise.http.Console;
import com.cisco.pxgrid.samples.ise.http.PxgridControl;
import com.cisco.pxgrid.samples.ise.http.SampleConfiguration;
import com.cisco.pxgrid.samples.ise.http.StompFrame;
import com.cisco.pxgrid.samples.ise.http.StompPubsubClientEndpoint;
import com.cisco.pxgrid.samples.ise.http.StompSubscription;
/**
* Demonstrates how to subscribe using REST/WS
* /
public class AncSubscribe {
       // Subscribe handler class
       private static class SubscriptionHandler implements StompSubscription.Handler {
               @Override
               public void handle(StompFrame message) {
                      System.out.println(new String(message.getContent()));
               }
       }
       public static void main(String [] args) throws Exception {
               // Read environment for config
               SampleConfiguration config = new SampleConfiguration();
               PxgridControl control = new PxgridControl(config);
               // AccountActivate
               while (control.accountActivate() != AccountState.ENABLED) {
                      Thread.sleep(60000);
               Console.log("pxGrid controller version=" + control.getControllerVersion());
               // Session ServiceLookup
               Console.log("Looking up service com.cisco.ise.config.anc");
               Service[] services = control.lookupService("com.cisco.ise.config.anc");
               if (services.length == 0) {
                      Console.log("Session service unavailabe");
                      return;
               }
               Service sessionService = services[0];
               String wsPubsubServiceName = sessionService.getProperties().get("wsPubsubService");
               String statusTopic = sessionService.getProperties().get("statusTopic");
               Console.log("wsPubsubServiceName=" + wsPubsubServiceName + " statusTopic=" + statusTopic);
               // Pubsub ServiceLookup
               services = control.lookupService(wsPubsubServiceName);
               if (services.length == 0) {
```



```
Console.log("Pubsub service unavailabe");
                      return;
               }
               // Select first one for sample purpose. Should cycle through until connects.
               Service wsPubsubService = services[0];
               String wsURL = wsPubsubService.getProperties().get("wsUrl");
               Console.log("url=" + wsURL);
               // pxGrid AccessSecret
               String secret = control.getAccessSecret(wsPubsubService.getNodeName());
               // WebSocket config
               ClientManager client = ClientManager.createClient();
               SSLEngineConfigurator sslEngineConfigurator = new
SSLEngineConfigurator(config.getSSLContext());
               client.getProperties().put(ClientProperties.SSL_ENGINE_CONFIGURATOR, sslEngineConfigurator);
               client.getProperties().put(ClientProperties.CREDENTIALS,
                              new Credentials(config.getUserName(), secret.getBytes()));
               // WebSocket connect
               StompPubsubClientEndpoint endpoint = new StompPubsubClientEndpoint();
               URI uri = new URI(wsURL);
               javax.websocket.Session session = client.connectToServer(endpoint, uri);
               // STOMP connect
               endpoint.connect(uri.getHost());
               // Subscribe
               StompSubscription subscription = new StompSubscription(statusTopic, new
SubscriptionHandler());
               endpoint.subscribe(subscription);
               Console.log("press <enter> to disconnect...");
               System.in.read();
               // STOMP disconnect
               endpoint.disconnect("ID-123");
               // Wait for disconnect receipt
               Thread.sleep(3000);
               session.close();
       }
```

### **ANCGetPolicies**

This code retrieves all ISE ANC Policies

```
package com.cisco.pxgrid.samples.ise.anc;
import java.io.IOException;
import com.cisco.pxgrid.model.AccountState;
import com.cisco.pxgrid.model.Service;
import com.cisco.pxgrid.samples.ise.http.Console;
import com.cisco.pxgrid.samples.ise.http.PxgridControl;
import com.cisco.pxgrid.samples.ise.http.SampleConfiguration;
import com.cisco.pxgrid.samples.ise.http.SampleHelper;
/***
 * Demonstrates how to query a session using IP address
 */
public class AncGetPolicies {
    private static void get(SampleConfiguration config) throws IOException {
        PxgridControl pxgrid = new PxgridControl(config);
        Service[] services = pxgrid.lookupService("com.cisco.ise.config.anc");
```





### **ANCGetPoliciesByName**

This code retrieves the ISE ANC policies by policy name

```
package com.cisco.pxgrid.samples.ise.anc;
import java.io.IOException;
import com.cisco.pxgrid.model.AccountState;
import com.cisco.pxgrid.model.Service;
import com.cisco.pxgrid.samples.ise.http.Console;
import com.cisco.pxgrid.samples.ise.http.PxgridControl;
import com.cisco.pxgrid.samples.ise.http.SampleConfiguration;
import com.cisco.pxgrid.samples.ise.http.SampleHelper;
/**
* Demonstrates how to query a session using IP address
*/
public class AncGetPolicyByName {
       private static class ByNameRequest {
               private String name;
               public String getName() {
                      return name;
               public void setName(String name) {
                      this.name = name;
               }
       private static void get(SampleConfiguration config, String name) throws IOException {
               PxgridControl pxgrid = new PxgridControl(config);
               Service[] services = pxgrid.lookupService("com.cisco.ise.config.anc");
               if (services == null || services.length == 0) {
                      System.out.println("Service unavailabe");
                      return;
               }
               Service service = services[0];
               String url = service.getProperties().get("restBaseUrl") + "/getPolicyByName";
               String secret = pxgrid.getAccessSecret(service.getNodeName());
               ByNameRequest request = new ByNameRequest();
               request.setName(name);
               SampleHelper.postObjectAndPrint(url, config.getUserName(), secret,
config.getSSLContext().getSocketFactory(), request);
       }
       public static void main(String [] args) throws Exception {
```



```
SampleConfiguration config = new SampleConfiguration();
PxgridControl pxgrid = new PxgridControl(config);
while (pxgrid.accountActivate() != AccountState.ENABLED)
            Thread.sleep(60000);
Console.log("pxGrid controller version=" + pxgrid.getControllerVersion());
String name = SampleHelper.prompt("Get policy name: ");
get(config, name);
}
```

### **ANCGetEndpoints**

This code retrieves endpoints with the ANC policy applied

```
package com.cisco.pxgrid.samples.ise.anc;
import java.io.IOException;
import com.cisco.pxgrid.model.AccountState;
import com.cisco.pxgrid.model.Service;
import com.cisco.pxgrid.samples.ise.http.Console;
import com.cisco.pxgrid.samples.ise.http.PxgridControl;
import com.cisco.pxgrid.samples.ise.http.SampleConfiguration;
import com.cisco.pxgrid.samples.ise.http.SampleHelper;
/**
\ast Demonstrates how to query a session using IP address
*/
public class AncGetEndpoints {
       private static void getEndpoints(SampleConfiguration config) throws IOException {
               PxgridControl pxgrid = new PxgridControl(config);
               Service[] services = pxgrid.lookupService("com.cisco.ise.config.anc");
               if (services == null || services.length == 0) {
                      System.out.println("Service unavailabe");
                      return;
               Service service = services[0];
               String url = service.getProperties().get("restBaseUrl") + "/getEndpoints";
               String secret = pxgrid.getAccessSecret(service.getNodeName());
               SampleHelper.postStringAndPrint(url, config.getUserName(), secret,
config.getSSLContext().getSocketFactory(), "{}");
       }
       public static void main(String [] args) throws Exception {
               SampleConfiguration config = new SampleConfiguration();
               PxgridControl pxgrid = new PxgridControl(config);
               while (pxgrid.accountActivate() != AccountState.ENABLED)
                      Thread.sleep(60000);
               Console.log("pxGrid controller version=" + pxgrid.getControllerVersion());
               getEndpoints(config);
       }
3
```

### **ANCGetEndpointsByMAC**

This code retrieves endpoints with an ISE ANC policy by MAC address

```
package com.cisco.pxgrid.samples.ise.anc;
import java.io.IOException;
import com.cisco.pxgrid.model.AccountState;
import com.cisco.pxgrid.model.Service;
import com.cisco.pxgrid.samples.ise.http.Console;
import com.cisco.pxgrid.samples.ise.http.PxgridControl;
import com.cisco.pxgrid.samples.ise.http.SampleConfiguration;
import com.cisco.pxgrid.samples.ise.http.SampleHelper;
/**
* Demonstrates how to query a session using IP address
*/
public class AncGetEndpointByMac {
       private static class ByMacRequest {
               private String mac;
               public String getMac() {
                      return mac;
               3
               public void setMac(String mac) {
                      this.mac = mac;
               }
       private static void get(SampleConfiguration config, String mac) throws IOException {
               PxgridControl pxgrid = new PxgridControl(config);
               Service[] services = pxgrid.lookupService("com.cisco.ise.config.anc");
               if (services == null || services.length == 0) {
                      System.out.println("Service unavailabe");
                      return;
               }
               Service service = services[0];
               String url = service.getProperties().get("restBaseUrl") + "/getEndpointByMacAddress";
               String secret = pxgrid.getAccessSecret(service.getNodeName());
               ByMacRequest request = new ByMacRequest();
               request.setMac(mac);
               SampleHelper.postObjectAndPrint(url, config.getUserName(), secret,
config.getSSLContext().getSocketFactory(), request);
       public static void main(String [] args) throws Exception {
               SampleConfiguration config = new SampleConfiguration();
               PxgridControl pxgrid = new PxgridControl(config);
               while (pxgrid.accountActivate() != AccountState.ENABLED)
                      Thread.sleep(60000);
               Console.log("pxGrid controller version=" + pxgrid.getControllerVersion());
               String mac = SampleHelper.prompt("Get endpoint by mac: ");
               get(config, mac);
       }
```

### **ANCApplyByIP**

This code retrieves endpoints with an ANCY policy by IP Address

```
package com.cisco.pxgrid.samples.ise.anc;
import java.io.IOException;
import com.cisco.pxgrid.model.AccountState;
import com.cisco.pxgrid.model.Service;
import com.cisco.pxgrid.samples.ise.http.Console;
import com.cisco.pxgrid.samples.ise.http.PxgridControl;
import com.cisco.pxgrid.samples.ise.http.SampleConfiguration;
import com.cisco.pxgrid.samples.ise.http.SampleHelper;
/**
* Demonstrates how to query a session using IP address
* /
public class AncApplyByIp {
       private static class ApplyEndpointPolicyByIpRequest {
               private String policyName;
               private String ipAddress;
               public void setPolicyName(String policyName) {
                      this.policyName = policyName;
               public void setIpAddress(String ipAddress) {
                      this.ipAddress = ipAddress;
               }
       }
       private static void apply(SampleConfiguration config, String policyName, String ip) throws
IOException {
               PxgridControl pxgrid = new PxgridControl(config);
               Service[] services = pxgrid.lookupService("com.cisco.ise.config.anc");
               if (services == null || services.length == 0) {
                      System.out.println("Service unavailabe");
                      return;
               }
               Service service = services[0];
               String url = service.getProperties().get("restBaseUrl") + "/applyEndpointByIpAddress";
               String secret = pxgrid.getAccessSecret(service.getNodeName());
               ApplyEndpointPolicyByIpRequest request = new ApplyEndpointPolicyByIpRequest();
               request.setPolicyName(policyName);
               request.setIpAddress(ip);
               SampleHelper.postObjectAndPrint(url, config.getUserName(), secret,
config.getSSLContext().getSocketFactory(), request);
       }
       public static void main(String [] args) throws Exception {
               SampleConfiguration config = new SampleConfiguration();
               PxgridControl pxgrid = new PxgridControl(config);
               while (pxgrid.accountActivate() != AccountState.ENABLED)
                      Thread.sleep(60000);
               Console.log("pxGrid controller version=" + pxgrid.getControllerVersion());
               String policyName = SampleHelper.prompt("Policy name: ");
               String ip = SampleHelper.prompt("IP address: ");
               apply(config, policyName, ip);
       }
}
```

### **ANCClearByIP**

This code clears or unquarantines endpoints with an ISE ANC policy by IP Address

```
package com.cisco.pxgrid.samples.ise.anc;
import java.io.IOException;
import com.cisco.pxgrid.model.AccountState;
import com.cisco.pxgrid.model.Service;
import com.cisco.pxgrid.samples.ise.http.Console;
import com.cisco.pxgrid.samples.ise.http.PxgridControl;
import com.cisco.pxgrid.samples.ise.http.SampleConfiguration;
import com.cisco.pxgrid.samples.ise.http.SampleHelper;
/**
* Demonstrates how to query a session using IP address
* /
public class AncClearByIp {
       private static class EndpointPolicyByIpRequest {
               private String ipAddress;
               public void setIpAddress(String ipAddress) {
                      this.ipAddress = ipAddress;
               }
       }
       private static void clear(SampleConfiguration config, String ip) throws IOException {
               PxgridControl pxgrid = new PxgridControl(config);
               Service[] services = pxgrid.lookupService("com.cisco.ise.config.anc");
               if (services == null || services.length == 0) {
                      System.out.println("Service unavailabe");
                      return;
               Service service = services[0];
               String url = service.getProperties().get("restBaseUrl") + "/clearEndpointByIpAddress";
               String secret = pxgrid.getAccessSecret(service.getNodeName());
               EndpointPolicyByIpRequest request = new EndpointPolicyByIpRequest();
               request.setIpAddress(ip);
               SampleHelper.postObjectAndPrint(url, config.getUserName(), secret,
config.getSSLContext().getSocketFactory(), request);
       }
       public static void main(String [] args) throws Exception {
               SampleConfiguration config = new SampleConfiguration();
               PxgridControl pxgrid = new PxgridControl(config);
               while (pxgrid.accountActivate() != AccountState.ENABLED)
                      Thread.sleep(60000);
               Console.log("pxGrid controller version=" + pxgrid.getControllerVersion());
               String ip = SampleHelper.prompt("IP address: ");
               clear(config, ip);
       }
```

### **ANCCreatePolicy**

This code creates an ISE ANC policy based on Quarantine, Shut\_down, and Port\_Bounce Actions

```
package com.cisco.pxgrid.samples.ise.anc;
import java.io.IOException;
import java.util.Arrays;
import java.util.List;
import com.cisco.pxgrid.model.AccountState;
import com.cisco.pxgrid.model.Service;
import com.cisco.pxgrid.samples.ise.http.Console;
import com.cisco.pxgrid.samples.ise.http.PxgridControl;
import com.cisco.pxgrid.samples.ise.http.SampleConfiguration;
import com.cisco.pxgrid.samples.ise.http.SampleHelper;
/**
\ast Demonstrates how to query a session using IP address
*/
public class AncCreatePolicy {
       private static class Policy {
              public enum Action {
                      QUARANTINE, SHUT DOWN, PORT BOUNCE
               private String name;
               private List<Action> actions;
               public String getName() {
                      return name;
               }
               public void setName(String name) {
                      this.name = name;
               }
               public List<Action> getActions() {
                      return actions;
               }
               public void setActions(List<Action> actions) {
                      this.actions = actions;
               }
       }
       private static void create(SampleConfiguration config, String policyName) throws IOException {
               PxgridControl pxgrid = new PxgridControl(config);
               Service[] services = pxgrid.lookupService("com.cisco.ise.config.anc");
               if (services == null || services.length == 0) {
                      System.out.println("Service unavailabe"
                      return;
               }
               Service service = services[0];
               String url = service.getProperties().get("restBaseUrl") + "/createPolicy";
               String secret = pxgrid.getAccessSecret(service.getNodeName());
               Policy policy = new Policy();
               policy.setName(policyName);
               policy.setActions(Arrays.asList(Policy.Action.QUARANTINE));
               SampleHelper.postObjectAndPrint(url, config.getUserName(), secret,
config.getSSLContext().getSocketFactory(), policy);
       }
       public static void main(String [] args) throws Exception {
               SampleConfiguration config = new SampleConfiguration();
               PxgridControl pxgrid = new PxgridControl(config);
               while (pxgrid.accountActivate() != AccountState.ENABLED)
                      Thread.sleep(60000);
               Console.log("pxGrid controller version=" + pxgrid.getControllerVersion());
               String policyName = SampleHelper.prompt("Create policy name: ");
               create(config, policyName);
       }
```

### **ANCDeletePolicy**

This code deletes an ISE ANC policy

```
package com.cisco.pxgrid.samples.ise.anc;
import java.io.IOException;
import java.util.Arrays;
import java.util.List;
import com.cisco.pxgrid.model.AccountState;
import com.cisco.pxgrid.model.Service;
import com.cisco.pxgrid.samples.ise.http.Console;
import com.cisco.pxgrid.samples.ise.http.PxgridControl;
import com.cisco.pxgrid.samples.ise.http.SampleConfiguration;
import com.cisco.pxgrid.samples.ise.http.SampleHelper;
/**
* Demonstrates how to query a session using IP address
*/
public class AncDeletePolicy {
       private static class Policy {
              private String id;
               public String getId() {
                      return id;
               }
               public void setId(String id) {
                      this.id = id;
               }
       }
       private static void delete(SampleConfiguration config, String id) throws IOException {
               PxgridControl pxgrid = new PxgridControl(config);
               Service[] services = pxgrid.lookupService("com.cisco.ise.config.anc");
               if (services == null || services.length == 0) {
                      System.out.println("Service unavailabe");
                      return:
               }
               Service service = services[0];
               String url = service.getProperties().get("restBaseUrl") + "/deletePolicyById";
               String secret = pxgrid.getAccessSecret(service.getNodeName());
               Policy policy = new Policy();
               policy.setId(id);
               SampleHelper.postObjectAndPrint(url, config.getUserName(), secret,
config.getSSLContext().getSocketFactory(), policy);
       }
       public static void main(String [] args) throws Exception {
               SampleConfiguration config = new SampleConfiguration();
               PxgridControl pxgrid = new PxgridControl(config);
               while (pxgrid.accountActivate() != AccountState.ENABLED)
                      Thread.sleep(60000);
               Console.log("pxGrid controller version=" + pxgrid.getControllerVersion());
               String id = SampleHelper.prompt("Create policy id: ");
               delete(config, id);
       }
```

### **ANCGetByOperationID**

This code obtains the OperationID for retrieving the ANC policy.

```
package com.cisco.pxgrid.samples.ise.anc;
import java.io.IOException;
import com.cisco.pxgrid.model.AccountState;
import com.cisco.pxgrid.model.Service;
import com.cisco.pxgrid.samples.ise.http.Console;
import com.cisco.pxgrid.samples.ise.http.PxgridControl;
import com.cisco.pxgrid.samples.ise.http.SampleConfiguration;
import com.cisco.pxgrid.samples.ise.http.SampleHelper;
/**
* Demonstrates how to query a session using IP address
*/
public class AncGetByOperationId {
       private static class GetOperationStatusRequest {
               private String operationId;
               public void setOperationId(String operationId) {
                       this.operationId = operationId;
               }
        }
       private static void getOperationStatus(SampleConfiguration config, String id) throws IOException {
               PxgridControl pxgrid = new PxgridControl(config);
               Service[] services = pxgrid.lookupService("com.cisco.ise.config.anc");
if (services == null || services.length == 0) {
                       System.out.println("Service unavailabe");
                       return;
               Service service = services[0];
               String url = service.getProperties().get("restBaseUrl") + "/getOperationStatus";
               String secret = pxgrid.getAccessSecret(service.getNodeName());
               GetOperationStatusRequest request = new GetOperationStatusRequest();
               request.setOperationId(id);
               SampleHelper.postObjectAndPrint(url, config.getUserName(), secret,
config.getSSLContext().getSocketFactory(), request);
       }
       public static void main(String [] args) throws Exception {
               SampleConfiguration config = new SampleConfiguration();
               PxgridControl pxgrid = new PxgridControl(config);
               while (pxgrid.accountActivate() != AccountState.ENABLED)
                       Thread.sleep(60000);
               Console.log("pxGrid controller version=" + pxgrid.getControllerVersion());
               String id = SampleHelper.prompt("Operation ID: ");
               getOperationStatus(config, id);
       }
```

## **Cisco pxGrid Context-In**

Cisco pxGrid Context-in provides ecosystem partners to publish device or endpoint asset context attributes into ISE based on the pxGrid client publishing to the asset topic or asset service, /topic/com.cisco.endpoint.asset. ISE will subscribe to this /topic/com.cisco.endpoint.asset topic. An ISE profiling policy will be created using these asset attributes. This ISE profile will be assigned an ISE logical profile to be used in an ISE authorization policy to provide secure network access for the device or endpoint asset. Custom attributes can also be defined in a profiling policy and can be used in an ISE authorization policy as well.

These attributes include: assetID, assetName, assetIpAddress, assetMacAddress, assetVendor, assetProductId, assetSerialNumber, assetDeviceType, assetSwRevision, assetHwRevision, assetProtocol and are defined in the IOT Assets profiling dictionary when creating the Asset's profiling policy. The asset attribute values are defined by the ecosystem partner in the custompublisher.java code. The API\_Simulator Code listing is provided in the API\_Simulator Context-In section.

IOTASSET
4
↓ ■
assetDeviceType
assetHwRevision
assetId
assetIpAddress
assetMacAddress
assetName
assetProductId
assetProtocol
assetSerialNumber
assetSwRevision
assetVendor

If the ecosystem partner has custom attributes that are not defined in the IOT Assets profiling dictionary, these can also be published. These custom attributes are defined in JSON array in the custompublisher.java code. These are also defined in the Custom Attributes profiling policies and defined in the ISE Identity Custom endpoint settings.

CUSTOMATTRIBUTE	
	\$
	1
assetGroup	
	1

The Cisco ISE pxGrid node controller becomes the subscriber to the endpoint Asset topic and consumes this information.

ululu cisco	Identity Services Engi	NO Home	Context Visibility	<ul> <li>Operations</li> </ul>	Policy A	dministration 🔹 W	ork Centers		1	License Warning	<u>\</u>	<b>6</b> 4
► Sy	stem  Identity Managen	nent   Network Res	sources	Portal Management	pxGrid Services	Feed Service	Threat Centric NAC	Clic	k here to do wireles	s setup and visibility	setup Do not sho	w this again.
	All Clients Web Clien	ts Capabilities	Live Log	Settings Cer	rtificates F	Permissions						
								Rows/Page	5 -	1 🗘 / 1 📄	Go 5 To	tal Rows
C I	Refresh										▼ Filter -	۰.
	Client Name	Connect To	Session Id	Certifi	icate	Subscriptions		Publications	IP Address	Status	Start time	
×	Client Name	-							IP Address	-		
	ise-admin-ise24fc3	ise24fc3	ise24fc3:0	CN=ise	e24fc3.lab	/topic/com.cisco.end	lpoint.asset		192.168.1.251	ON	2018-04-14 15:	24:18 UTC
	ise-fanout-ise24fc3	ise24fc3	ise24fc3:1	CN=ise	e24fc3.lab	/topic/wildcard			127.0.0.1	ON	2018-04-14 15	25:01 UTC
	ise-fanout-ise24fc3	ise24fc3	ise24fc3:2	CN=ise	e24fc3.lab	/topic/distributed		/topic/distributed	192.168.1.251	ON	2018-04-14 15:	25:01 UTC
	ise-mnt-ise24fc3	ise24fc3	ise24fc3:3	CN=ise	e24fc3.lab	/topic/com.cisco.ise.	session.internal	/topic/com.cisco.ise.se.	192.168.1.251	ON	2018-04-14 15:	25:01 UTC
	ise-bridge-ise24fc3	ise24fc3	ise24fc3:4	CN=ise	e24fc3.lab			/topic/com.cisco.ise.se.	127.0.0.1	ON	2018-04-14 15:	28:04 UTC

The pxGrid client publishes to the Asset topic, /topic/com.cisco.endpoint.asset. IO1 is the registered pxGrid client publishing to this asset topic.

cisco	Identity Services Engin	ne Home I	Context Visibility	<ul> <li>Operations</li> </ul>	▶ Policy	- Administration	Work Centers		1	icense Warning 🔺	् 🔮	0 ¢
► Sy	stem   Identity Managem	ent + Network Res	ources + Device F	ortal Management	pxGrid Ser	rvices Feed Serv	ice	Clic	k here to do wireless	setup and visibility se	tup Do not show	this again. ×
,	Mclianta Wah Cliant	a Canabilities	Live Lee	Sottingo C	artificatos	Pormissions						
	All Clients Web Client	s Capabilities	Live Log	Settings C	erincates	Permissions						
								Rows/Page	6 -	1 0/1	Go 6 Tota	al Rows
C F	Refresh										▼ Filter ▼	¢-
	Client Name	Connect To	Session Id	Certi	ficate	Subscription	s Publications		IP Address	Status	Start time	
×	Client Name	•							IP Address	•		
	ise-admin-ise24fc3	ise24fc3	ise24fc3:0	CN=i	se24fc3.lab	. /topic/com.cis	co.endpo		192.168.1.25	I ON	2018-04-14 1	15:24:18 L
	ise-fanout-ise24fc3	ise24fc3	ise24fc3:1	CN=i	se24fc3.lab	. /topic/wildcard			127.0.0.1	ON	2018-04-14 1	15:25:01 L
	ise-fanout-ise24fc3	ise24fc3	ise24fc3:2	CN=i	se24fc3.lab	. /topic/distribut	ed /topic/distribute	ed	192.168.1.25	ON	2018-04-14 1	15:25:01 L
	ise-mnt-ise24fc3	ise24fc3	ise24fc3:3	CN=i	se24fc3.lab	. /topic/com.cis	co.ise.se /topic/com.cisc	co.ise.session.internal,/topic	c 192.168.1.25	ON	2018-04-14 1	15:25:01 L
	ise-bridge-ise24fc3	ise24fc3	ise24fc3:4	CN=i	se24fc3.lab		/topic/com.cisc	co.ise.session.aroup	127.0.0.1	ON	2018-04-14 1	15:28:04 1
	IOT1	ise24fc3	ise24fc3:5	CN=	lohns-Macb		/topic/com.cisc	co.endpoint.asset	192.168.1.13	3 ON	2018-04-14 1	17:18:23 l

### Enabling pxGrid as Subscriber for Profiling

 Step 1
 Select Administrator->System->Deployment>the ISE node->Edit the node->Profiling Configuration

- Step 2 Enable pxGrid
  - 🗹 🚽 pxGrid

Description The PXgrid probe to fetch attributes of MAC or IP-Address as a subscriber from PXGrid

#### Step 3 Select Save

#### Step 4 Select Administrator->pxGrid Services->Web Clients

Note: The ISE admin node has subscribed to the endpoint asset topic

ារប្រ cisco	Identity Services Engin	Ie Home 🔸	Context Visibility	Operations	Policy - Ac	Iministration • Work Centers		🧻 License Warning 🔺	0 0 ¢
► Sy	stem  Identity Managen	nent I Network Res	ources + Device F	Portal Management	pxGrid Services	Feed Service + Threat C	entric NAC	Click here to do wireless setup and visibility setup Do not sl	how this again. ×
All	Clients Web Clients	Capabilities Live L	og Settings (	Certificates Permis	ssions				
								Rows/Page 6 \$ 1 /1 Go	6 Total Rows
C	Refresh							Ŧ	Filter 🗸 🌣 🗸
	Client Name	Connect To	Session Id	Certifi	cate	Subscriptions		Publications	IP Address
×	Client Name	ŧ							IP Address
	ise-fanout-ise24fc2	ise24fc2	ise24fc2:0	CN=ise	e24fc2.lab	/topic/wildcard			127.0.0.1
	ise-admin-ise24fc2	ise24fc2	ise24fc2:1	CN=ise	e24fc2.lab	/topic/com.cisco.endpoint.asset			192.168.1.250
	ise-mnt-ise24fc2	ise24fc2	ise24fc2:3	CN=ise	e24fc2.lab	/topic/com.cisco.ise.session.inte	ernal	/topic/com.cisco.ise.session.internal,/topic/com.cisco.ise.se.	192.168.1.250
	ise-bridge-ise24fc2	ise24fc2	ise24fc2:4	CN=ise	e24fc2.lab			/topic/com.cisco.ise.session.group	127.0.0.1
	ise-fanout-ise24fc2	ise24fc2	ise24fc2:5	CN=ise	e24fc2.lab	/topic/distributed		/topic/distributed	192.168.1.250
	CiscolSEpxGridApp2	ise24fc2	ise24fc2:7	CN=qr	adar3.lab1	/topic/com.cisco.ise.session,/top	pic/com.cisco.ise.radiu		192.168.1.249

### **Running API\_Simulator**

The API Simulator is a tool that is used to provide pxGrid context-in simulating a pxGrid client asset devicepublishing asset attributes to the ISE pxgrid node.

The ./run\_publisher38 script contains the command-line arguments to run the simulator, this also provides the access secret between the publisher and ISE pxGrid node to begin publishing content from the asset device or client endpoint in the script.

If you desire to modify the existing attributes in the script you must modify the custompublisher.java code. An example is provided using Maven.

A profiling policy will be created based on the asset attribute values from the client device. The profiling policy will then be assigned a Logical Profile that can be assigned in an ISE authorization condition rule that can be used in an ISE authorization policy to determine network access for the asset device. We will also illustrate the same example using custom attributes.

- **Step 1** Extract the contents of the API Simulator to a folder
- **Step 2** Create a PKCS12 certificate from **ISE->Administration->pxGrid Services->Certificates** and provide the following information:

Note: CN name should be Fully Qualified Domain Name (FQDN) resolvable. PKCS12 format is not supported using Python libraries.

dentity Services Engine Ho	me   Context Visibility   Operations   Policy   Administration   Work Centers	
System      Identity Management     Ne	twork Resources	
All Clients Web Clients Capabilities	Live Log Settings Certificates Permissions	
Generate pxGrid Certificates		
I want to *	Generate a single certificate (without a certificate signing request)	•
Common Name (CN) *	Johns-macbook-pro.Jab10.com	
Description	pxGrid	
Certificate Template	PxGrid_Certificate_Template	
Subject Alternative Name (SAN)	FQDN John-macbook-pro.lab1( = 1	
Certificate Download Format *	PKCS12 format (including certificate chain; one file for both the certificate chain and key)	- 0
Certificate Password *		0
Confirm Password *		
	Reset Create	
Connected to pxGrid ise24fc3.lab10.com		

- Step 3 Select Create
- Step 4Download the zipped file<br/>You should see:

Contificant States 2
johns-macbook-
pro.iab10.coiab10.com.p12
29
1524516829815_cert.zip

Step 5 Edit the */run\_publisher38.sh* file:

where:

-DPXGRID\_HOSTNAMES= ISE pxGrid node (i.e. ise24fc3.lab10.com) -DPXRID\_USERNAME= pxGrid client name (i.e. IOT1) -DPXGRID\_GROUP= pxGrid client group (i.e. Session) -DPXGRID\_KEYSTORE\_FILENAME= client keystore file (i.e. Johns-Macbook-Pro.lab10.com\_Johns-Macbook-Pro.lab10.com.p12) -DPXGRID\_KEYSTORE\_PASSWORD=client keystore file password (i.e. Cisco123) -DPXGRID\_TRUSTSTORE\_FILENAME= truststore or root keystore files (i.e. Johns-Macboo-Pro.lab10.com\_Johns-Macbook-Pro.lab10.com.p12) -DPXGRID\_KEYSTORE\_PASSWORD=truststore or root keystore file password (i.e. Cisco123)

```
java -jar -DPXGRID_HOSTNAMES="ise24fc23.lab10.com" -DPXGRID_USERNAME="IOT1" -DPXGRID_GROUP="Session" -
DPXGRID_KEYSTORE_FILENAME="Johns-Macbook-Pro.lab10.com_Johns-Macbook-Pro.lab10.com.pl2" -
DPXGRID_KEYSTORE_PASSWORD="Ciscol23" -DPXGRID_TRUSTSTORE_FILENAME="Johns-Macbook-Pro.lab10.com_Johns-Macbook-
Pro.lab10.com.pl2" -DPXGRID_TRUSTSTORE_PASSWORD="Ciscol23" pxgrid-rest-ws-samples-2.0.0-SNAPSHOT.jar
```









username=IOT1 password=null groups=Session description=null keystoreFilename=Johns-Macbook-Pro.lab10.com Johns-Macbook-Pro.lab10.com.p12 keystorePassword=Cisco123 truststoreFilename=Johns-Macbook-Pro.lab10.com\_Johns-Macbook-Pro.lab10.com.p12 truststorePassword=Cisco123 2018-04-14 13:18:19.919 INFO 5546 --- [ main] c.c.p.samples.ise.http.PxgridControl : Request={} 2018-04-14 13:18:20.590 INFO 5546 --- [ main] c.c.p.samples.ise.http.PxgridControl • Response={"accountState":"ENABLED", "version": "2.0.0.13"} 14-Apr-18 13:18:20.591 [main-1]: pxGrid controller version=2.0.0.13 2018-04-14 13:18:20.601 INFO 5546 --- [ main] c.c.p.samples.ise.http.PxgridControl Request={"name":"com.cisco.endpoint.asset","properties":{"wsPubsubService":"com.cisco.ise.pubsub", "restBaseUR L":"http://raghdasa-lnv1:8080","assetTopic":"/topic/com.cisco.endpoint.asset"}} 2018-04-14 13:18:20.735 INFO 5546 --- [ main] c.c.p.samples.ise.http.PxgridControl : Response={} 2018-04-14 13:18:20.742 INFO 5546 -main] c.c.p.samples.ise.http.PxgridControl Request={"name":"com.cisco.ise.pubsub"} main] c.c.p.samples.ise.http.PxgridControl 2018-04-14 13:18:20.751 INFO 5546 --Response={"services":[{"name":"com.cisco.ise.pubsub", "nodeName":"ise-pubsubise24fc3", "properties": {"wsUrl": "wss://ise24fc3.lab10.com:8910/pxgrid/ise/pubsub"}}]} 14-Apr-18 13:18:20.751 [main-1]: wsUrl=wss://ise24fc3.lab10.com:8910/pxgrid/ise/pubsub 2018-04-14 13:18:20.757 INFO 5546 --- [ main] c.c.p.samples.ise.http.PxgridControl Request={"peerNodeName":"ise-pubsub-ise24fc3"} 2018-04-14 13:18:20.909 INFO 5546 main] c.c.p.samples.ise.http.PxgridControl ſ Response={"secret":"tASXtHAbKDKbAODh"} Grizzly(1)] c.c.p.s.i.h.StompPubsubClientEndpoint 2018-04-14 13:18:22.072 INFO 5546 --- [ : WS onOpen 2018-04-14 13:18:22.075 INFO 5546 --- [ main] c.c.p.s.i.h.StompPubsubClientEndpoint : STOMP CONNECT host=ise24fc3.lab10.com press <enter> to start the publishing...2018-04-14 13:18:22.084 INFO 5546 --- [ Grizzly(2)] c.c.p.s.i.h.StompPubsubClientEndpoint : STOMP CONNECTED version=1.2

### Viewing Asset Device in Context Visibility Screen

#### Step 1 Select Context Visibility->Endpoints->Endpoint Classification

You will see the client endpoint generated from the script as determined by the MAC address and the client IP address.

Note: if you do not see the endpoint, verify that the ISE admin node has subscribed to the endpoint asset topic. If the ISE Admin node has not subscribed to the ISE endpoint asset topic, ensure that pxGrid is enabled for Profiling under Administration->System->Deployment->edit the node and select Profiling Configuration.



**Step 2** Click on the **MAC address**, and select **Attributes**, you will see the attributes:

Identity Services Eng	gine Home	- Context Visibility	<ul> <li>Operation</li> </ul>	ns Policy	► Admi	nistration
Endpoints Users Netwo	rk Devices App	olication				
Endpoints > 7A:F9:AE:9A:1	8:00					
7A:F9:AE:9A:18:00	) ឲ្យ					
Username: Endpoint P Current IP Location:	rofile: Unknown Address: 123.157.	.179.78				
Applications	Attributes	Authentication	Threats	Vulnerabilities		
General Attributes						
Description						
Static Assignment	false					
Endpoint Policy	Unknown					
Static Group Assignment	false					
Identity Group Assignment	Unknown					
Custom Attributes						
				Ŧ	Filter -	<b>\$</b> -
Attribute Name		Attribute Va	alue			
× Attribute Name		Attribute Va	lue			

Other Attributes	
BYODRegistration	Unknown
DeviceRegistrationStatus	NotRegistered
ElapsedDays	0
EndPointPolicy	Unknown
EndPointProfilerServer	ise24fc3.lab10.com
EndPointSource	PXGRIDPROBE
IdentityGroup	Unknown
InactiveDays	0
MACAddress	7A:F9:AE:9A:18:00
MatchedPolicy	Unknown
OUI	UNKNOWN
PolicyVersion	0
PostureApplicable	Yes
StaticAssignment	false
StaticGroupAssignment	false
Total Certainty Factor	0
assetConnectedLinks	[{"value":"3","key":"indattr2"},{"value":"Root","key":"assetGroup"},{"value":"1","key":"indattr3"}]
assetDeviceType	EtherNet/IP Node
assetHwRevision	5.6
assetId	230

assetIpAddress	123.157.179.78
assetMacAddress	7a:f9:ae:9a:18:00
assetName	Abjergaryn - 45
assetProductId	IE2000
assetProtocol	CIP
assetSerialNumber	1212121213243
assetSwRevision	4.6
assetVendor	Cisco Systems
ip	123.157.179.78

### **Creating Profiling Policy Based on Asset Attributes**

In this example, we define an IOT\_Example1 profiling policy based on the attribute values of the client or asset device. These attribute values are mapped to the IOTASSETS dictionary attributes to define the policy. This profiling policy will then be assigned to an IOTDevices1 logical profile. This logical profile can then be added as an ISE authorization condition rule to be used in an ISE authorization policy to determine the asset's network access. A security group tag of IOT devices will also be created and added to the ISE authorization policy.

Note: When running the script, you will not see the RADIUS events that trigger this rule, since the script simulates endpoint. However, when you implement the code and test in your environment, this rule will be triggered.

Step 1 Select Policy->Profiling->Profiling Policies->Add You will see the following:

Identity Services	Engine	Home	Context Visibility	Operations	- Policy	Administration	Work Centers		1
Policy Sets Profiling	Posture	Client Provision	ing + Policy Eleme	nts					
Profiling		<u>م</u> ڇر	Profiler Policy List : Profiler Policy • Minir • Network Create an identity	New Profiler Polic     Name     Policy Enable num Certainty Facto     Exception Action     Scan (NMAP) Action     Group for the police	ay e   d Ø ir 10 n NONE n NONE y ⊙ Yes, ○ No, i	create matching identi	Description     (Valid Range 1 to 65532     v     v     y Group     up hierarchy	)	
			Rules If Condition C Submit Canc	* Parent Polic ssociated CoA Type System Type onditions $\diamondsuit$ Th al	y NONE e Global : e	Settings	v v t 10	@•	

- Step 2 Enter: IOT\_Example1 for Name
- Step 3 Under Rules->If Condition->Create New Condition (Advanced Option)
- Step 4 Under Expression->Select Attribute->Iotas set
- **Step 5** You should see:

1

* Name	IOT Example1	Description		
- Name	IOI_Example1	Description		
Policy Enabled				
* Minimum Certainty Factor	10	IOTASSET		
* Exception Action	NONE -	<	<b>م</b> ₊	
* Network Scan (NMAP) Action	NONE -	assetDeviceType		
Create an Identity Group for the policy	Yes, create matching Identity Group	assetHwRevision		
	O No, use existing Identity Group his	assetId		
* Parent Policy	NONE	assetIpAddress		
i urone i onoy	NONE	assetMacAddress		
* Associated CoA Type	Global Settings -	assetName		
System Type		assetProductId		
		assetProtocol		
Pulos		assetSerialNumber		
Rules		assetSwRevision		
If Condition Select_Attribute G	Then Certainty Factor Increases	assetVendor		- tiga -
Condition Name	Expression			
Submit C:	Select Attribute			÷

- Step 6 Select assetDeviceType:contains:EtherNetVIP Node, enter, increase certainty factor to 100
- Step 7 Select "Gear"->"Insert line below"

Profiler Policy List > New Profiler Policy Profiler Policy	
* Name	IOT_Example1 Description
Policy Enabled	2
* Minimum Certainty Factor	10 (Valid Range 1 to 65535)
* Exception Action	NONE
* Network Scan (NMAP) Action	NONE v
Create an Identity Group for the policy	Yes, create matching Identity Group
	O No, use existing Identity Group hierarchy
* Parent Policy	NONE
* Associated CoA Type	Global Settings •
System Type	
Rules	
If Condition IOTASSET_assetDevic	eType_CONTAINS 💠 Then Certainty Factor Increases 🔻 100
If Condition Select_Attribute <	> Then Certainty Factor Increases
Submit Cancel	

- Step 8 Under Rules->If Condition->Create New Condition(Advanced Option)
- Step 9 Under Expression->Select Attribute->IOTAsset
- Step 10 Select assetHwRevision:contains:5.6, enter, increase certainty factor to 100
- Step 11 Select "Gear"->"Insert line below"
- **Step 12** You should see:

Policy Enabled	
* Minimum Certainty Factor	10 (Valid Range 1 to 65535)
* Exception Action	NONE
* Network Scan (NMAP) Action	NONE
Create an Identity Group for the policy	Yes, create matching Identity Group
	○ No, use existing Identity Group hierarchy
* Parent Policy	NONE
* Associated CoA Type	Global Settings v
System Type	
Rules	
If Condition IOTASSET_assetDevic	eType_CONTAINS $\diamondsuit$ Then Certainty Factor Increases $\bullet$ 100
If Condition IOTASSET_assetHwRe	wision_CONTAINS_5.6 $\diamondsuit$ Then Certainty Factor Increases 🔹 100
If Condition Conditions	n Certainty Factor Increases 🔹 10
Submit Cancel	

- Step 13 Under Rules->If Condition->Create New Condition(Advanced Option)
- Step 14 Under Expression->Select Attribute->IOT Asset
- Step 15 Select assetId:contains:230, enter, increase certainty factor to 100
- Step 16 Select "Gear"->"Insert line below"
- Step 17 Under Rules->If Condition->Create New Condition(Advanced Option)
- Step 18 Under Expression->Select Attribute->IOT Asset
- Step 19 Select assetIpAddress:contains:123.157.78.79, enter, increase certainty factor to 100
- Step 20 Select "Gear"->"Insert line below
- Step 21 Under Rules->If Condition->Create New Condition(Advanced Option)
- Step 22 Under Expression->Select Attribute->IOT Asset
- Step 23 Select assetMacAddress:contains:78:f9:ae:9a:18:00, enter, increase certainty factor to 100
- Step 24 Select "Gear"->"Insert line below
- Step 25 Under Rules->If Condition->Create New Condition(Advanced Option)
- Step 26 Under Expression->Select Attribute->IOT Asset
- Step 27 Select assetId:contains:230, enter, increase certainty factor to 100
- Step 28 Select "Gear"->"Insert line below
- Step 29 Under Rules->If Condition->Create New Condition(Advanced Option)
- Step 30 Under Expression->Select Attribute->IOT Asset
- Step 31 Select assetName:contains:Abjergaryn 45 enter, increase certainty factor to 100
- Step 32 Select "Gear"->"Insert line below
- Step 33 Under Rules->If Condition->Create New Condition(Advanced Option)
- Step 34 Under Expression->Select Attribute->IOT Asset
- Step 35 Select assetProductId:contains:IE2000, enter, increase certainty factor to 100
- Step 36 Select "Gear"->"Insert line below
- Step 37 Under Rules->If Condition->Create New Condition(Advanced Option)
- Step 38 Under Expression->Select Attribute->IOT Asset
- **Step 39** Select **assetProtocol:contains:CIP**, enter, increase certainty factor to 100
- Step 40 Select "Gear"->"Insert line below
- Step 41 Under Rules->If Condition->Create New Condition(Advanced Option)

cisco



- Step 42 Under Expression->Select Attribute->IOT Asset
- Step 43 Select assetSerialNumber:contains: 1212121213243, enter, increase certainty factor to 100
- Step 44 Select "Gear"->"Insert line below
- Step 45 Under Rules->If Condition->Create New Condition(Advanced Option)
- Step 46 Under Expression->Select Attribute->IOT Asset
- Step 47 Select assetSwRevision:contains:4.6, enter, increase certainty factor to 100
- Step 48 Select "Gear"->"Insert line below
- Step 49 Under Rules->If Condition->Create New Condition(Advanced Option)
- Step 50 Under Expression->Select Attribute->IOT Asset
- Step 51 Select assetVendor:contains:Cisco Systems, enter, increase certainty factor to 100
- Step 52 Select "Gear"->"Insert line below
- **Step 53** You should see the following

Rules
If Condition IOTASSET_assetIpAddress_CONTAINS_4 $\diamondsuit$ Then Certainty Factor Increases <b>v</b> 100
If Condition IOTASSET_assetMacAddress_CONTAINS $$ Then Certainty Factor Increases $\checkmark$ 100
If Condition IOTASSET_assetName_CONTAINS_Abjer 💠 Then Certainty Factor Increases 🔹 100
If Condition IOTASSET_assetProductId_CONTAINS_IE $\diamondsuit$ Then Certainty Factor Increases $\bullet$ 100
If Condition IOTASSET_assetProtocol_CONTAINS_CIP 🔶 Then Certainty Factor Increases 🔹 100
If Condition IOTASSET_assetSerialNumber_CONTAIN $\diamondsuit$ Then Certainty Factor Increases $\bullet$ 100
If Condition IOTASSET_assetSwRevision_CONTAINS_4.6 💠 Then Certainty Factor Increases 🔹 100
If Condition IOTASSET_assetVendor_CONTAINS_Cisc 💠 Then Certainty Factor Increases 🔹 100

#### Step 54 Select Submit

Step 55 Select Funnel, enter IOT\_Example1 under the Profiling Policy Name, you should see:

Profiling P	olicies								
							Selected 0   Total 1	- 😚	÷.
/ Edit 🕂	Add Duplicate	🗙 Delete 🔻	Import	t 🕞 Export 👻		Show	Quick Filter	-	R
Profiling	Policy Name		•	Policy Enabled	System Type	Description			—
IOT_E	ample1		8						
IOT_Ex	ample1			Enabled	Administrator Created				

 Step 56
 Add IOT\_Example1 to Logical Profile

 Select Profiling->Logical Profiles->Add->Name: IOT1Devices->Policy Assignment->Available

 Policies->IOT\_Example1 and move to Assigned Polices

սիսի	Identity	/ Services	Engine	Homo	Context Visibility		- Policy		Mark Contarn	
CISCO	Teornal		Linginio	Home	Context Visibility	Operations	◆Policy	Administration	Work Centers	
Pol	icy Sets	Profiling	Posture	Client Provisio	oning <ul> <li>Policy Elem</li> </ul>	ents				
1	Profiling				Logical Profiles Li	st > New Logical P	rofile			
				Q	Logical Profile	5				
		Ŧ		ŵ.,	* Name	IOT1Devices		Description		
	Prof	iling Policies		171 V						
	🕨 🦰 Logi	cal Profiles			* Policy Assignm	opt				
	_				Policy Assignin	lent				
					Available Policie	s:			Assigned Policies:	
					2Wire-Device				IOT Example1	
					3Com-Device					
					Aastra-Device			<		
					Aastra-IP-Phon	e				
					Aerohive-Acces	s-Point		>>		
					Aerohive-Devic	e				
					American-Powe	r-Conversion-Devic	e			
					Android					
					Submit Can	cel				

#### Step 57 Select Submit

**Step 58** You should see the logical profile IOTDevices

Cisco Identity Services Engine Home	Context Visibility > Operations - Policy > Adr	ninistration	🚺 License Warning 🔺 🔍 @	<b>O</b> 3
Policy Sets Profiling Posture Client Provision	ing   Policy Elements			
Profiling	Logical Profiles		Selected 0   Total 9	<del>ତ</del> କ୍ଷି 🗸
	/ Edit + Add X Delete + Restore Cisco Provided Lo	ogical Profiles	Show All	- 8
Elogical Profiles	Logical Profiles	<ul> <li>System Type</li> </ul>	Description	
·	Cameras	Cisco Provided	Default logical profile for cameras.	
	Gaming Devices	Cisco Provided	Default logical profile for gaming devices.	
	Home Network Devices	Cisco Provided	Default logical profile for home network devices.	
	IOT1Devices	Administrator Created		
	IP-Phones	Cisco Provided	Default logical profile for IP Phones.	
	Infrastructure Network Devices	Cisco Provided	Default logical profile for infrastructure network devices.	
	Medical Devices	Cisco Provided	Default logical profile for medical devices.	
	Mobile Devices	Cisco Provided	Default logical profile for mobile devices.	
	Printers	Cisco Provided	Default logical profile for printers.	

### **Creating Authorization Policy Based on Asset's Logical Profile**

We will create an Authorization policy that determines the asset's network access. A Security Group Tag (SGT) will also be added to the Authorization Policy provided label to classify network traffic if Cisco's TrustSec Solution is used.

 Step 1
 Create IOT device SGT which will be used in the Authorization Policy
 Select WorkCenters->Trustsec->Components->Add->IOT Devices



dentity Services Engin	10	Home	Context Vision	bility	• 0	peration
Network Access     Guest Access	ccess		▶ BYOD	Profil	er	Postu
Overview      Components	Frus	StSec Policy	Policy Sets	► SXF	• •	Trouble
Security Groups IP SGT Static Mapping Security Group ACLs Network Devices Trustsec AAA Servers	G	Security Gr Security • Name IOT_Dev • Icon	oups List > IC / Groups ices	DT_Devic	es ∩ ∎	<ul> <li>♀</li> <li>♀</li> <li>♀</li> </ul>

- Step 2 Select Save
- **Step 3** Create ISE Authorization Profile Policy
- Step 4 Select Policy->Policy Sets->View ">"->Authorization Policy
- **Step 5** You should see the authorization policy:

dinito lo cisco	lentity Serv	ices Engine Home 🕨	Context V	isibility	<ul> <li>Operations</li> </ul>	✓ Policy	<ul> <li>Administration</li> </ul>	► Wor	k Centers		License Warning 🔺	Q,	0	•
Policy S	ets Profili	ng Posture Client Provisionin	g 🔸 Po	olicy Elem	ents									
* Aut	nonzation	Folicy - Local Exceptions												
> Aut	horization	Policy - Global Exceptions (	)											
<b>∀</b> Aut	horization	Policy (14)												
									Results					
+	Status	Rule Name	Cond	litions					Profiles	Security Gro	oups		Hits	Actions
Searc	h													
	Ø	MDM	Ŀ	MDM·N	IDMServerName I	EQUALS Ge	rmantown03		+ PermitAccess	BYOD	× *	+	0	۵
					Wireless_Acces	s								
	$\odot$	Wireless Black List Default	AND	45	IdentityGroup·Na Groups:Blacklist	ame EQUAL	S Endpoint Identity		*Blackhole_Wireless_Access	Select from	list	+	0	¢
	Ø	Profiled Cisco IP Phones	*	IdentityGroup-Name EQUALS Endpoint Identity Groups:Profiled:Cisco-IP-Phone					Cisco_IP_Phones +	Select from	list	+	0	¢

- Step 6 Under Actions, Click on "gear" and "Insert new rule above
- **Step 7** You should see the following:



dialo Id	lentity Serv	vices Engine	Home	Context Visibility	<ul> <li>Operations</li> </ul>	✓ Policy	<ul> <li>Administration</li> </ul>	♦ Work	Centers			License Warning	<u> </u>
Policy Sets Profiling Posture Client Provisioning        Policy Elements      Authorization Policy - Local Exceptions													
Authorization Policy - Global Exceptions (1)													
<b>∀</b> Aut	horization	Policy (15)											
									Results				
+	+ Status Rule Name Conditions								Profiles		Security Gro	ups	
Searc	h												
1	Authorization Rule 1					+			Select from list	+	Select from I	ist	• +

- Step 8 Name Authorization Rule 1 to IOT\_Example\_Rule, Under Condition, select "+"
- **Step 9** Click on **Attribute**, you should see the following Dictionary Attributes

brary		Editor								
Search by Name		ĥ	Click to	o add an attribute						
	0 k 후	~	Select a	attribute for condition						
BYOD_is_Registered	٢		•	⊑ 0 <b>&amp;</b> ⊕	<u> </u>		0	10	) "ເ	Ŵ
Catalyst_Switch_Local_Web_Authentication	on i			Dictionary		Attribute		ID	Info	
Compliance_Unknown_Devices	<i>(i)</i>			All Dictionaries	¢	Attribute		ID		
	~		([1=	Airespace		Aire-Data-Bandwidth-	Average	7		
Compliant_Devices			((:-	Airespace		Aire-Data-Bandwidth-	Average	13	<i>(i)</i>	
EAP-MSCHAPy2	(i)			Airespace		Aire-Data-Bandwidth-	Burst-Do	9		

#### Step 10 Select Endpoints->Logical Profile->Contains->IOT1Devices

Editor					
	EndPoints·LogicalProfile	3			
ĥ	Equals -	IOT1Devices	Ŧ		
	Set to 'Is not'		uplicate	Sav	•

#### Step 11 Select Use

You should see:

cise	lder	ntity Serv	ices Engine	Home >	Context Visibility	Operations     Po	licy Administration	Work Centers		1	License Warning 🔺			•	
Po	licy Sets	Profil rization	ing Posture Cli Policy - Local Ex	ient Provisionii <b>kceptions</b>	ng  Policy Elem	nents									
>	Autho	rization	Policy - Global E	Exceptions	(1)										
*	Autho	rization	Policy (15)												
								Results							
	•	Status	Rule Name		Conditions			Profiles	Sec	urity Group	OS		Hits	Actions	\$
	Search														
	ACME_IOT_Example_Rule & EndPoints-LogicalProf						INS ACME_IOT Device	Select from list	+ Se	lect from list		+		¢	



#### Step 12 Under Profiles, select Permit Access

Step 13 Under Security Groups, select IOT Devices You should see:

cisco	Identity Ser	vices Engine	Home	<ul> <li>Context Vis</li> </ul>	sibility	- Policy	Administration	Work Cer	nters		1	License Warning	j 🔺	
Policy	Sets Profi	ing Posture	Client Provisi	oning + Poli	cy Elements									
≯ Au	thorization	Policy - Loca	I Exceptions	5										
> Au	thorization	Policy - Glob	al Exceptior	ns (1)										
❤ Au	thorization	Policy (15)												
		Rule Name						Re	esults					
T	Status			Condi	tions			Pro	Profiles		Security Gro	ups		
Sea	rch													
/	0	ACME_IOT_E	kample_Rule	ĉ	EndPoints·LogicalProfile	CONTAINS AC	ME_IOT Device	[1	× PermitAccess	+	IOT_Devices	6	× -	+

#### Step 14 Select Save

#### **Step 15** You should see:

ahaha k cisco	dentity S	ervices Engin	Home	Context Visibil	lity > Operations	▼Policy	Administration	<ul> <li>Work Centers</li> </ul>		1	License Warning	<u> </u>	0	Θ	۰.
Policy \$	Sets P	rofiling Postur	e Client Provis	sioning + Policy	Elements										
Search															
	Ø	Default		Default polic	cy set						Default Network Acc	ess	x - +	2	241
> Aut	hentica	tion Policy (3													
> Authorization Policy - Local Exceptions															
> Aut	horizati	on Policy - G	obal Exceptio	ns (1)											
❤ Aut	horizati	on Policy (14													
								Results							
•	Statu	s Rule Nam	е	Condition	ns			Profiles		Security Gr	oups		Hits	Actio	ons
Sear	ch														
	Ø	ACME_IOT	_Example_Rule	έ; Er	ndPoints·LogicalProfile	CONTAINS A	ACME_IOT Device	× PermitAccess	+	IOT_Device	S X	• +	0	¢	ŀ

### Verifying Asset as Defined by the Logical Profile

The Asset policy is now assigned to the logical profile and verified in the Context Visibility Endpoint Classification screen.

#### Step 1 Run script again, type: ./run\_publisher38.sh





2018-04-14 15:42:19.129 INFO 5926 --- [ main] o.s.j.e.a.AnnotationMBeanExporter Registering beans for JMX exposure on startup 2018-04-14 15:42:19.434 INFO 5926 --- [ main] s.b.c.e.t.TomcatEmbeddedServletContainer : Tomcat started on port(s): 8080 (http) 2018-04-14 15:42:19.446 INFO 5926 --- [ main] c.c.p.samples.ise.http.CustomPublisher : Started CustomPublisher in 11.951 seconds (JVM running for 18.438) ----- properties -----hostnames=ise24fc3.lab10.com username=IOT1 password=null groups=Session description=null keystoreFilename=Johns-Macbook-Pro.lab10.com\_Johns-Macbook-Pro.lab10.com.p12 keystorePassword=Cisco123 truststoreFilename=Johns-Macbook-Pro.lab10.com Johns-Macbook-Pro.lab10.com.pl2 truststorePassword=Cisco123 2018-04-14 15:42:21.139 INFO 5926 --- [ mainl c.c.p.samples.ise.http.PxgridControl : Request={} 2018-04-14 15:42:21.209 INFO 5926 --- [ main] c.c.p.samples.ise.http.PxgridControl Response={"accountState":"ENABLED", "version":"2.0.0.13"} 14-Apr-18 15:42:21.210 [main-1]: pxGrid controller version=2.0.0.13 2018-04-14 15:42:21.219 INFO 5926 --- [ main] c.c.p.samples.ise.http.PxgridControl Request={"name":"com.cisco.endpoint.asset","properties":{"wsPubsubService":"com.cisco.ise.pubsub","restBaseUR L":"http://raghdasa-lnv1:8080","assetTopic":"/topic/com.cisco.endpoint.asset"}} 2018-04-14 15:42:21.369 INFO 5926 --- [ main] c.c.p.samples.ise.http.PxgridControl : Response={} 2018-04-14 15:42:21.375 INFO 5926 --main] c.c.p.samples.ise.http.PxgridControl Request={"name":"com.cisco.ise.pubsub"} 2018-04-14 15:42:21.386 INFO 5926 --- [ main] c.c.p.samples.ise.http.PxgridControl Response={"services":[{"name":"com.cisco.ise.pubsub","nodeName":"ise-pubsub-• ise24fc3", "properties": { "wsUrl": "wss://ise24fc3.lab10.com:8910/pxgrid/ise/pubsub" } } ]} 14-Apr-18 15:42:21.386 [main-1]: wsUrl=wss://ise24fc3.lab10.com:8910/pxgrid/ise/pubsub 2018-04-14 15:42:21.392 INFO 5926 main] c.c.p.samples.ise.http.PxgridControl Request={"peerNodeName":"ise-pubsub-ise24fc3"} 2018-04-14 15:42:21.429 INFO 5926 --- [ Response={"secret":"tASXtHAbKDKbAODh"} main] c.c.p.samples.ise.http.PxgridControl • 2018-04-14 15:42:22.692 INFO 5926 --- [ Grizzly(1)] c.c.p.s.i.h.StompPubsubClientEndpoint : WS onOpen 2018-04-14 15:42:22.695 INFO 5926 --- [ main] c.c.p.s.i.h.StompPubsubClientEndpoint : STOMP CONNECT host=ise24fc3.lab10.com press <enter> to start the publishing...2018-04-14 15:42:22.703 INFO 5926 --- [ Grizzlv(2)] : STOMP CONNECTED version=1.2 c.c.p.s.i.h.StompPubsubClientEndpoint 2018-04-14 15:42:29.801 INFO 5926 --- [ main] c.c.p.s.i.h.StompPubsubClientEndpoint : STOMP SEND topic=/topic/com.cisco.endpoint.asset command=SEND, headers={'content-length':'593','destination':'/topic/com.cisco.endpoint.asset',}, content.length=593 14-Apr-18 15:42:29.802 [main-1]: [14-Api-18 15:42:29.802 [main-1]: {"assetHwRevision":"5.6", "assetProtocol":"CIP", "assetConnectedLinks":[{"value":"3", "key":"indattr2"}, {"value" "Root", "key": "assetGroup"}, {"value":"1", "key":"indattr3"}], "assetVendor":"Cisco Systems", "assetSwRevision":"4.6", "assetCustomAttributes":[{"value":"3", "key":"indattr2"}, {"value":"Root", "key ":"assetGroup"}, {"value":"1", "key":"indattr3"}], "assetProductId":"IE2000", "assetSerialNumber":"1212121213243" ,"assetMacAddress":"b0:2c:27:93:fe:94", "assetId":"215", "assetIpAddress":"125.84.172.120", "assetName":"Abjerga ryn - 49","assetDeviceType":"EtherNet\/IP Node"}

#### Step 2 Select->Context Visibility->Endpoint->Endpoint Classification

Note that both the original asset endpoint and the updated one are now assigned to the IOT\_Example1 profile.
alulu ta	lentity Services	Engine		Contract Markeller		Deller							0	
Endpoir		letwork Device	Annlication	Context visibility	Operations	Policy	Administration	ration ⊧ wo	K Centers				~	<b></b> .
Lindboll	Authentiestion	PVOD	Complication		omiand Endpoints	Endosis	t Classificati	on ()	t Miles	mahla E	adasista Hardusra			*d
	Authentication	BTOD	Compile	ance compr	omiseu Enupoinis	Endpoir	II Glassilicati	on Gue	a vuine	stable E	nupuinis naruware			<u>∽</u> /
E	NDPOINTS	0		ø	Ø END		FEGORIE	S O	ø	Ø	NETWORK DEVICES	P	Ø	-
_	Type Profile				OUI	OS Types Ide	entity Group				Location Type Device Name			
	mobilvices: [33.	33%]	).	mise: [66.67%]	abt	le, inc.: [33.33%		unk	iown: [66.67%]		locat_store: [1009,]		Go	3 Total Rows
c	+ 🖸 🏛	ANC -	Change Auth	orization - Clea	r Threats & Vulner	abilities Exp	ort = Imj	port = MDM	Actions - Re	elease R	Rejected Revoke Certificate		<b>▼</b> F	ilter + 🗘 +
	MAC Address	AI	nomalous	IPv4 Address	Userna	me Host	name	Location	Endpoint Pr	rofile	Description	OUI	05	S Types
×	MAC Address	A	nomalous Be	IPv4 Address	Usema	me Host	name	Location	Endpoint Pro	ofile	Description	OUI	0	S Types
	10:DD:B1:C9	:3C:39		192.168.1.136	jeppich	johns	-macbo	Location + Al	Apple-Device	9		Apple, Inc.	Ар	ple Mac OS X 10.
	7A:F9:AE:9A:	:18:00		123.157.179.78					IOT_Example	e1		UNKNOWN		
	B0:2C:27:93:	FE:94		125.84.172.120					IOT_Example	e1		UNKNOWN		

Step 3 If you click on either of the MAC addresses, you will see the logical profile

Identity Services Eng	jine Home	- Context Visibility	► Operations	Policy	Administration									
Endpoints Users Netwo	rk Devices Applie	ation												
Endpoints > 7A:F9:AE:9A:1	Endpoints > 7A:F9:AE:9A:18:00													
7A:F9:AE:9A:18:00	១ ០៤	8												
MAC Address: 7A:F9:AE:9A:18:00 Username: Endpoint Profile: IOT_Example1 Current IP Address: 123.157.179.78 Location:														
Applications	Attributes	Authentication TI	hreats Vuln	erabilities										
General Attributes														
Description														
Static Assignment	false													
Endpoint Policy	IOT_Example1													
Static Group Assignment	false													
Identity Group Assignment	IOT_Example1													

#### **Creating Profiling Policy based on Asset Custom Attributes**

A custom asset policy "CustomIOT" will be created. The custom attribute will be "assetgroup" and the value" will be "root". The custom attribute endpoint setting must be enabled in ISE to use this feature.

#### **Enabling Custom Attribute Value**

Step 1 Select Administration->System->Settings->Profiling->enable->Enable

	cisco Iden	tity Services E	Engine	Home Co	ontext Visibility	<ul> <li>Operations</li> </ul>	▶ Policy	- Administration	→ W		
	- System	Identity Mar	nagement	Network Resou	rces Device	Portal Managemer	nt pxGrid S	Services Feed S	ervice		
	Deploymen	t Licensing	Certificate	s I Logging	Maintenance	Upgrade F	Backup & Res	store      Admin Acc	ess 🔻		
	Client Provisi	oning	G	Profiler Co	nfiguration						
FIPS Mode					* CoA Ty	pe: No CoA	No CoA 👻				
Security Settings				Cu	•						
	Alarm Setting	IS		Ch	ange custom SNN	IP community strin	ngs:				
	Posture			Confirm cha	nged custom SNM	IP community strin	ngs:				
	Profiling				En	dPoint Attribute Fil	iter: 🗌 Enab	led (i)			
	Protocols			E	Enable Anomalous	Behaviour Detecti	ion: 🗌 Enab	led (i)			
	Proxy			Ena	ble Anomalous Be	haviour Enforceme	ent: 🗌 Enab	led			
	SMTP Server	r		Enable Cus	stom Attribute for	Profiling Enforceme	ent: 🗹 Enab	led			

#### Step 2 Select Save

#### Step 3 Select->Administration->Identity Management->Settings->Endpoint Custom Attributes





#### Step 5 Select Policy-Profiling->Profiling Policies->Add

You will see the following:

dhaha cisco	Identity	Services	Engine	Home	Context Visibility	Operations	✓ Policy	Administration	Work Centers		1
Polic	cy Sets	Profiling	Posture	Client Provisio	ning + Policy Element	ents					
	rofiling	ing Policies		<u>م</u> هَ٠	Profiler Policy List Profiler Policy • Mini • Network Create an Identi	> New Profiler Poll * Nan Policy Enable mum Certainty Fact * Exception Acti * Exception	icy	, create matching identi use existing identity G Settings	Valid Range 1 to 6553 V V Ity Group vup hierarchy V	(5)	
					Rules						
					If Condition	Conditions 🔶 1	Then Certa	ainty Factor Increases	• 10		
					Submit Can	cel					

- **Step 6** Enter: **CustomIOT** for Name
- Step 7 Under Rules->If Condition->Create New Condition(Advanced Option)
- Step 8 Under Expression->Select Attribute->CUSTOMATTRIBUTE
- **Step 9** You should see:

Profiler Policy List > New Profiler Policy				
rofiler Policy				
* Name	CustomIOT	Description		
Policy Enabled				
* Minimum Certainty Factor	10	(Valid Range 1 to 65535)		
* Exception Action	NONE -	Dictionaries		
* Network Scan (NMAP) Action	NONE 🔻	<₽ •   ■ •		
Create an Identity Group for the policy	• Yes, create matching Identity Gro		>	
	O No, use existing Identity Group h	ie 🧮 ACTIVEDIRECTORY	>	
* Parent Policy	NONE -	ACTIVEDIRECTORY_PROBE	>	
* Associated CoA Type	Global Settings 🔹	CUSTOMATTRIBUTE	>	
System Type		DHCP	>	
		IOTASSET	>	
Rules		IP	>	
		ELLDP	>	
If Condition Select_Attribute G	Then Certainty Factor Increases	MAC	> 🎲 v	
Condition Name	Expression	-	-	
Submit Ca	Select Attribute	<b>O</b>	÷.	-

#### **Step 10** Select **assetGroup:contains:Root**, enter, increase certainty factor to 100

* Name	CustomIOT Description
Policy Enabled	
* Minimum Certainty Factor	10 (Valid Range 1 to 65535)
* Exception Action	NONE
* Network Scan (NMAP) Action	NONE
Create an Identity Group for the policy	● Yes, create matching Identity Grou 🖉 🔻 🔜 👻
	O No, use existing Identity Group hit assetGroup
* Parent Policy	NONE
* Associated CoA Type	Global Settings 🔹
System Type	
Rules	
If Condition CUSTOMATTRIBUTE_	assetGroup_CONT Then 10
Condition Name	e Expression
Submit Ca	CUSTOMATTRIB O CONTAINS - Root

#### Step 11 Select Submit

You should see the policy:

dentity Services Engine Home	Context Visibility     Operations	s  Policy  Administration  Work Cent	ers	🚺 License Warning 🔺 🔍 🥥 💿	ø
Policy Sets Profiling Posture Client Prov	sioning   Policy Elements  Profiling Policies		Click he	ere to do wireless setup and visibility setup Do not show this again	. ×
	/ Edit 4 Add By Duplicate	XDelete • DImport DExport •		Selected 0   Total 1 🚭 🎡	, Z
Profiling Policies     Logical Profiles	Profiling Policy Name custom CustomIOT	Policy Enabled     Enabled	System Type Administrator Created	Description	

Step 12 Add CustomIOTDevices to Logical Profile

Select Profiling->Logical Profiles->Add->Name: CustomIOTDevices->Policy Assignment->Available Policies->CustomIOT and move to Assigned Polices

Logical Profiles List > CustomIOTDevices Logical Profile												
* Name CustomIOTDevices	Description											
* Policy Assignment												
Available Policies:		Assigned Policies:										
2Wire-Device	>	CustomIOT										
Aastra-Device	<											
Aerohive-Access-Point	>>											
Aerohive-Device American-Power-Conversion-Device	<<											
Android												

#### Step 13 Select Submit



#### Step 14 You should see the logical profile CustomIOTDevices

-thatte Identity Services Engine Home	Context Visibility ► Operations ▼Policy ► Admin	nistration	🔳 License Warning 🔺	ø
Policy Sets Profiling Posture Client Provisionin	ng			
Profiling	Logical Profiles			
٩			Selected 0   Total 10 😵 🎡	*
@ • E • @•	/ Edit 🕂 Add 🗙 Delete 🕂 Restore Cisco Provided Log	ical Profiles	Show All	5
Profiling Policies	Logical Profiles	System Type	Description	
	Cameras	Cisco Provided	Default logical profile for cameras.	
	CustomIOTDevices	Administrator Created		
	Gaming Devices	Cisco Provided	Default logical profile for gaming devices.	
	Home Network Devices	Cisco Provided	Default logical profile for home network devices.	
	IOT1Devices	Administrator Created		
	IP-Phones	Cisco Provided	Default logical profile for IP Phones.	
	Infrastructure Network Devices	Cisco Provided	Default logical profile for infrastructure network devices.	
	Medical Devices	Cisco Provided	Default logical profile for medical devices.	
	Mobile Devices	Cisco Provided	Default logical profile for mobile devices.	
	Printers	Cisco Provided	Default logical profile for printers.	

#### Step 15 Run script /run\_publisher38.sh

Johns-MacBook-Pro-2:api_simulator jeppich\$ ./run_publisher38.sh										
<pre></pre>										
2018-04-15 00:16:17.854 INFO 7211 [ main] c.c.p.s	amples.ise.http.CustomPublisher : Starting									
CustomPublisher v2.0.0-SNAPSHOT on johns-macbook-pro.lab10.com w	ith PID 7211									
(/Applications/api_partner_fc3/api_simulator/pxgrid-rest-ws-samp	les-2.0.0-SNAPSHOT.jar started by jeppich in									
/Applications/api_partner_fc3/api_simulator)										
2018-04-15 00:16:17.910 INFO 7211 [ main] c.c.p.s.	amples.ise.http.CustomPublisher : No active									
profile set, failing back to default profiles: default	figEmboddodWoblanligstionContout .									
Refreshing org springframework boot context embedded AnnotationCo	nrigEmbeddedWebApplicationContext@bebdb06.									
startup date [Sun Apr 15 00:16:18 EDT 2018]: root of context hie	rarchy									
2018-04-15 00:16:25.288 INFO 7211 [ main] s.b.c.e	.t.TomcatEmbeddedServletContainer : Tomcat									
initialized with port(s): 8080 (http)										
2018-04-15 00:16:25.416 INFO 7211 [ main] o.apach	e.catalina.core.StandardService : Starting									
service [Tomcat]										
2018-04-15 00:16:25.438 INFO 7211 [ main] org.apa	che.catalina.core.StandardEngine : Starting									
Servlet Engine: Apache Tomcat/8.5.16										
2018-04-15 00:16:25.889 INFO 7211 [ost-startStop-1] o.a.c.c	.C.[Tomcat].[localhost].[/] :									
Initializing Spring embedded WebApplicationContext										
2018-04-15 00:16:25.890 INFO 7211 [ost-startStop-1] o.s.web	.context.ContextLoader : Root									
WebApplicationContext: initialization completed in 7506 ms	and a constate provide the base of the second									
2018-04-15 00:16:26.113 INFO /211 [OST-StartStop-1] O.S.D.W	.servlet.ServletRegistrationBean : Mapping									
2018 04 15 00.16.26 120 INFO 7211 Logt startstop 11 o g b H	correlat FilterPogistrationPoan . Manning									
filter: 'characterEncodingEilter' to: [/*]	.serviet.filterkegistrationbean : Mapping									
2018-04-15 00.16.26 130 INFO 7211 [ost-startStop-1] o s b w	servlet FilterRegistrationBean · Manning									
filter: 'hiddenHttpMethodFilter' to: [/*]	.service.rifterkegistrationsean · happing									
2018-04-15 00:16:26.130 INFO 7211 [ost-startStop-1] 0.5.b.w	.servlet.FilterRegistrationBean : Mapping									
filter: 'httpPutFormContentFilter' to: [/*]										
2018-04-15 00:16:26.130 INFO 7211 [ost-startStop-1] o.s.b.w	.servlet.FilterRegistrationBean : Mapping									
filter: 'requestContextFilter' to: [/*]										
2018-04-15 00:16:26.851 INFO 7211 [ main] s.w.s.m	.m.a.RequestMappingHandlerAdapter : Looking									
for @ControllerAdvice:										
org.springframework.boot.context.embedded.AnnotationConfigEmbedd	edWebApplicationContext@bebdb06: startup date									
[Sun Apr 15 00:16:18 EDT 2018]; root of context hierarchy										
2018-04-15 00:16:27.035 INFO 7211 [ main] s.w.s.m	.m.a.RequestMappingHandlerMapping : Mapped									
['{[/getAssets]}' onto public com.cisco.pxgrid.samples.ise.http.D	eviceList									
Com.clsco.pxgrid.samples.ise.nttp.PublisherController.device()	m a RequestManningHandlerManning . Mary -									
2018-04-15 00:10:2/.039 INFO /211 [ Main] S.W.S.M	.m.a.kequestMappingHandlerMapping : Mapped									
[ {[/error]} Onco public org.springinamework.nttp.ResponseEntity	<pre>&gt; Java.utii.map&lt; Java.iang.Stiing,</pre>									



java.lang.Object>>		
org.springframework.boot.autoconfigure.web.BasicErrorCo	ntroller.error(javax.servlet.http.HttpServletReg	uest)
2018-04-15 00:16:27.039 INFO 7211 [ main	] s.w.s.m.m.a.RequestMappingHandlerMapping : Map	ped
"{[/error],produces=[text/html]}" onto public org.sprir	gframework.web.servlet.ModelAndView	
org.springframework.boot.autoconfigure.web.BasicErrorCo	ntroller.errorHtml(javax.servlet.http.HttpServle	tReque
<pre>st,javax.servlet.http.HttpServletResponse)</pre>		
2018-04-15 00:16:27.080 INFO 7211 [ main	] o.s.w.s.handler.SimpleUrlHandlerMapping : Map	ped
URL path [/webjars/**] onto handler of type [class		
org.springframework.web.servlet.resource.ResourceHttpRe	questHandler]	
2018-04-15 00:16:27.080 INFO 7211 [ main	] o.s.w.s.handler.SimpleUrlHandlerMapping : Map	ped
URL path [/**] onto handler of type [class		
org.springframework.web.servlet.resource.ResourceHttpRe	questHandler]	
2018-04-15 00:16:27.153 INFO 7211 [ main	] o.s.w.s.handler.SimpleUrlHandlerMapping : Map	ped
URL path [/**/favicon.ico] onto handler of type [class		
org.springiramework.web.serviet.resource.ResourceHttpRe	questHandler	
2018-04-15 00:16:27.508 INFO 7211 [ main	] o.s.j.e.a.AnnotationMBeanExporter :	
Registering beans for JMX exposure on startup		
2018-04-15 00:16:27.847 INFO /211 [ main	] s.b.c.e.t.TomcatEmbeddedServletContainer : Tom	cat
started on port(s): 8080 (nttp)	le sur semiles des blue GuelemBublishers (Gle	
2018-04-15 00:16:27.858 INFO /211 [ main	] C.C.p.samples.ise.nttp.CustomPublisher : Sta	τεα
customPublisher in 13.385 seconds (JVM running for 21.8	49)	
heathermone-ice24fe2_leb10_com		
nostnames-isezaics.iabio.com		
username-1011		
groung=Soggion		
description=null		
keystoreFilename=Johns_Machook_Pro lab10 com Johns_Ma	chook_Pro lab10 com p12	
keystorePassword=Ciscol23		
<pre>truststoreFilename=Johns-Macbook-Pro.labl0.com Johns-</pre>	Macbook-Pro, lab10.com.p12	
truststoreFilename=Johns-Macbook-Pro.1ab10.com_Johns- truststorePassword=Cisco123	Macbook-Pro.lab10.com.p12	
truststoreFilename=Johns-Macbook-Pro.lab10.com_Johns- truststorePassword=Cisco123	Macbook-Pro.labl0.com.pl2	
truststoreFilename=Johns-Macbook-Pro.labl0.com_Johns- truststorePassword=Cisco123 	Macbook-Pro.lab10.com.p12	
truststoreFilename=Johns-Macbook-Pro.labl0.com_Johns- truststorePassword=Ciscol23  2018-04-15 00:16:30.642 INFO 7211 [ mair Request={}	Macbook-Pro.lab10.com.p12	
truststoreFilename=Johns-Macbook-Pro.1ab10.com_Johns- truststorePassword=Cisco123 	Macbook-Pro.lab10.com.p12	
truststoreFilename=Johns-Macbook-Pro.labl0.com_Johns- truststorePassword=Ciscol23 	<pre>Macbook-Pro.lab10.com.p12  ] c.c.p.samples.ise.http.PxgridControl : ] c.c.p.samples.ise.http.PxgridControl : } ion=2 0.0.13</pre>	
truststoreFilename=Johns-Macbook-Pro.1ab10.com_Johns- truststorePassword=Cisco123 	<pre>Macbook-Pro.lab10.com.p12  ] c.c.p.samples.ise.http.PxgridControl : ] c.c.p.samples.ise.http.PxgridControl : } ion=2.0.0.13 </pre>	
truststoreFilename=Johns-Macbook-Pro.1ab10.com_Johns- truststorePassword=Cisco123 	<pre>Macbook-Pro.lab10.com.p12  ] c.c.p.samples.ise.http.PxgridControl : ] c.c.p.samples.ise.http.PxgridControl : } ion=2.0.0.13 ] c.c.p.samples.ise.http.PxgridControl : ; {"wsPubsubService":"com.cisco.ise.pubsub" "rest"</pre>	Baselip
<pre>truststoreFilename=Johns-Macbook-Pro.labl0.com_Johns- truststorePassword=Ciscol23 </pre>	<pre>Macbook-Pro.labl0.com.pl2  ] c.c.p.samples.ise.http.PxgridControl : ] c.c.p.samples.ise.http.PxgridControl : } ion=2.0.0.13 ] c.c.p.samples.ise.http.PxgridControl : :{"wsPubsubService":"com.cisco.ise.pubsub","rest: Lcisco.endpoint.asset"} </pre>	BaseUR
truststoreFilename=Johns-Macbook-Pro.1ab10.com_Johns- truststorePassword=Cisco123 	<pre>Macbook-Pro.labl0.com.pl2  ] c.c.p.samples.ise.http.PxgridControl : ] c.c.p.samples.ise.http.PxgridControl : }  ion=2.0.0.13 ] c.c.p.samples.ise.http.PxgridControl : :{"wsPubsubService":"com.cisco.ise.pubsub","rest: a.cisco.endpoint.asset"}} </pre>	BaseUR
truststoreFilename=Johns-Macbook-Pro.1ab10.com_Johns- truststorePassword=Cisco123 	<pre>Macbook-Pro.labl0.com.pl2  a] c.c.p.samples.ise.http.PxgridControl :</pre>	BaseUR
truststoreFilename=Johns-Macbook-Pro.1ab10.com_Johns- truststorePassword=Cisco123 	<pre>Macbook-Pro.labl0.com.pl2  a] c.c.p.samples.ise.http.PxgridControl :</pre>	BaseUR
truststoreFilename=Johns-Macbook-Pro.1ab10.com_Johns- truststorePassword=Cisco123 	<pre>Macbook-Pro.labl0.com.pl2  a] c.c.p.samples.ise.http.PxgridControl :</pre>	BaseUR
truststoreFilename=Johns-Macbook-Pro.1ab10.com_Johns- truststorePassword=Cisco123  2018-04-15 00:16:30.642 INFO 7211 [ mair Request={} 2018-04-15 00:16:30.740 INFO 7211 [ mair Response={"accountState":"ENABLED","version":"2.0.0.13" 15-Apr-18 00:16:30.741 [main-1]: pxGrid controller vers 2018-04-15 00:16:30.750 INFO 7211 [ mair Request={"name":"com.cisco.endpoint.asset","properties" L":"http://raghdasa-lnv1:8080","assetTopic":"/topic/com 2018-04-15 00:16:30.906 INFO 7211 [ mair Response={} 2018-04-15 00:16:30.913 INFO 7211 [ mair Request={"name":"com.cisco.ise.pubsub"} 2018-04-15 00:16:30.924 INFO 7211 [ mair	<pre>Macbook-Pro.labl0.com.pl2  a] c.c.p.samples.ise.http.PxgridControl :</pre>	BaseUR
truststoreFilename=Johns-Macbook-Pro.1ab10.com_Johns- truststorePassword=Cisco123  2018-04-15 00:16:30.642 INFO 7211 [ mair Request={} 2018-04-15 00:16:30.740 INFO 7211 [ mair Response={"accountState":"ENABLED","version":"2.0.0.13" 15-Apr-18 00:16:30.741 [main-1]: pxGrid controller vers 2018-04-15 00:16:30.750 INFO 7211 [ mair Request={"name":"com.cisco.endpoint.asset","properties" L":"http://raghdasa-lnv1:8080","assetTopic":"/topic/com 2018-04-15 00:16:30.906 INFO 7211 [ mair Response={} 2018-04-15 00:16:30.913 INFO 7211 [ mair Request={"name":"com.cisco.ise.pubsub"} 2018-04-15 00:16:30.924 INFO 7211 [ mair Response={"services":[{"name":"com.cisco.ise.pubsub"}"."r	<pre>Macbook-Pro.labl0.com.pl2  ] c.c.p.samples.ise.http.PxgridControl : ] ion=2.0.0.13 ] c.c.p.samples.ise.http.PxgridControl : :{"wsPubsubService":"com.cisco.ise.pubsub","rest: .cisco.endpoint.asset"}} ] c.c.p.samples.ise.http.PxgridControl : !] c.c.p.samples.ise.http.PxgridContr</pre>	BaseUR
truststoreFilename=Johns-Macbook-Pro.1ab10.com_Johns- truststorePassword=Cisco123 	<pre>Macbook-Pro.labl0.com.p12  ] c.c.p.samples.ise.http.PxgridControl : ] c.c.p.samples.ise.http.PxgridControl : } dion=2.0.0.13 ] c.c.p.samples.ise.http.PxgridControl : :{"wsPubsubService":"com.cisco.ise.pubsub","rest: a.cisco.endpoint.asset"}} ] c.c.p.samples.ise.http.PxgridControl : d] c.c.p.samples.ise.http.PxgridCont</pre>	BaseUR
truststoreFilename=Johns-Macbook-Pro.1ab10.com_Johns- truststorePassword=Cisco123 	<pre>Macbook-Pro.labl0.com.pl2  I] c.c.p.samples.ise.http.PxgridControl : I] c.c.p.samples.ise.http.PxgridControl : I] c.c.p.samples.ise.http.PxgridControl : I[ wsPubsubService":"com.cisco.ise.pubsub","rest: I.cisco.endpoint.asset"}} I] c.c.p.samples.ise.http.PxgridControl : I] c.c</pre>	BaseUR
truststoreFilename=Johns-Macbook-Pro.1ab10.com_Johns- truststorePassword=Cisco123 	<pre>Macbook-Pro.labl0.com.pl2  I] c.c.p.samples.ise.http.PxgridControl : I] c.c.p.samples.ise.http.PxgridControl</pre>	BaseUR
<pre>truststoreFilename=Johns-Macbook-Pro.lab10.com_Johns- truststorePassword=Ciscol23 </pre>	<pre>Macbook-Pro.labl0.com.pl2  I] c.c.p.samples.ise.http.PxgridControl : I] c.c.p.samples.ise.http.PxgridControl</pre>	BaseUR
truststoreFilename=Johns-Macbook-Pro.1ab10.com_Johns- truststorePassword=Ciscol23 	<pre>Macbook-Pro.labl0.com.pl2  I] c.c.p.samples.ise.http.PxgridControl : I] c.c.p.samples.ise.http.PxgridControl : I] c.c.p.samples.ise.http.PxgridControl : I[ wsPubsubService":"com.cisco.ise.pubsub","rest: I.cisco.endpoint.asset"}} I] c.c.p.samples.ise.http.PxgridControl : I] c.c</pre>	BaseUR
truststoreFilename=Johns-Macbook-Pro.1ab10.com_Johns- truststorePassword=Cisco123 	<pre>Macbook-Pro.labl0.com.pl2  I] c.c.p.samples.ise.http.PxgridControl : I] c.c.p.samples.ise.http.PxgridControl : I] c.c.p.samples.ise.http.PxgridControl : I[ wsPubsubService":"com.cisco.ise.pubsub","rest: I.cisco.endpoint.asset"}} I] c.c.p.samples.ise.http.PxgridControl : I] c.c</pre>	BaseUR
truststoreFilename=Johns-Macbook-Pro.1ab10.com_Johns- truststorePassword=Ciscol23  2018-04-15 00:16:30.642 INFO 7211 [ mair Request={} 2018-04-15 00:16:30.740 INFO 7211 [ mair Response={"accountState":"ENABLED","version":"2.0.0.13" 15-Apr-18 00:16:30.741 [main-1]: pxGrid controller vers 2018-04-15 00:16:30.750 INFO 7211 [ mair Request={"name":"com.cisco.endpoint.asset","properties" L":"http://raghdasa-lnv1:8080","assetTopic":"/topic/com 2018-04-15 00:16:30.906 INFO 7211 [ mair Response={} 2018-04-15 00:16:30.913 INFO 7211 [ mair Response={} 2018-04-15 00:16:30.924 INFO 7211 [ mair Response={"services":{"wsUrl":"wss://ise24fc3.lab10.c 15-Apr-18 00:16:30.924 [main-1]: wsUrl=wss://ise24fc3.lab10.c 15-Apr-18 00:16:30.924 [main-1]: wsUrl=wss://ise24fc3.lab10.c 15-Apr-18 00:16:30.929 INFO 7211 [ mair Request={"peerNodeName":"ise-pubsub-ise24fc3"} 2018-04-15 00:16:30.955 INFO 7211 [ mair Response={"secret":"tASXtHAbKDKbAODh"} 2018-04-15 00:16:33.757 INFO 7211 [ mair	<pre>Macbook-Pro.labl0.com.p12  ] c.c.p.samples.ise.http.PxgridControl : ] ion=2.0.0.13 ] c.c.p.samples.ise.http.PxgridControl : :{"wsPubsubService":"com.cisco.ise.pubsub","rest: .cisco.endpoint.asset"}} ] c.c.p.samples.ise.http.PxgridControl : !] c.c.p.sinples.ise.http.PxgridControl : !] c.c.p.sinples.ise.http.PxgridContr</pre>	BaseUR
truststoreFilename=Johns-Macbook-Pro.1ab10.com_Johns- truststorePassword=Cisco123  2018-04-15 00:16:30.642 INFO 7211 [ mair Request={} 2018-04-15 00:16:30.740 INFO 7211 [ mair Response={"accountState":"ENABLED","version":"2.0.0.13" 15-Apr-18 00:16:30.741 [main-1]: pxGrid controller vers 2018-04-15 00:16:30.750 INFO 7211 [ mair Request={"name":"com.cisco.endpoint.asset","properties" L":"http://raghdasa-lnv1:8080","assetTopic":"/topic/com 2018-04-15 00:16:30.906 INFO 7211 [ mair Response={} 2018-04-15 00:16:30.913 INFO 7211 [ mair Response={} 2018-04-15 00:16:30.924 INFO 7211 [ mair Response={"services":[{"name":"com.cisco.ise.pubsub"} 2018-04-15 00:16:30.924 INFO 7211 [ mair Response={"services":[{"wSUrl":"wSs://ise24fc3.lab10.c 15-Apr-18 00:16:30.929 INFO 7211 [ mair Request={"peerNodeName":"ise-pubsub-ise24fc3"} 2018-04-15 00:16:30.929 INFO 7211 [ mair Response={"secret":"tASXtHAbKDKbAODh"} 2018-04-15 00:16:33.757 INFO 7211 [ mair Response={"secret":"tASXtHAbKDKbAODh"} 2018-04-15 00:16:33.757 INFO 7211 [ mair	<pre>Macbook-Pro.labl0.com.pl2  I] c.c.p.samples.ise.http.PxgridControl : I] c.c.p.samples.ise.http.PxgridControl : I] c.c.p.samples.ise.http.PxgridControl : I[ wsPubsubService":"com.cisco.ise.pubsub","rest: I.cisco.endpoint.asset"} I] c.c.p.samples.ise.http.PxgridControl : I] c.c.p.si.h.StompPubsubClientEndpoint : WS of the store is the</pre>	BaseUR DnOpen 4P
truststoreFilename=Johns-Macbook-Pro.1ab10.com_Johns- truststorePassword=Ciscol23  2018-04-15 00:16:30.642 INFO 7211 [ mair Request={} 2018-04-15 00:16:30.740 INFO 7211 [ mair Response={"accountState":"ENABLED","version":"2.0.0.13" 15-Apr-18 00:16:30.741 [main-1]: pxGrid controller vers 2018-04-15 00:16:30.750 INFO 7211 [ mair Request={"name":"com.cisco.endpoint.asset","properties" L":"http://raghdasa-lnv1:8080","assetTopic":"/topic/com 2018-04-15 00:16:30.906 INFO 7211 [ mair Response={} 2018-04-15 00:16:30.913 INFO 7211 [ mair Response={} 2018-04-15 00:16:30.924 INFO 7211 [ mair Response={"services":[{"name":"com.cisco.ise.pubsub"} 2018-04-15 00:16:30.924 INFO 7211 [ mair Response={"services":[{"wSUrl":"wSs://ise24fc3.lab10.c 15-Apr-18 00:16:30.929 INFO 7211 [ mair Request={"peerNodeName":"ise-pubsub-ise24fc3"} 2018-04-15 00:16:30.929 INFO 7211 [ mair Response={"secret":"tASXtHAbKDKAODh"} 2018-04-15 00:16:33.757 INFO 7211 [ mair Response={"secret":"tASXtHAbKDKAODh"} 2018-04-15 00:16:33.760 INFO 7211 [ mair Response={"secret":"taSXtHAbKDKAODh"} 2018-04-15 00:16:33.760 INFO 7211 [ mair Response={"secret":"taSXtHAbKDKAODh"}	<pre>Macbook-Pro.labl0.com.p12  I] c.c.p.samples.ise.http.PxgridControl : I] c.c.p.samples.ise.http.PxgridControl : I] c.c.p.samples.ise.http.PxgridControl : I[ wsPubsubService":"com.cisco.ise.pubsub","rest: I.cisco.endpoint.asset"}} I] c.c.p.samples.ise.http.PxgridControl : I] c.c.p.sinh.StompPubsubClientEndpoint : WS of I] c.c.p.si.h.StompPubsubClientEndpoint : STOP </pre>	BaseUR DnOpen 4P
<pre>truststoreFilename=Johns-Macbook-Pro.lab10.com_Johns- truststorePassword=Ciscol23 </pre>	<pre>Macbook-Pro.labl0.com.p12  I] c.c.p.samples.ise.http.PxgridControl : I] c.c.p.samples.ise.http.PxgridControl : I] c.c.p.samples.ise.http.PxgridControl : I[ wsPubsubService":"com.cisco.ise.pubsub","rest: I.cisco.endpoint.asset"}} I] c.c.p.samples.ise.http.PxgridControl : I] c.c.p.sinh.StompPubsubClientEndpoint : WS of I] c.c.p.si.h.StompPubsubClientEndpoint : STOP 6:33.768 INFO 7211 [ Grizzly(2)]</pre>	BaseUR onOpen 4P

#### Step 16 Press Enter

ClientEndpoint : STOMP SEND topic=/topic/com.cisco.endpoint.asset

command=SEND, headers={'content-length':'591','destination':'/topic/com.cisco.endpoint.asset',}, content.length=591 15-Apr-18 00:19:00.175 [main-1]: {"assetHwRevision":"5.6","assetProtocol":"CIP","assetConnectedLinks":[{"value":"3","key":"indattr2"},{"value" :"Root","key":"assetGroup"},{"value":"1","key":"indattr3"}],"assetVendor":"Cisco Systems","assetSwRevision":"4.6","assetCustomAttributes":[{"value":"3","key":"indattr2"},{"value":"Root","key ":"assetGroup"},{"value":"1","key":"indattr3"}],"assetProductId":"IE2000","assetSerialNumber":"1212121213243" ,"assetMacAddress":"48:b2:d0:63:d1:32","assetId":"260","assetIpAddress":"56.56.217.16","assetName":"Abjergary n - 47","assetDeviceType":"EtherNet\/IP Node"}

Step 17 Select Context Visibility->Endpoints-Authentication or Endpoint Classification You will see the asset device

altala cisco	Identit	y Servic	es Engine	Hom	e 🗸 Context	Visibility Visibility	Operations	▶ Policy	► Administration	▶ Work Center	S			1	License W	/aming 🔺	ৎ 📀	0 4
End	points Authe	Users ntication	Network De	vices Ap	plication Compliance	Compromise	d Endpoints	Endpo	int Classification	Guest	Vulnerable En	ndpoints H	Click here to laroware	do wireless	setup and v	isibility setup	Do not show	r this again.
<	INAC	TIVE E	ENDPOIN	ITS 0		<b>9</b> 0	AUTH	ENTICAT	No data availa	• ble.	0	AUTHENT Failure Reason	Identity Store	Identity Gi ata availa	ble.	Ø	Ø	NETWO Location
				st Activity I	Jate							F	Rows/Page 1	•	◀ 1	©/1 ►	► Go 1	Total Rows
-	· +	C	ANC ·	- Char	ge Authorization	<ul> <li>Clear Threa</li> </ul>	ats & Vulnerat	ilities Ex	port - Import -	MDM Actions -	Release R	ejected Revok	e Certificate				<b>T</b> Filter	- <b>\$</b> -
(	MA	C Addre	SS	Status		IPv4 Address	s (	Jsername	Hostname	Location	Endpoint F	Profile A	uthentication F	Failure Re	ason	Authenticat	ion Policy	Authoriza
>	MA	C Addres	is	Status	•	IPv4 Address		Username	Hostname	Location	Endpoint P	rofile	Authentication Fa	ilure Reas	n	Authenticatio	on Policy	Authorizat
(	- 48	8:B2:D0:6	3:D1:32			56.56.217.16					IOT_Examp	le1						

Step 18 Select MAC address of the endpoint devices, then Attributes

You will see "AssetGroup" under "Attribute Name" and "Root" under "Attribute Value". Note the Logical Profile still shows IOT\_Example1.

dentity Services En	igine Home	- Context Visibility	<ul> <li>Operations</li> </ul>	Policy	Administration
Endpoints Users Netwo	ork Devices Applic	ation			
Static Assignment	false				
Endpoint Policy	IOT_Example1				
Static Group Assignment	false				
Identity Group Assignment	IOT_Example1				
Custom Attributes					
				▼ Fi	lter - 🗘 -
Attribute Name	)	Attribute Valu	e		
× Attribute Name		Attribute Value	•		
		Reat			
L ogioal Brofile		ROOL			
LogicalFiolile					
Other Attributes					
Other Attributes					
BYODRegistration	Unknown				
DeviceRegistrationStatus	NotRegistered				
ElapsedDays	0				
EndPointPolicy	IOT_Example1				
EndPointProfilerServer	ise24fc3.lab10.com				
EndPointSource	PXGRIDPROBE				
IdentityGroup	IOT_Example1				
InactiveDays	0				
LogicalProfile	IOT1Devices				

## Step 19Select Policy->Profiling->Profiling Policies->Quick filter->"iot"<br/>You will see both profiling policies

dentity Services Engine Home	Context Visibility → Operations → Policy	Administration     Work Centers	1 License Warning 🔺 🔍 🐵 💰
Policy Sets Profiling Posture Client Provision	ng		Click here to do wireless setup and visibility setup Do not show this again.
Profiling	Profiling Policies		
(م			Selected 0   Total 2 😵 🎡 🚽
<₽ ▼ ₩ ▼		Export V	Show Quick Filter
Profiling Policies	Profiling Policy Name	<ul> <li>Policy Enabled</li> <li>System Type</li> </ul>	Description
<ul> <li>Logical Profiles</li> </ul>	iot		
	CustomIOT	Enabled Administrator Create	1
	IOT_Example1	Disabled Administrator Creater	i l

#### Step 20 Select->IOT\_Example1->uncheck policy enabled



Step 21 Select Save

#### Step 22 Select-> Context Visibility->Endpoints or Endpoint Classification

You will now see the proper endpoint profile and logical profile.

altalta cisco	Ident	tity Servic	es Engine	Home 🔫 C	Context Vis	ibility 🕨	Operations	▶ Policy	Administration	Work Centers	3			1 L	icense Warni	ng 🔺	۹ (		ø
End	points	Users	Network Devic	es Application									Click here to	do wireless se	tup and visibi	litv setup	Do not sh	ow this agai	in. >
	Aut	hentication	BYOD	Complian	ice	Compromise	ed Endpoints	Endpoir	nt Classification	Guest	Vulnerable En	ndpoints H	laroware			.,,		4	<del>р.</del> -
<	INA(	CTIVE E	ENDPOINT Last /	S 🛛		e o	AUTH	IENTICAT	No data availab	e.	G 0	AUTHEN Failure Reason	Identity Store	Identity Group	e.	Ø	Q	NETW Location	0
													Rows/Page 1	• • •	1 0	/1 🕨	Go	1 Total Rov	WS
-	° +	ß	m ANC ▼	Change Author	rization -	Clear Thr	eats & Vulnera	ibilities Exp	oort - Import -	MDM Actions -	Release R	ejected Revol	ke Certificate				▼ Filt	er≖ ¢	*
C	M	AC Addre	ss s	Status	I	Pv4 Addres	SS	Username	Hostname	Location	Endpoint F	Profile A	Authentication I	ailure Reas	on Aut	henticati	on Policy	Autho	riza
>	×	AC Addres	is	Status	•	IPv4 Addres	s	Username	Hostname	Location	Endpoint P	rofile	Authentication Fa	ailure Reason	Aut	henticatio	n Policy	Autho	rizat
0		48:B2:D0:6	3:D1:32		5	6.56.217.16					CustomIOT								

# **Creating Authorization Policy Based on Asset's Logical Profile for Custom Attributes**

An authorization policy will be created that determines the asset's network access. A Security Group Tag (SGT) will also be added to the Authorization Policy provided label to classify network traffic if Cisco's TrustSec Solution is used.

 Step 1
 Create IOT device SGT which will be used in the Authorization Policy
 Select WorkCenters->Trustsec->Components->Add->CustomIOTDevices

* Name Custor	nIOT	Device	s			
* Icon						
	Ļ		1	$\times$	•	•
A			2	>		0
<b>P</b>	٥			١	۵	4
Descrip	tion					

- Step 2 Select Save
- **Step 3** Create ISE Authorization Profile Policy
- Step 4 Select Policy->Policy Sets->View ">"->Authorization Policy
- **Step 5** You should see:

disco	Identity Se	rvices Engine	Home >	Context Vi	sibility	<ul> <li>Operations</li> </ul>	→ Policy	<ul> <li>Administration</li> </ul>	♦ World	Centers		1	License Warni	ng 🔺	0	0	•	
Polic	y Sets Pro	filing Posture	Client Provisioni	ing Pol	icy Eleme	ents												
		,,																
> A	uthorizatio	n Policy - Glob	al Exceptions	(1)														
~	uthorizatio	n Policy (14)																
	athonizatio	(14)																
6										Results								
	Status	Rule Name		Condi	itions					Profiles		Security Gro	oups			Hits	Actions	
Se	arch																	
	Ø	MDM		ĥ	MDM·M	IDMServerName I	EQUALS Ger	mantown03		× PermitAccess	+	BYOD		<b>x</b> •	+	0	٥	
						Wireless_Acces	S											
	Ø	Wireless Black	List Default	AND	<u>188</u>	IdentityGroup-Na Groups:Blacklist	ame EQUAL	S Endpoint Identity		×Blackhole_Wireless_Access	+	Select from	list	Ŧ	+	0	¢	
	Ø	Profiled Cisco	IP Phones	484	Identity Groups:	ntityGroup-Name EQUALS Endpoint Identity oups:Profiled:Cisco-IP-Phone				Cisco_IP_Phones	+	Select from	list	Ŧ	+	0	¢	



#### Step 6 Under Actions, Click on "gear" and "Insert new rule above

**Step 7** You should see the following:

dialo Id	entity Serv	v <mark>ices Engine Ho</mark> me ► C	ontext Visibility	- Policy	Administration	Nork Centers	1	License Warning 🔺	୍
Policy Se	ets Profi	ing Posture Client Provisioning	<ul> <li>Policy Elements</li> </ul>						
Auth	norization	Policy - Local Exceptions							
> Auth	norization	Policy - Global Exceptions (1	)						
✓ Auth	norization	Policy (15)							
						Results			
+	Status	Rule Name	Conditions			Profiles	Security Gro	ups	
Search	h								
1	Ø	Authorization Rule 1		+		Select from list	Select from I	ist 🔻	r +
				-					

- Step 8 Name Authorization Rule 1 to CustomIOTDevices, Under Condition, select "+"
- **Step 9** Click on **Attribute**, you should see the following Dictionary Attributes

Conditions Studio				0
Library		Editor		
Search by Name		ų	Click to add an attribute	
♥ ☶ □ 卷 ⊕ 및 ⊈ 땀 ◙ 7 ▣ 안 ↓	© ¦: 奈		Select attribute for condition	×
BYOD_is_Registered	(i)		♀ □□ ▲ ● ♀ ■ P □ ▷ □ ○ ▲ ● ↓	((:-
Catalyst_Switch_Local_Web_Authentication	on 🕡		Dictionary Attribute ID Info	
Compliance_Unknown_Devices	()		All Dictionaries    Attribute  ID	
Compliant Devices			Respace         Aire-Data-Bandwidth-Average         7         ()	
	Ŭ		Respace         Aire-Data-Bandwidth-Average         13         ()	
EAP-MSCHAPv2	<i>i</i>		Airespace Aire-Data-Bandwidth-Burst-Do 9	



Conditions Studio				<b>3</b> × 6
Library	Editor			
Search by Name		EndPoints-assetGroup		8
	٤	Contains   Root		
BYOD_is_Registered ()		Set to 'Is not' Duplica	te Sa	ave

Step 11 Select Use

- Step 12 Under Profiles, select Permit Access
- Step 13Under Security Groups, select CustomIOTDevices<br/>You should see:

diale Id	entity Serv	vices Engine	Home	Context Vis	ibility > Operations	→ Policy	Administration	Work Cer	nters		1	License Warni	ng 🔺	् 🛛	•	þ
Policy Se	ets Profil	ing Posture	Client Provisi	ioning   Poli	cy Elements					Click h	ere to do wireles	s setup and visibi	lity setup D	o not show	this again.	>
> Auth	orization F	Policy - Local Ex	ceptions													
> Auth	orization F	Policy - Global E	xceptions													
✓ Auth	orization F	Policy (14)														
									Results							
+	Status	Rule Name		Condit	ions				Profiles		Security Gro	oups		Hits	Action	5
Search	1															
1	Ø	CustomIOT_Dev	vices	Ë	EndPoints-assetGroup C	ONTAINS Ro	oot		* PermitAccess	+	CustomIOTI	Devices	× - +	0	¢	



#### Verifying Asset as Defined by the Logical Profile for Custom Attributes

The custom asset policy is now assigned to the logical profile. This is verified in the Context Visibility Endpoint Classification screen.



```
___
                          (v1.5.6.RELEASE)
 :: Spring Boot ::
2018-04-14 15:42:10.009 INFO 5926 --- [
                                                   main] c.c.p.samples.ise.http.CustomPublisher
                                                                                                   : Starting
CustomPublisher v2.0.0-SNAPSHOT on johns-macbook-pro.lab10.com with PID 5926
(/Applications/api_partner_fc3/api_simulator/pxgrid-rest-ws-samples-2.0.0-SNAPSHOT.jar started by jeppich in
/Applications/api_partner_fc3/api_simulator)
2018-04-14 15:42:10.042 INFO 5926 --- [
                                                   main] c.c.p.samples.ise.http.CustomPublisher
                                                                                                   : No active
profile set, falling back to default profiles: default
2018-04-14 15:42:10.508 INFO 5926 --
                                                   main] ationConfigEmbeddedWebApplicationContext :
                                     - [
Refreshing org.springframework.boot.context.embedded.AnnotationConfigEmbeddedWebApplicationContext@bebdb06:
startup date [Sat Apr 14 15:42:10 EDT 2018]; root of context hierarchy
2018-04-14 15:42:17.010 INFO 5926 --
                                                   main] s.b.c.e.t.TomcatEmbeddedServletContainer : Tomcat
initialized with port(s): 8080 (http)
2018-04-14 15:42:17.136 INFO 5926 --- [
                                                   main] o.apache.catalina.core.StandardService
                                                                                                   : Starting
service [Tomcat]
2018-04-14 15:42:17.158 INFO 5926 --- [
                                                   main] org.apache.catalina.core.StandardEngine
                                                                                                  : Starting
Servlet Engine: Apache Tomcat/8.5.16
2018-04-14 15:42:17.587 INFO 5926 --- [ost-startStop-1] o.a.c.c.C.[Tomcat].[localhost].[/]
                                                                                                   :
Initializing Spring embedded WebApplicationContext
2018-04-14 15:42:17.588 INFO 5926 --- [ost-startStop-1] o.s.web.context.ContextLoader
                                                                                                   : Root
WebApplicationContext: initialization completed in 7105 ms
2018-04-14 15:42:17.813 INFO 5926 --- [ost-startStop-1] o.s.b.w.servlet.ServletRegistrationBean
                                                                                                  : Mapping
servlet: 'dispatcherServlet' to [/]
2018-04-14 15:42:17.828 INFO 5926 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean
                                                                                                   : Mapping
filter: 'characterEncodingFilter' to: [/*]
2018-04-14 15:42:17.829 INFO 5926 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean
                                                                                                   : Mapping
filter: 'hiddenHttpMethodFilter' to: [/*]
```

# cisco.

2018-04-14 15:42:17.829 INFO 5926 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'httpPutFormContentFilter' to: [/\*] 2018-04-14 15:42:17.830 INFO 5926 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'requestContextFilter' to: [/\*] 2018-04-14 15:42:18.471 INFO 5926 --- [ main] s.w.s.m.m.a.RequestMappingHandlerAdapter : Looking for @ControllerAdvice: org.springframework.boot.context.embedded.AnnotationConfigEmbeddedWebApplicationContext@bebdb06: startup date [Sat Apr 14 15:42:10 EDT 2018]; root of context hierarchy 2018-04-14 15:42:18.653 INFO 5926 --- [ main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/getAssets]}" onto public com.cisco.pxgrid.samples.ise.http.DeviceList com.cisco.pxgrid.samples.ise.http.PublisherController.device() 2018-04-14 15:42:18.657 INFO 5926 --- [ main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/error]}" onto public org.springframework.http.ResponseEntity<java.util.Map<java.lang.String, java.lang.Object>> org.springframework.boot.autoconfigure.web.BasicErrorController.error(javax.servlet.http.HttpServletRequest) 2018-04-14 15:42:18.658 INFO 5926 --- [ main] s.w.s.m.m.a.RequestMappingHandlerN "{[/error],produces=[text/html]}" onto public org.springframework.web.servlet.ModelAndView main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped org.springframework.boot.autoconfigure.web.BasicErrorController.errorHtml(javax.servlet.http.HttpServletReque st,javax.servlet.http.HttpServletResponse) 2018-04-14 15:42:18.692 INFO 5926 -main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/webjars/\*\*] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler] main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped 2018-04-14 15:42:18.692 INFO 5926 --URL path [/\*\*] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler] 2018-04-14 15:42:18.762 INFO 5926 --- [ main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/\*\*/favicon.ico] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler] 2018-04-14 15:42:19.129 INFO 5926 --- [ main] o.s.j.e.a.AnnotationMBeanExporter : Registering beans for JMX exposure on startup 2018-04-14 15:42:19.434 INFO 5926 --- [ main] s.b.c.e.t.TomcatEmbeddedServletContainer : Tomcat started on port(s): 8080 (http) 2018-04-14 15:42:19.446 INFO 5926 --- [ main] c.c.p.samples.ise.http.CustomPublisher : Started CustomPublisher in 11.951 seconds (JVM running for 18.438) ----- properties -----hostnames=ise24fc3.lab10.com username=IOT1 password=null groups=Session description=null keystoreFilename=Johns-Macbook-Pro.lab10.com Johns-Macbook-Pro.lab10.com.pl2 keystorePassword=Cisco123 truststoreFilename=Johns-Macbook-Pro.lab10.com Johns-Macbook-Pro.lab10.com.pl2 truststorePassword=Cisco123 \_\_\_\_\_ 2018-04-14 15:42:21.139 INFO 5926 --- [ main] c.c.p.samples.ise.http.PxgridControl : Request={} 2018-04-14 15:42:21.209 INFO 5926 --- [ main] c.c.p.samples.ise.http.PxgridControl : Response={"accountState":"ENABLED", "version": "2.0.0.13"} 14-Apr-18 15:42:21.210 [main-1]: pxGrid controller version=2.0.0.13 2018-04-14 15:42:21.219 INFO 5926 --- [ main] c.c.p.samples.ise.http.PxgridControl Request={"name":"com.cisco.endpoint.asset","properties":{"wsPubsubService":"com.cisco.ise.pubsub","restBaseUR
L":"http://raghdasa-lnv1:8080","assetTopic":"/topic/com.cisco.endpoint.asset"}} 2018-04-14 15:42:21.369 INFO 5926 --- [ main] c.c.p.samples.ise.http.PxgridControl : Response={} 2018-04-14 15:42:21.375 INFO 5926 --- [ main] c.c.p.samples.ise.http.PxgridControl • Request={"name":"com.cisco.ise.pubsub"} 2018-04-14 15:42:21.386 INFO 5926 --- [ main] c.c.p.samples.ise.http.PxgridControl . Response={"services":[{"name":"com.cisco.ise.pubsub","nodeName":"ise-pubsubise24fc3", "properties":{"wsUrl":"wss://ise24fc3.lab10.com:8910/pxgrid/ise/pubsub"}}]} **14-Apr-18** 15:42:21.386 [main-1]: wsUrl=wss://ise24fc3.lab10.com:8910/pxgrid/ise/pubsub 2018-04-14 15:42:21.392 INFO 5926 --- [ main] c.c.p.samples.ise.http.Pxgrid main] c.c.p.samples.ise.http.PxgridControl Request={"peerNodeName":"ise-pubsub-ise24fc3"} 2018-04-14 15:42:21.429 INFO 5926 --- [ main] c.c.p.samples.ise.http.PxgridControl : Response={"secret":"tASXtHAbKDKbAODh"} 2018-04-14 15:42:22.692 INFO 5926 --- [ Grizzly(1)] c.c.p.s.i.h.StompPubsubClientEndpoint : WS onOpen 2018-04-14 15:42:22.695 INFO 5926 --- [ main] c.c.p.s.i.h.StompPubsubClientEndpoint : STOMP CONNECT host=ise24fc3.lab10.com press <enter> to start the publishing...2018-04-14 15:42:22.703 INFO 5926 --- [ Grizzly(2)] : STOMP CONNECTED version=1.2 c.c.p.s.i.h.StompPubsubClientEndpoint 2018-04-14 15:42:29.801 INFO 5926 --- [ main] c.c.p.s.i.h.StompPubsubClientEndpoint : STOMP SEND topic=/topic/com.cisco.endpoint.asset

command=SEND, headers={'content-length':'593','destination':'/topic/com.cisco.endpoint.asset',},
content.length=593
14-Apr-18 15:42:29.802 [main-1]:
{"assetHwRevision":"5.6","assetProtocol":"CIP","assetConnectedLinks":[{"value":"3","key":"indattr2"},{"value"
'"Root","key":"assetGroup"},{"value":"1","key":"indattr3"}],"assetVendor":"Cisco
Systems","assetSwRevision":"4.6","assetCustomAttributes":[{"value":"3","key":"indattr2"},{"value":"Root","key":"indattr2"},{"value":"1","key":"indattr3"}],"assetProductId":"IE2000","assetSerialNumber":"121212121212123243"
,"assetMacAddress":"b0:2c:27:93:fe:94","assetId":"215","assetIpAddress":"125.84.172.120","assetName":"Abjerga
ryn - 49","assetDeviceType":"EtherNet\/IP Node"}

Step 2 Select->Context Visibility->Endpoint->Endpoint Classification

Note that the client endpoint is now assigned to the CustomIOT profile.

iden	tity Services Eng	ine Ho	ome 👻 Co	ontext Visibility	Operations	Policy      Adi	ministration	Work Centers			1 License Warning 4	্ ০	0	0	ø
Endpoints	Users Networ	k Devices	Application							Click here to do v	vireless setup and visibility se	tup Do r	not show th	his again.	>
Aut	hentication	BYOD	Complianc	e Comprom	ised Endpoints	Endpoint Class	sification	Guest Vulnerable	Endpoints	Hardware				₽. ₽	Ŧ
	POINTS <sup>1</sup>			<b>P</b> C		DINT CATEGO S Types Identity Gr		0 9	NETW Location	ORK DEVICES <sup>(3)</sup> Type Device Name	t	9 Ø		-	-
										No data	available.				
	misc: (100%)_	C			unk	nown: [100%]	$\mathbf{O}$			Rows/Page 2			Go 2 Tr	otal Rows	8
c +	· © 💼 /	ANC - Ch	nange Authori:	zation - Clear T	nreats & Vulnerabil	lities Export -	Import -	IDM Actions - Release	Rejected F	tevoke Certificate			▼ Filter -	۰.	
M	AC Address	Anom	nalous Il	Pv4 Address	Username	Hostname	Location	Endpoint Profile	Desc	ription	OUI		OS Type	es	
×	AC Address	Anom	nalous Be	Pv4 Address	Username	Hostname	Location	Endpoint Profile	Desc	ription	OUI		OS Type	es	
	94:D4:CB:57:CA:7	с	0	.163.69.187				CustomIOT			UNKNOWN				

**Step 3** If you click on the MAC address, you will see the logical profile of CustomIOT devices. You may also see IOTDevices if you did not remove the CutomIOT device policy from IOT devices logical profile.

dentity Services	Engine Home	✓ Context Visibility	<ul> <li>Operations</li> </ul>	Policy	Administration
Endpoints Users N	etwork Devices Applicat	ion			
Static Assignment	false				
Endpoint Policy	CustomIOT				
Static Group Assignme	int false				
Identity Group Assignment	tent CustomIOT				
Custom Attributes					
				<b>▼</b> Fi	ter - 🗘 -
Attribute N	ame	Attribute Value	е		
× Attribute Na	me	Attribute Value			
assetGroup		Root			
Other Attributes					
BYODRegistration	Unknown				
DeviceRegistrationStat	us NotRegistered				
ElapsedDays	1				
	CustomIOT				
EndPointPolicy					
EndPointPolicy EndPointProfilerServer	ise24fc3.lab10.com				
EndPointPolicy EndPointProfilerServer EndPointSource	ise24fc3.lab10.com				
EndPointPolicy EndPointProfilerServer EndPointSource IdentityGroup	PXGRIDPROBE CustomIOT				
EndPointPolicy EndPointProfilerServer EndPointSource IdentityGroup InactiveDays	ise24fc3.lab10.com PXGRIDPROBE CustomIOT 1				

#### **Editing Script Values**

The API\_Simulator CustomPublisher.java code needs to be modified to modify the asset values is the script. In this example, Maven is used.

**Step 1** Copy the api\_simulator/pxgrid-rest-ws folder over to your maven repository.

/Users/jeppich/maven/repo/pxgrid-rest-ws

**Step 2** Go to the java folder, and see the pom.xml file and the src folder

```
johns-macbook-pro:pxgrid-rest-ws jeppich$ cd java
johns-macbook-pro:java jeppich$ ls
pom.xml src
run_publisher.sh target
```

#### **Step 3** Go to the http folder

/Users/jeppich/maven/repo/pxgrid-rest-ws/java/src/main/java/com/cisco/pxgrid/samples/ise/http

#### **Step 4** You will see the sample code

Console.java	SampleHelper.java
CustomPublisher.java	SessionQueryAll.java
CustomSubscriber.java	SessionQueryByIP.java
DeviceList.java	SessionSubscribe.java
Devices.java	StompFrame.java
PublisherController.java	StompPubsubClientEndpoint.java
PxgridControl.java	StompSubscription.java
SampleConfiguration.java	

Step 5 Edit the CustomPublisher.java file, to change any of the values, keep the prefix name (i.e.) object.put(prefix+"Vendor", and change the value name (i.e. "Cisco Systems"); in the section of the code below. The (prefix + ".") values cannot be changed as they represent the values if the IOTASSETS dictionary.

```
package com.cisco.pxgrid.samples.ise.http;
import java.net.URI;
import java.nio.charset.StandardCharsets;
import java.util.HashMap;
import java.util.Map;
import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.SSLSession;
import java.util.Random;
import org.glassfish.tyrus.client.ClientManager;
import org.glassfish.tyrus.client.ClientProperties;
import org.glassfish.tyrus.client.SlEngineConfigurator;
import org.glassfish.tyrus.client.auth.Credentials;
import org.json.simple.JSONArray;
import java.util.concurrent.ThreadLocalRandom;
```

```
cisco.
```

```
import com.cisco.pxgrid.model.AccountState;
import com.cisco.pxgrid.model.Service;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
/**
* Sample creation of a Dynamic Service by a client, also publishes. Another Client may subscribe to this
service and receive
* notifications when this service is published to.
* USE CASE for Dynamic Services: My company that requires all devices that connect to our network to be
specially authenticated so they need
 * more information. Using pxGrid dynamic services, I can set up a service that broadcasts the requirements
to all devices that connect and tell
* them to send this information over to allow for authentication.
 * @author anirvenk
*/
@SpringBootApplication
public class CustomPublisher {
       private static final String SERVICE NAME = "com.cisco.endpoint.asset";
       private static final String TOPIC_PATH = "/topic/com.cisco.endpoint.asset";
       private static final String PUBSUBSERVICE = "com.cisco.ise.pubsub";
       public static int counter = 0;
       private static String prefix = "asset";
       private static int counterIncrement = 1;
        /**
        * Static class that is used to create a sample Service object.
        */
       public static Service createService() {
               Service service = new Service();
               service.setName(SERVICE NAME);
               service.setNodeName("dynamic capability");
               Map<String, String> properties = new HashMap<String, String>();
               properties.put("wsPubsubService", PUBSUBSERVICE);
               properties.put("assetTopic", TOPIC_PATH);
properties.put("restBaseURL", "http://raghdasa-lnv1:8080");
               service.setProperties(properties);
               return service;
       public static String getRandomIpAddress(){
               Random r = new Random();
               return r.nextInt(256) + "." + r.nextInt(256) + "." + r.nextInt(256) + "." + r.nextInt(256);
    public static String getRandomMacAddress(){
       Random rand = new Random();
       byte[] macAddr = new byte[6];
       rand.nextBytes(macAddr);
       macAddr[0] = (byte)(macAddr[0] & (byte)254); //zeroing last 2 bytes to make it unicast and locally
adminstrated
       StringBuilder sb = new StringBuilder(18);
       for(byte b : macAddr){
               if(sb.length() > 0)
               sb.append(":");
                       sb.append(String.format("%02x", b));
               }
       return sb.toString();
    }
       public static JSONObject createJsonObject() {
               JSONObject object = new JSONObject();
               //change prefix to asset later
               object.put(prefix+"Id"
Integer.toString(1+counterIncrement)+Integer.toString(15*ThreadLocalRandom.current().nextInt(1, 6 + 1)));
```

```
object.put(prefix+"Name", "Abjergaryn -
4"+Integer.toString(ThreadLocalRandom.current().nextInt(1, 9 + 1)));
                 object.put(prefix+"IpAddress", getRandomIpAddress());
//+Integer.toString(ThreadLocalRandom.current().nextInt(45, 100 + 1)));
                  //object.put(prefix+"MacAddress", "28:63:36:a2:94:"+Integer.toString(30+counterIncrement));
                 object.put(prefix+"MacAddress", getRandomMacAddress());
object.put(prefix+"Vendor", "Cisco Systems");
                  object.put(prefix+"ProductId", "IE2000");
                 object.put(prefix+ Floatelia, 12000, ),
object.put(prefix+"SerialNumber", "1212121213243");
object.put(prefix+"DeviceType", "EtherNet/IP Node");
object.put(prefix+"SwRevision", "4.6");
object.put(prefix+"HwRevision", "5.6");
                 object.put(prefix+"HwRevision", "5.6"
object.put(prefix+"Protocol", "CIP");
                   JSONArray customAttr = new JSONArray();
                   JSONArray connectedLinks = new JSONArray();
    JSONObject object1 = new JSONObject();
                 object1.put("key","indattr2");
object1.put("value","3");
                  customAttr.add(object1);
                  JSONObject object2 = new JSONObject();
object2.put("key","assetGroup");
object2.put("value","Root");
                   customAttr.add(object2);
                   JSONObject object3 = new JSONObject();
          object3.put("key","indattr3");
object3.put("value","1");
          customAttr.add(object3);
          connectedLinks.add(object1);
                   connectedLinks.add(object2);
                   connectedLinks.add(object3);
                  object.put("assetCustomAttributes",customAttr);
                 object.put("assetConnectedLinks", connectedLinks);
                  counterIncrement++;
                 return object;
        }
        public static void main(String[] args) throws Exception {
                  // setting up the environment from passed in arguments
         SpringApplication.run(CustomPublisher.class, args);
                 SampleConfiguration config = new SampleConfiguration();
                  //creates PxGridControl object with the environment
                 PxgridControl control = new PxgridControl(config);
                  // AccountActivate
                 while (control.accountActivate() != AccountState.ENABLED) {
                           Thread.sleep(45000);
                 Console.log("pxGrid controller version=" + control.getControllerVersion());
                  //creating new service.
                 Service service = createService();
                  //registers the service that we created above
                 control.registerService(service.getName(), service.getProperties());
                 Service[] list_of_services;
                  //below lookup should find main pxGrid server.
                 list_of_services = control.lookupService(PUBSUBSERVICE);
                 if (list_of_services.length == 0) {
                           Console.log("service isn't there");
                           return:
                 }
                  // takes the main pxGrid server node
                  Service wsPubsubService = list of services[0];
                  //get wsURL so we can get the URI later from it via REST query.
                  String wsURL = wsPubsubService.getProperties().get("wsUrl");
                 Console.log("wsUrl=" + wsURL);
```



```
// pxGrid AccessSecret
                String secret = control.getAccessSecret(wsPubsubService.getNodeName());
                //setting up client manager which will use ssl connection for authentication.
                ClientManager client = ClientManager.createClient();
                SslEngineConfigurator sslEngineConfigurator = new
SslEngineConfigurator(config.getSSLContext());
                sslEngineConfigurator.setHostnameVerifier(new HostnameVerifier() {
                        @Override
                        public boolean verify(String hostname, SSLSession session) {
                                return true:
                        }
                });
                client.getProperties().put(ClientProperties.SSL ENGINE CONFIGURATOR, sslEngineConfigurator);
                client.getProperties().put(ClientProperties.CREDENTIALS,
                                new Credentials(config.getUserName(), secret.getBytes()));
                // WebSocket connect
                StompPubsubClientEndpoint endpoint = new StompPubsubClientEndpoint();
                //get URI, connect pxGrid client to the pxGrid server so that we can publish to dynamic
service
                URI uri = new URI(wsURL);
                javax.websocket.Session session = client.connectToServer(endpoint, uri);
                // STOMP connect
                endpoint.connect(uri.getHost());
                 * publishing to the dynamic service. This message "dynamic topic publish" will be received by
all subscribers to the service.
                 * only triggers when key is pressed. This is to make it so we can subscribe another client to
the service by running
                 * DynamicServiceSubscribe.java to see if it receives the published info.
                 */
                SampleHelper.prompt("press <enter> to start the publishing...");
                //for multi publishing
                //for(int i = 0; i < 20000; i++) {</pre>
                        JSONArray deviceArr = new JSONArray();
                        JSONObject device object = new JSONObject();
                        JSONObject device = createJsonObject();
                        deviceArr.add(device);
                        device_object.put("asset", device);
device_object.put("opType","UPDATE");
                        byte[] array = device_object.toJSONString().getBytes(StandardCharsets.UTF_8);
                        endpoint.publish(TOPIC_PATH, array);
                        Console.log(device.toJSONString());
            11
                        Console.log(Integer.toString(i));
               //}
                /*JSONArray deviceArr = new JSONArray();
                JSONObject device object = new JSONObject();
                JSONObject device1 = createJsonObject();
                JSONObject device2 = createJsonObject();
                JSONObject device3 = createJsonObject();
                JSONObject device4 = createJsonObject();
               device2.put(prefix+"SwRevision", "7.8");
device3.put(prefix+"SwRevision", "3.3");
device3.put(prefix+"HwRevision", "2.5");
device4.put(prefix+"SwRevision", "3.5");
                deviceArr.add(device1);
                deviceArr.add(device2);
                deviceArr.add(device3);
                deviceArr.add(device4);
                device_object.put("asset", device1);
                device_object.put("opType","UPDATE");
```



```
byte[] array = device_object.toJSONString().getBytes(StandardCharsets.UTF_8);
endpoint.publish(TOPIC_PATH, array);
*/
//SampleHelper.prompt("press <enter> to disconnect...");
// STOMP disconnect
//endpoint.disconnect("ID-123");
// Wait for disconnect receipt
//Thread.sleep(3000);
//session.close();
}
```

#### **Step 6** Change the Vendor Name to "ACME" and Save the file

🛑 🕒 💭 sers/jeppich/maven/i	repo/pxgrid-res	t-ws/java/src/main/java/com/cisco/pxgrid/samples/ise/http/CustomPublisher.java (Getting Started) — Brackets
	83	object.put(prefix+"1d",
Working Files 🛛 🔅 🖽	3	Integer.toString(1+counterIncrement)+Integer.toString(15*ThreadLocalRandom.current().n
		extInt(1, 6 + 1)));
run_publisher38.sh	84	object.put(prefix+"Name", "Abjergaryn -
README md		4"+Integer.toString(ThreadLocalRandom.current().nextInt(1, 9 + 1)));
READINE.IIIG	85	object.put(prefix+ <mark>"IpAddress</mark> ", getRandomIpAddress());
pom.xml	86	//+Integer.toString(ThreadLocalRandom.current().nextInt(45, 100 + 1)));
PublisherController.iava	87	//object.put(prefix+"MacAddress",
		"28:63:36:a2:94:"+Integer.toString(30+counterIncrement));
<ul> <li>CustomPublisher.java</li> </ul>	88	object.put(prefix+" <mark>MacAddress</mark> ", getRandomMacAddress());
	89	object.put(prefix+"Vendor", "ACME");
	90	object.put(prefix+"ProductId", "IE2000");
Getting Started 👻	91	object.put(prefix+"SerialNumber", "1212121213243");
0.000	92	<pre>object.put(prefix+"DeviceType", "EtherNet/IP Node");</pre>
index html	93	<pre>object.put(prefix+"SwRevision", "4.6");</pre>
macxintin	94	<pre>object.put(prefix+"HwRevision", "5.6");</pre>
main.css	95	object.put(prefix+"Protocol", "CIP");
▶ screenshots	96	JSONArray customAttr = new JSONArray();
P Screenshots	97	JSONArray connectedLinks = new JSONArray();
	98	
	99	JSONObject object1 = new JSONObject();
	100	object1.put("key","'ndattr2");
	Line 89, Colum	145 – 239 Lines INS UTF-8 ▼ Java ▼ ○ Spaces: 4

#### **Step 7** Go to folder

}

johns-macbook-pro:java	jeppich\$ ls
pom.xml	src
run publisher.sh	target
johns-macbook-pro:java	jeppich\$
5 1 5	

#### **Step 8** Run mvn install to recompile

```
johns-macbook-pro:java jeppich$ mvn install
[INFO] Scanning for projects...
[INFO]
[INFO] ------< com.cisco.pxgrid:pxgrid-rest-ws-samples >------
[INFO] Building Publisher 2.0.0-SNAPSHOT
```



-----[ jar ]------[INFO] -[INFO] [INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ pxgrid-rest-ws-samples ---[INFO] Using 'UTF-8' encoding to copy filtered resources. [INFO] Copying 40 resources [INFO] [INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ pxgrid-rest-ws-samples ---[INFO] Changes detected - recompiling the module! [INFO] Compiling 40 source files to /Users/jeppich/maven/repo/pxgrid-rest-ws/java/target/classes [WARNING] /Users/jeppich/maven/repo/pxgrid-restws/java/src/main/java/com/cisco/pxgrid/samples/ise/http/CustomPublisher.java: /Users/jeppich/maven/repo/pxgrid-restws/java/src/main/java/com/cisco/pxgrid/samples/ise/http/CustomPublisher.java uses unchecked or unsafe operations. [WARNING] /Users/jeppich/maven/repo/pxgrid-restws/java/src/main/java/com/cisco/pxgrid/samples/ise/http/CustomPublisher.java: Recompile with -Xlint:unchecked for details. [INFO] [INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ pxgrid-rest-ws-samples ---[INFO] Using 'UTF-8' encoding to copy filtered resources. [INFO] skip non existing resourceDirectory /Users/jeppich/maven/repo/pxgrid-rest-ws/java/src/test/resources [INFO] [INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ pxgrid-rest-ws-samples ---[INFO] No sources to compile [INFO] [INFO] --- maven-surefire-plugin:2.18.1:test (default-test) @ pxgrid-rest-ws-samples ---[INFO] No tests to run. [INFO] [INFO] --- maven-jar-plugin:2.6:jar (default-jar) @ pxgrid-rest-ws-samples ---[INFO] Building jar: /Users/jeppich/maven/repo/pxgrid-rest-ws/java/target/pxgrid-rest-ws-samples-2.0.0-SNAPSHOT.jar [INFO] [INFO] --- spring-boot-maven-plugin:1.5.6.RELEASE:repackage (default) @ pxgrid-rest-ws-samples ---[INFO] [INFO] --- maven-install-plugin:2.5.2:install (default-install) @ pxgrid-rest-ws-samples ---[INFO] Installing /Users/jeppich/maven/repo/pxgrid-rest-ws/java/target/pxgrid-rest-ws-samples-2.0.0-SNAPSHOT.jar to /Users/jeppich/.m2/repository/com/cisco/pxgrid/pxgrid-rest-ws-samples/2.0.0-SNAPSHOT/pxgridrest-ws-samples-2.0.0-SNAPSHOT.jar [INFO] Installing /Users/jeppich/maven/repo/pxgrid-rest-ws/java/pom.xml to /Users/jeppich/.m2/repository/com/cisco/pxgrid/pxgrid-rest-ws-samples/2.0.0-SNAPSHOT/pxgrid-rest-ws-samples-2.0.0-SNAPSHOT.pom [INFO] -----[INFO] BUILD SUCCESS [INFO] -----[INFO] Total time: 02:20 min [INFO] Finished at: 2018-04-21T16:13:14-04:00 [INFO] ----johns-macbook-pro:java jeppich\$

# Step 9Replace the /Applications/api\_partner\_fc3/api\_simulator/snapshot.jar file with the one that was just<br/>created. (i.e. /Users/jeppich/.m2/repository/com/cisco/pxgrid/pxgrid-rest-ws-samples/2.0.0-<br/>SNAPSHOT)

cp pxgrid-rest-ws-samples-2.0.0-SNAPSHOT.jar /Applications/api\_partner\_fc3/api\_simulator

**Step 10** Run the script, you should see "ACME" in the Asset Vendor

```
ClientEndpoint : STOMP SEND topic=/topic/com.cisco.endpoint.asset
command=SEND, headers={'content-length':'581','destination':'/topic/com.cisco.endpoint.asset',},
content.length=581
```



# 21-Apr-18 16:35:37.330 [main-1]: {"assetHwRevision":"5.6","assetProtocol":"CIP","assetConnectedLinks":[{"value":"3","key":"indattr2"},{"value" :"Root","key":"assetGroup"},{"value":"1","key":"indattr3"}],"assetVendor":"ACME","assetSwRevision":"4.6","ass etCustomAttributes":[{"value":"3","key":"indattr2"},{"value":"Root","key":"assetGroup"},{"value":"1","key":"indattr2"},{"value":"Root","key":"assetGroup"},{"value":"1","key":"indattr2"},{"value":"Root","key":"assetGroup"},{"value":"1","key":"indattr2"},{"value":"Root","key":"assetGroup"},{"value":"1","key":"indattr2"},{"value":"Root","key":"assetGroup"},{"value":"1","key":"indattr2"},{"value":"Root","key":"assetGroup"},{"value":"1","key":"indattr3"}],"assetProductId":"IE2000","assetSerialNumber":"1212121212121243","assetMacAddress":"00:66:cd:b6:17:d6 ","assetId":"275","assetIpAddress":"115.86.71.7","assetName":"Abjergaryn 41","assetDeviceType":"EtherNet\/IP Node"}

#### Step 11 You can now create a Profiling Policy based on IOTASSET assetvendor attribute using

* Name	VendorACME	Description		
Policy Enabled	0		h	
* Minimum Certainty Factor	10	(Valid Range 1 to 65535	)	
* Exception Action	NONE -			
* Network Scan (NMAP) Action	NONE -			
Create an Identity Group for the policy	<ul> <li>Yes, create matching Identity Gr</li> </ul>	oup		
	O No, use existing Identity Group I	nierarchy		
* Parent Policy	NONE -			
* Associated CoA Type	Global Settings 👻			
System Type	Administrator Created			
Rules				

#### Step 12 You can also create a logical profile VendorACME

Logical Profiles List > VendorACME		
Logical Profile		
* Name VendorACME	Description	
* Policy Assignment		
Available Policies:		Assigned Policies:
2Wire-Device		VendorACME
3Com-Device		
Aastra-Device	<	
Aastra-IP-Phone		
Aerohive-Access-Point	>>	
Aerohive-Device		
American-Power-Conversion-Device		
Android		

### Step 13 Select Context Visibility->Endpoints->Authentications, you should see "VendorACME" for the EndpointProfile

<u>Note</u>: If you see, "CustomIOTDevices", disable "CustomIOTDevices" profiling policy for this exercise.



altalta   cisco	dentity Services Engine	Home - Context	Visibility • Operations	▶ Policy	Administration			1 License	ə Warning 🔺 🔍 🥘	<b>0</b> 0
Endpoi	nts Users Network Dev	vices Application						Click here to do wireless setup an	d visibility setup Do not show	w this again.
	Authentication BYC	D Compliance	Compromised Endpoint	s Endpoin	t Classification	Guest	Vulnerable Endpoints	Haroware		- Q
" <	NACTIVE ENDPOIN	ITS 0		HENTICATI	No data availab	e.	ල AUTHI Failure R	ENTICATIONS  Control of the second se	0	Location
	Las	at Activity Date								
0 Sele	cted							Rows/Page 1 1	© /1 ► ► Go	1 Total Rows
S	+ 🛛 🖬 ANC -	<ul> <li>Change Authorization</li> </ul>	<ul> <li>Clear Threats &amp; Vulne</li> </ul>	rabilities Expe	ort - Import -	MDM Actions -	Release Rejected R	evoke Certificate	▼ Filter	r <b>- ₽</b> -
	MAC Address	Status	IPv4 Address	Username	Hostname	Location	Endpoint Profile	Authentication Failure Reason	Authentication Policy	Authoriza
×	MAC Address	Status 👻	IPv4 Address	Username	Hostname	Location	Endpoint Profile	Authentication Failure Reason	Authentication Policy	Authorizat
	F2:72:EE:16:D8:49		201.154.160.46				VendorACME			

#### **Editing/Adding Customer Values**

**Step 1** Copy the api\_simulator/ pxgrid-rest-ws folder over to your maven repository

```
/Users/jeppich/maven/repo/pxgrid-rest-ws
```

**Step 2** Go to the java folder, and see the pom.xml file and the src folder

```
johns-macbook-pro:pxgrid-rest-ws jeppich$ cd java
johns-macbook-pro:java jeppich$ ls
pom.xml src
run_publisher.sh target
```

#### **Step 3** Go to the http folder

/Users/jeppich/maven/repo/pxgrid-rest-ws/java/src/main/java/com/cisco/pxgrid/samples/ise/http

**Step 4** You will see the sample code

Console.java	SampleHelper.java
CustomPublisher.java	SessionQueryAll.java
CustomSubscriber.java	SessionQueryByIP.java



DeviceList.java Devices.java PublisherController.java PxgridControl.java SampleConfiguration.java SessionSubscribe.java StompFrame.java StompPubsubClientEndpoint.java StompSubscription.java

- Step 5 Edit the CustomPublisher.java file, replace object1.put ("key", "inadttr2") with object1.put ("key", "CVSS")"
- Step 6 Also replace object1.put("value","3") with object1.put ("value","7") in the highlighted fields below.

These fields will represent the custom attribute values of "key" and "value" that will be added to the ISE identity custom attribute screen and also when creating the CUSTOMATTRIBUTE profiling policy.

```
package com.cisco.pxgrid.samples.ise.http;
import java.net.URI;
import java.nio.charset.StandardCharsets;
import java.util.HashMap;
import java.util.Map;
import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.SSLSession;
import java.util.Random;
import org.glassfish.tyrus.client.ClientManager;
import org.glassfish.tyrus.client.ClientProperties;
import org.glassfish.tyrus.client.SslEngineConfigurator;
import org.glassfish.tyrus.client.auth.Credentials;
import org.json.simple.JSONArray;
import org.json.simple.JSONObject;
import java.util.concurrent.ThreadLocalRandom;
import com.cisco.pxgrid.model.AccountState;
import com.cisco.pxgrid.model.Service;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
/**
* Sample creation of a Dynamic Service by a client, also publishes. Another Client may subscribe to this
service and receive
* notifications when this service is published to.
* USE CASE for Dynamic Services: My company that requires all devices that connect to our network to be
specially authenticated so they need
* more information. Using pxGrid dynamic services, I can set up a service that broadcasts the requirements
to all devices that connect and tell
* them to send this information over to allow for authentication.
* @author anirvenk
*/
@SpringBootApplication
public class CustomPublisher {
       private static final String SERVICE_NAME = "com.cisco.endpoint.asset";
private static final String TOPIC_PATH = "/topic/com.cisco.endpoint.asset";
       private static final String PUBSUBSERVICE = "com.cisco.ise.pubsub";
       public static int counter = 0;
       private static String prefix = "asset";
       private static int counterIncrement = 1;
        /**
        * Static class that is used to create a sample Service object.
        */
       public static Service createService() {
               Service service = new Service();
               service.setName(SERVICE_NAME);
               service.setNodeName("dynamic capability");
               Map<String, String> properties = new HashMap<String, String>();
               properties.put("wsPubsubService", PUBSUBSERVICE);
               properties.put("assetTopic", TOPIC_PATH);
```



```
properties.put("restBaseURL", "http://raghdasa-lnv1:8080");
                 service.setProperties(properties);
                return service;
        }
        public static String getRandomIpAddress(){
                Random r = new Random();
return r.nextInt(256) + "." + r.nextInt(256) + "." + r.nextInt(256) + "." + r.nextInt(256);
        }
    public static String getRandomMacAddress(){
        Random rand = new Random();
        byte[] macAddr = new byte[6];
        rand.nextBytes(macAddr);
        macAddr[0] = (byte)(macAddr[0] & (byte)254); //zeroing last 2 bytes to make it unicast and locally
adminstrated
        StringBuilder sb = new StringBuilder(18);
        for(byte b : macAddr){
                if(sb.length() > 0)
                sb.append(":");
                         sb.append(String.format("%02x", b));
                }
        return sb.toString();
    }
        public static JSONObject createJsonObject() {
                JSONObject object = new JSONObject();
                 //change prefix to asset later
                object.put(prefix+"Id",
Integer.toString(1+counterIncrement)+Integer.toString(15*ThreadLocalRandom.current().nextInt(1, 6 + 1)));
                object.put(prefix+"Name", "Abjergaryn .
4"+Integer.toString(ThreadLocalRandom.current().nextInt(1, 9 + 1)));
                object.put(prefix+"IpAddress", getRandomIpAddress());
                 //+Integer.toString(ThreadLocalRandom.current().nextInt(45, 100 + 1)));
                 //object.put(prefix+"MacAddress", "28:63:36:a2:94:"+Integer.toString(30+counterIncrement));
                 object.put(prefix+"MacAddress", getRandomMacAddress());
                 object.put(prefix+"Vendor", "Cisco Systems");
                object.put(prefix+"ProductId", "IE2000");
                object.put(prefix+"ProductId", "IE2000");
object.put(prefix+"SerialNumber", "1212121213243");
object.put(prefix+"DeviceType", "EtherNet/IP Node");
object.put(prefix+"SwRevision", "4.6");
object.put(prefix+"HwRevision", "5.6");
object.put(prefix+"Protocol", "CIP");
                  JSONArray customAttr = new JSONArray();
                  JSONArray connectedLinks = new JSONArray();
    JSONObject object1 = new JSONObject();
                                                  // original (object1.put("key", "indattr2");
                 object1.put("key","CVSS");
                 object1.put("value","7");
                                                       // original (object1.put("value", "3");
                 customAttr.add(object1);
                  JSONObject object2 = new JSONObject();
                  object2.put("key","assetGroup");
object2.put("value","Root");
                  customAttr.add(object2);
                  JSONObject object3 = new JSONObject();
         object3.put("key", "indattr3");
object3.put("value", "1");
          customAttr.add(object3);
          connectedLinks.add(object1);
                  connectedLinks.add(object2);
                  connectedLinks.add(object3);
                 object.put("assetCustomAttributes",customAttr);
                object.put("assetConnectedLinks", connectedLinks);
                counterIncrement++;
                return object;
        }
        public static void main(String[] args) throws Exception {
                 // setting up the environment from passed in arguments
         SpringApplication.run(CustomPublisher.class, args);
```



```
SampleConfiguration config = new SampleConfiguration();
               //creates PxGridControl object with the environment
               PxgridControl control = new PxgridControl(config);
               // AccountActivate
               while (control.accountActivate() != AccountState.ENABLED) {
                       Thread.sleep(45000);
               Console.log("pxGrid controller version=" + control.getControllerVersion());
               //creating new service.
               Service service = createService();
               //registers the service that we created above
               control.registerService(service.getName(), service.getProperties());
               Service[] list_of_services;
               //below lookup should find main pxGrid server.
               list_of_services = control.lookupService(PUBSUBSERVICE);
               if (\overline{\text{list}} \text{ of services.length} == 0) {
                       Console.log("service isn't there");
                       return;
               }
               // takes the main pxGrid server node
               Service wsPubsubService = list of services[0];
               //get wsURL so we can get the URI later from it via REST query.
               String wsURL = wsPubsubService.getProperties().get("wsUrl");
               Console.log("wsUrl=" + wsURL);
               // pxGrid AccessSecret
               String secret = control.getAccessSecret(wsPubsubService.getNodeName());
               //setting up client manager which will use ssl connection for authentication.
               ClientManager client = ClientManager.createClient();
               SslEngineConfigurator sslEngineConfigurator = new
SslEngineConfigurator(config.getSSLContext());
               sslEngineConfigurator.setHostnameVerifier(new HostnameVerifier() {
                       @Override
                       public boolean verify(String hostname, SSLSession session) {
                              return true;
                       }
               });
               client.getProperties().put(ClientProperties.SSL_ENGINE_CONFIGURATOR, sslEngineConfigurator);
               client.getProperties().put(ClientProperties.CREDENTIALS,
                              new Credentials(config.getUserName(), secret.getBytes()));
               // WebSocket connect
               StompPubsubClientEndpoint endpoint = new StompPubsubClientEndpoint();
               //get URI, connect pxGrid client to the pxGrid server so that we can publish to dynamic
service
               URI uri = new URI(wsURL);
               javax.websocket.Session session = client.connectToServer(endpoint, uri);
               // STOMP connect
               endpoint.connect(uri.getHost());
               /*
                * publishing to the dynamic service. This message "dynamic topic publish" will be received by
all subscribers to the service.
                * only triggers when key is pressed. This is to make it so we can subscribe another client to
the service by running
                * DynamicServiceSubscribe.java to see if it receives the published info.
                */
               SampleHelper.prompt("press <enter> to start the publishing...");
               //for multi publishing
```



```
//for(int i = 0; i < 20000; i++) {</pre>
                  JSONArray deviceArr = new JSONArray();
                  JSONObject device_object = new JSONObject();
                  JSONObject device = createJsonObject();
                  deviceArr.add(device);
                  device_object.put("asset", device);
device_object.put("opType","UPDATE");
                  byte[] array = device_object.toJSONString().getBytes(StandardCharsets.UTF_8);
                  endpoint.publish(TOPIC_PATH, array);
                  Console.log(device.toJSONString());
    //
                  Console.log(Integer.toString(i));
         //}
         /*JSONArray deviceArr = new JSONArray();
         JSONObject device object = new JSONObject();
         JSONObject device1 = createJsonObject();
         JSONObject device2 = createJsonObject();
         JSONObject device3 = createJsonObject();
         JSONObject device4 = createJsonObject();
        device2.put(prefix+"SwRevision", "7.8");
device3.put(prefix+"SwRevision", "3.3");
device3.put(prefix+"HwRevision", "2.5");
device4.put(prefix+"SwRevision", "3.5");
         deviceArr.add(device1);
         deviceArr.add(device2);
         deviceArr.add(device3);
         deviceArr.add(device4);
        device_object.put("asset", device);
device_object.put("opType","UPDATE");
         byte[] array = device_object.toJSONString().getBytes(StandardCharsets.UTF_8);
         endpoint.publish(TOPIC_PATH, array);
         */
         //SampleHelper.prompt("press <enter> to disconnect...");
         // STOMP disconnect
         //endpoint.disconnect("ID-123");
         // Wait for disconnect receipt
         //Thread.sleep(3000);
         //session.close();
}
```

#### **Step 7** Go to folder

```
johns-macbook-pro:java jeppich$ ls
pom.xml src
run_publisher.sh target
johns-macbook-pro:java jeppich$
```

#### **Step 8** Run mvn install to recompile

cisco.

[INFO] [INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ pxgrid-rest-ws-samples ---[INFO] Changes detected - recompiling the module! [INFO] Compiling 40 source files to /Users/jeppich/maven/repo/pxgrid-rest-ws/java/target/classes [WARNING] /Users/jeppich/maven/repo/pxgrid-restws/java/src/main/java/com/cisco/pxgrid/samples/ise/http/CustomPublisher.java: /Users/jeppich/maven/repo/pxgrid-restws/java/src/main/java/com/cisco/pxgrid/samples/ise/http/CustomPublisher.java uses unchecked or unsafe operations. [WARNING] /Users/jeppich/maven/repo/pxgrid-restws/java/src/main/java/com/cisco/pxgrid/samples/ise/http/CustomPublisher.java: Recompile with -Xlint:unchecked for details. [INFO] -- maven-resources-plugin:2.6:testResources (default-testResources) @ pxgrid-rest-ws-samples ---[INFO] · [INFO] Using 'UTF-8' encoding to copy filtered resources. [INFO] skip non existing resourceDirectory /Users/jeppich/maven/repo/pxgrid-rest-ws/java/src/test/resources [INFO] [INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ pxgrid-rest-ws-samples ---[INFO] No sources to compile [INFO] [INFO] --- maven-surefire-plugin:2.18.1:test (default-test) @ pxgrid-rest-ws-samples ---[INFO] No tests to run. [INFO] [INFO] --- maven-jar-plugin:2.6:jar (default-jar) @ pxgrid-rest-ws-samples ---[INFO] Building jar: /Users/jeppich/maven/repo/pxgrid-rest-ws/java/target/pxgrid-rest-ws-samples-2.0.0-SNAPSHOT.jar [INFO] [INFO] --- spring-boot-maven-plugin:1.5.6.RELEASE:repackage (default) @ pxgrid-rest-ws-samples ---[INFO] [INFO] --- maven-install-plugin:2.5.2:install (default-install) @ pxgrid-rest-ws-samples ---[INFO] Installing /Users/jeppich/maven/repo/pxgrid-rest-ws/java/target/pxgrid-rest-ws-samples-2.0.0-SNAPSHOT.jar to /Users/jeppich/.m2/repository/com/cisco/pxgrid/pxgrid-rest-ws-samples/2.0.0-SNAPSHOT/pxgridrest-ws-samples-2.0.0-SNAPSHOT.jar [INFO] Installing /Users/jeppich/maven/repo/pxgrid-rest-ws/java/pom.xml to /Users/jeppich/.m2/repository/com/cisco/pxgrid/pxgrid-rest-ws-samples/2.0.0-SNAPSHOT/pxgrid-rest-ws-samples-2.0.0-SNAPSHOT.pom [INFO] -----[INFO] BUILD SUCCESS [INFO] -----[INFO] Total time: 19.717 s [INFO] Finished at: 2018-04-21T21:47:59-04:00 [INFO] ----johns-macbook-pro:java jeppich\$

Step 9 Replace the /Applications/api\_partner\_fc3/api\_simulator/snapshot.jar file with the one that was just created. (i.e /Users/jeppich/.m2/repository/com/cisco/pxgrid/pxgrid-rest-ws-samples/2.0.0-SNAPSHOT)

cp pxgrid-rest-ws-samples-2.0.0-SNAPSHOT.jar /Applications/api\_partner\_fc3/api\_simulator

Step 10 Run the script, you should see "value":"7", "key":"CVSS" in the output

```
ClientEndpoint : STOMP SEND topic=/topic/com.cisco.endpoint.asset
command=SEND, headers={'content-length':'576','destination':'/topic/com.cisco.endpoint.asset',},
content.length=576
21-Apr-18 21:56:49.171 [main-1]:
{"assetHwRevision":"5.6","assetProtocol":"CIP","assetConnectedLinks":[{"value":"7","key":"CVSS"},{"value":"Ro
ot","key":"assetGroup"},{"value":"1","key":"indattr3"}],"assetVendor":"ACME","assetSwRevision":"4.6","assetCu
stomAttributes":[{"value":"7","key":"CVSS"},{"value":"Root","key":"assetGroup"},{"value":"1","key":"indattr3"
}],"assetProductId":"IE2000","assetSerialNumber":"1212121213243","assetMacAddress":"b4:54:0a:e8:c1:fd","asset
```



Id":"275","assetIpAddress":"117.65.215.140","assetName":"Abjergaryn - 43","assetDeviceType":"EtherNet\/IP
Node"}

Step 11 Select Administration->System->Settings->Endpoint Custom Attributes, and add CVSS as the attribute name and int as the Type

dentity Services Engine	Home   Context	Visibility	Policy Adminis	tration Vork Centers			
System - Identity Management	Network Resources	Device Portal Management	pxGrid Services	Feed Service + Threat Co			
Identities Groups External Identit	y Sources Identity S	ource Sequences - Settings					
User Custom Attributes User Authentication Settings	Endpoint Cust	tom Attributes					
Endpoint Purge	Endpoint Attr	Attribute Name		Data Tura			
Endpoint Custom Attributes	Mandatory	Attribute Name		Data Type			
		PostureApplicable		STRING			
		EndPointPolicy		STRING			
		AnomalousBehaviour		STRING			
		OperatingSystem		STRING			
		BYODRegistration		STRING			
		PortalUser		STRING			
		LastAUPAcceptanceHours		INT			
		LogicalProfile		STRING			
	Endpoint Cust	tom Attributes					
		Attribute name	т	уре			
	assetGroup		String	<b>–</b>			
	CVSS		Int	- <b>O</b>			
			F	save Save			

Step 12 Select Save

**Step 13** Create CVSSAttribute Profiling Policy that will be used to define the CUSTOMATTRIBUTE

Profiler Policy List > CVSSAttribute	
Profiler Policy	
* Name	CVSSAttribute Description
Policy Enabled	
* Minimum Certainty Factor	10 (Valid Range 1 to 65535)
* Exception Action	NONE
* Network Scan (NMAP) Action	NONE
Create an Identity Group for the policy	Yes, create matching Identity Group
	○ No, use existing Identity Group hierarchy
* Parent Policy	NONE -
* Associated CoA Type	Global Settings 👻
System Type	Administrator Created
Rules	
If Condition CUSTOMATTRIBUTE_	CVSS_CONTAINS_7 💠 Then Certainty Factor Increases 🔹 100

**Step 14** Create CVSSAttribute Logical Profile is also created.

Note: Disable VendorACME policy if enabled

Logical Profiles List > CVSSAttributeProfile		
Logical Profile		
* Name CVSSAttributeProfile	Description	
* Policy Assignment		
Available Policies:		Assigned Policies:
2Wire-Device 3Com-Device	>	CVSSAttribute
Aastra-Device Aastra-IP-Phone	<	
Aerohive-Access-Point	>>	
Aerohive-Device American-Power-Conversion-Device	<<	
Android	]	

Step 15 Run the script, you should see "value":"7", "key":"CVSS" in the output

ClientEndpoint : STOMP SEND topic=/topic/com.cisco.endpoint.asset
<pre>command=SEND, headers={'content-length':'575','destination':'/topic/com.cisco.endpoint.asset',},</pre>
content.length=575
21-Apr-18 22:25:59.684 [main-1]:
{"assetHwRevision":"5.6","assetProtocol":"CIP","assetConnectedLinks":[{"value":"7","key":"CVSS"},{"value":"Ro
ot","key":"assetGroup"},{"value":"1","key":"indattr3"}],"assetVendor":"ACME","assetSwRevision":"4.6","assetCu
<pre>stomAttributes":[{"value":"7","key":"CVSS"},{"value":"Root","key":"assetGroup"},{"value":"1","key":"indattr3"</pre>
}],"assetProductId":"IE2000","assetSerialNumber":"1212121213243","assetMacAddress":"66:fe:67:4f:2c:4b","asset
Id":"230","assetIpAddress":"92.56.144.244","assetName":"Abjergaryn - 44","assetDeviceType":"EtherNet\/IP
Node" }

Step 16 Select Context Visibility->Endpoints->Authentication, you will see CVSSAttribute Endpoint Profile

dialo le	dentity Servic	es Engine	Home	- Context	Visibility	Operations	▶ Policy	Administration	Work Center	8		1	License Warning 🔺	ৎ (	• • •
Endpoi	nts Users	Network Dev	vices Applic	ation											
	Authentication	BYC	D Co	mpliance	Compromis	sed Endpoints	Endpo	int Classification	Guest	Vulnerable Er	indpoints Hardware				¢ -
I	NACTIVE E	INDPOIN	ITS <sup>0</sup>		0 B	AUTH	HENTICA	TION STATUS	©	0	AUTHENTICATIO	Store Identity Gro	bup	ល	NETWO
<													Jie.		>
		Las	st Activity Date	)											
0 Sele	cted										Rows/Pag	2 .	< <u>1</u> )/1 •	Go	2 Total Rows
c	+ ©	â ANC	- Change	Authorization	<ul> <li>Clear Th</li> </ul>	reats & Vulnera	abilities Ex	port - Import -	MDM Actions -	Release R	Rejected Revoke Certifica	te		<b>▼</b> Filte	er - 🗘 -
	MAC Addre	SS	Status		IPv4 Addre	ess	Username	Hostname	Location	Endpoint I	Profile Authentica	ation Failure Rea	ason Authentic	ation Policy	Authoriza
×	MAC Addres	is	Status	•	IPv4 Addres	ss	Username	Hostname	Location	Endpoint P	Profile Authentica	tion Failure Reaso	n Authentic	ation Policy	Authorizat
	66:FE:67:4	F:2C:4B			92.56.144.24	44				CVSSAttrib	oute				

#### **API\_Simulator Context-In Code**

#### SampleConfiguration

This code configures the command-line arguments to run the inside the **/run\_publisher** script.

```
package com.cisco.pxgrid.samples.ise.http;
import java.io.FileInputStream;
import java.io.IOException;
import java.net.Authenticator;
import java.net.PasswordAuthentication;
import java.net.Socket;
import java.security.GeneralSecurityException;
import java.security.KeyStore;
import java.security.Principal;
import java.security.PrivateKey;
import java.security.cert.Certificate;
import java.security.cert.CertificateException;
import java.security.cert.CertificateFactory;
import java.security.cert.X509Certificate;
import java.util.Collection;
import java.util.Enumeration;
import javax.net.ssl.HttpsURLConnection;
import javax.net.ssl.KeyManager;
import javax.net.ssl.KeyManagerFactory;
import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManager;
import javax.net.ssl.TrustManagerFactory;
import javax.net.ssl.X509KeyManager;
import javax.net.ssl.X509TrustManager;
public class SampleConfiguration {
    protected final static String PROP HOSTNAMES="PXGRID HOSTNAMES";
    protected final static String PROP_USERNAME="PXGRID_USERNAME";
    protected final static String PROP_PASSWORD="PXGRID_PASSWORD";
    protected final static String PROP_GROUP="PXGRID_GROUP";
    protected final static String PROP_DESCRIPTION="PXGRID_DESCRIPTION";
    protected final static String PROP_KEYSTORE_FILENAME="PXGRID_KEYSTORE_FILENAME";
    protected final static String PROP KEYSTORE PASSWORD="PXGRID KEYSTORE PASSWORD";
    protected final static String PROP_TRUSTSTORE_FILENAME="PXGRID_TRUSTSTORE_FILENAME";
    protected final static String PROP_TRUSTSTORE_PASSWORD="PXGRID_TRUSTSTORE_PASSWORD";
    private String[] hostnames;
    private String username;
    private String password;
    private String[] groups;
    private String description;
    private SSLContext sslContext;
    private String keystoreFilename;
    private String keystorePassword;
    private String truststoreFilename;
    private String truststorePassword;
       public SampleConfiguration() throws GeneralSecurityException, IOException {
               load();
               print();
       }
    public String getUserName() {
               return username;
       }
    public void setUsername(String username) {
               this.username = username;
       }
```



```
public String[] getGroups() {
                return groups;
        }
        public String getDescription() {
               return description;
        ł
    public SSLContext getSSLContext() {
        return sslContext;
    public String getPassword() {
                return password;
        3
    public String[] getHostnames() {
               return hostnames;
        }
    private void load() throws GeneralSecurityException, IOException {
        String hostnameProperty = System.getProperty(PROP_HOSTNAMES);
        username = System.getProperty(PROP USERNAME);
        password = System.getProperty(PROP_PASSWORD);
        String group_property = System.getProperty(PROP_GROUP);
        description = System.getProperty(PROP DESCRIPTION);
        keystoreFilename = System.getProperty(PROP_KEYSTORE_FILENAME);
        keystorePassword = System.getProperty(PROP_KEYSTORE_PASSWORD);
        truststoreFilename = System.getProperty(PROP_TRUSTSTORE_FILENAME);
        truststorePassword = System.getProperty(PROP_TRUSTSTORE_PASSWORD);
if (hostnameProperty == null || hostnameProperty.isEmpty()) throw new
IllegalArgumentException("Missing " + PROP_HOSTNAMES);
        if (username == null || username.isEmpty()) throw new IllegalArgumentException("Missing " +
PROP_USERNAME);
        if (truststoreFilename == null || truststoreFilename.isEmpty()) throw new
IllegalArgumentException("Missing " + PROP TRUSTSTORE FILENAME);
        if (truststorePassword == null || truststorePassword.isEmpty()) throw new
IllegalArgumentException("Missing " + PROP_TRUSTSTORE_PASSWORD);
        hostnames = hostnameProperty.split(",");
        if (group_property != null && !group_property.isEmpty()) {
                        groups = group_property.split(",");
        }
        if (description != null) {
                 if (description.isEmpty()) description = null;
                 else description = description.trim();
        }
        sslContext = SSLContext.getInstance("TLSv1.2");
        sslContext.init(getKeyManagers(), getTrustManagers(), null);
    }
    public void setupAuth(HttpsURLConnection https) throws GeneralSecurityException, IOException {
                Authenticator.setDefault(new MyAuthenticator());
    }
    private class MyAuthenticator extends Authenticator {
        public PasswordAuthentication getPasswordAuthentication() {
            return (new PasswordAuthentication(username, password.toCharArray()));
        }
    }
       private KeyManager[] getKeyManagers() throws IOException, GeneralSecurityException {
    if (keystoreFilename == null || keystoreFilename.isEmpty())
                        return null;
                KeyStore ks = keystoreFilename.endsWith(".pl2") ? KeyStore.getInstance("pkcs12") :
KeyStore.getInstance("JKS");
                FileInputStream in = new FileInputStream(keystoreFilename);
```

# cisco.

```
ks.load(in, keystorePassword.toCharArray());
               in.close();
               KeyManagerFactory kmf =
KeyManagerFactory.getInstance(KeyManagerFactory.getDefaultAlgorithm());
               kmf.init(ks, keystorePassword.toCharArray());
               KeyManager[] mngrs = kmf.getKeyManagers();
               if (mngrs == null || mngrs.length == 0) {
                      throw new GeneralSecurityException("no key managers found");
               }
               if (mngrs[0] instanceof X509KeyManager == false) {
                      throw new GeneralSecurityException("key manager is not for X509");
               }
               return new KeyManager[] { new SampleX509KeyManager((X509KeyManager) mngrs[0]) };
       }
       private TrustManager[] getTrustManagers() throws IOException, GeneralSecurityException {
               FileInputStream in = new FileInputStream(truststoreFilename);
               KeyStore ks = null;
               if(truststoreFilename.endsWith(".pem")) {
                      ks = KeyStore.getInstance("JKS");
                      ks.load(null, null);
                      CertificateFactory certFac = CertificateFactory.getInstance("X.509");
                      Collection<? extends Certificate> certs = certFac.generateCertificates(in);
                      int i = 0;
                      for(Certificate c : certs) {
                              ks.setCertificateEntry("trust-" + i, c);
               } else if(truststoreFilename.endsWith(".p12")) {
                      ks = KeyStore.getInstance("pkcs12");
                      ks.load(in, truststorePassword.toCharArray());
               } else {
                      ks = KeyStore.getInstance("JKS");
                      ks.load(in, truststorePassword.toCharArray());
               }
               in.close();
               Enumeration<String> e = ks.aliases();
               boolean hasCertEntries = false;
               while (e.hasMoreElements()) {
                      String alias = e.nextElement();
                      if (ks.isCertificateEntry(alias)) {
                              hasCertEntries = true;
                      }
               }
               if (hasCertEntries == false) {
                      e = ks.aliases();
                      while (e.hasMoreElements()) {
                              String alias = e.nextElement();
                              if (ks.isKeyEntry(alias)) {
                                      Certificate[] chain = ks.getCertificateChain(alias);
                                      for (int i = 0; i < chain.length; ++i) {
                                             ks.setCertificateEntry(alias + "." + i, chain[i]);
                                      }
                              }
                      }
               }
               TrustManagerFactory tmf =
TrustManagerFactory.getInstance(TrustManagerFactory.getDefaultAlgorithm());
               tmf.init(ks);
               TrustManager[] tms = tmf.getTrustManagers();
```



```
if (tms == null || tms.length == 0) {
                      throw new GeneralSecurityException("no trust managers found");
               }
               if (tms[0] instanceof X509TrustManager == false) {
                      throw new GeneralSecurityException("trust manager is not for X509");
               }
               return new TrustManager[] { new SampleX509TrustManager((X509TrustManager) tms[0]) };
       }
       private static class SampleX509KeyManager implements X509KeyManager {
               private X509KeyManager mngr;
               public SampleX509KeyManager(X509KeyManager mngr) {
                      this.mngr = mngr;
               }
               @Override
               public String chooseClientAlias(String[] arg0, Principal[] arg1, Socket arg2) {
                      String alias = mngr.chooseClientAlias(arg0, arg1, arg2);
                      if (alias == null) {
                              alias = mngr.chooseClientAlias(arg0, null, arg2);
                              if (alias == null) {
                                      throw new RuntimeException("no client certificate found ...");
                              }
                      }
                      return alias;
               }
               @Override
               public String chooseServerAlias(String arg0, Principal[] arg1, Socket arg2) {
                      throw new RuntimeException("Not implemented");
               }
               @Override
               public X509Certificate[] getCertificateChain(String arg0) {
                      return mngr.getCertificateChain(arg0);
               }
               @Override
               public String[] getClientAliases(String arg0, Principal[] arg1) {
                      return mngr.getClientAliases(arg0, null);
               }
               @Override
               public PrivateKey getPrivateKey(String arg0) {
                      return mngr.getPrivateKey(arg0);
               }
               @Override
               public String[] getServerAliases(String arg0, Principal[] arg1) {
                      throw new RuntimeException("Not implemented");
               }
       }
       private static class SampleX509TrustManager implements X509TrustManager {
               private X509TrustManager mngr;
               public SampleX509TrustManager(X509TrustManager mngr) {
                      this.mngr = mngr;
               }
               @Override
               public void checkClientTrusted(X509Certificate[] arg0, String arg1) throws
CertificateException {
                      throw new RuntimeException("not implemented");
               }
```



@Override public void checkServerTrusted(X509Certificate[] arg0, String arg1) throws CertificateException { try { mngr.checkServerTrusted(arg0, arg1); } catch (CertificateException e) { throw new CertificateException("Server certificate is not trusted:" + arg0[0].getSubjectX500Principal(), e); } } @Override public X509Certificate[] getAcceptedIssuers() { return mngr.getAcceptedIssuers(); } } private void print() { System.out.println("----- properties ------"); System.out.print(" hostnames="); for (String hostname : hostnames) System.out.print(hostname + " "); System.out.println(); System.out.println(' username=" + username); System.out.println(" password=" + password); System.out.print(" groups="); for (String group : groups) System.out.print(group + " "); System.out.println(); System.out.println(" System.out.println(" description=" + description); keystoreFilename=" + keystoreFilename); keystorePassword=" + keystorePassword); System.out.println(" System.out.println(" truststoreFilename=" + truststoreFilename); System.out.println(" truststorePassword=" + truststorePassword); System.out.println("------"); }

#### PxgridControl

This code provides the pxGrid client with account creation on the ISE pxGrid node and service lookup request and access secret to the peer node.

```
package com.cisco.pxgrid.samples.ise.http;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.URL;
import java.util.Base64;
import java.util.Map;
import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.HttpsURLConnection;
import javax.net.ssl.SSLSession;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import com.cisco.pxgrid.model.AccessSecretRequest;
import com.cisco.pxgrid.model.AccessSecretResponse;
import com.cisco.pxgrid.model.AccountActivateRequest;
import com.cisco.pxgrid.model.AccountActivateResponse;
import com.cisco.pxgrid.model.AccountCreateRequest;
import com.cisco.pxgrid.model.AccountCreateResponse;
import com.cisco.pxgrid.model.AccountState;
import com.cisco.pxgrid.model.Authorization;
import com.cisco.pxgrid.model.AuthorizationRequest;
import com.cisco.pxgrid.model.AuthorizationResponse;
```

# cisco.

```
import com.cisco.pxgrid.model.Service;
import com.cisco.pxgrid.model.ServiceLookupRequest;
import com.cisco.pxgrid.model.ServiceLookupResponse;
import com.cisco.pxgrid.model.ServiceRegisterRequest;
import com.cisco.pxgrid.model.ServiceRegisterResponse;
import com.google.gson.Gson;
/**
 * Using HTTPS for pxGrid control
*/
public class PxgridControl {
        private static Logger logger = LoggerFactory.getLogger(PxgridControl.class);
        private SampleConfiguration config;
    private String controllerVersion;
    public PxgridControl(SampleConfiguration config) {
        this.config = config;
        }
        private <T> T sendRequest(HttpsURLConnection https, Object request, Class<T> responseClass) throws
IOException {
                https.setRequestProperty("Content-Type", "application/json");
                https.setRequestProperty("Accept", "application/json");
                Gson gson = new Gson();
                OutputStreamWriter out = new OutputStreamWriter(https.getOutputStream());
logger.info("Request={}", gson.toJson(request));
                gson.toJson(request, out);
                out.flush();
        InputStreamReader in = new InputStreamReader(https.getInputStream());
        T response = gson.fromJson(in, responseClass);
                logger.info("Response={}", gson.toJson(response));
        return response;
        }
    private HttpsURLConnection getHttpsURLConnection(String urlSuffix) throws IOException {
    String url = "https://" + config.getHostnames()[0] + ":8910/pxgrid/control/" + urlSuffix;
                URL conn = new URL(url);
                HttpsURLConnection https = (HttpsURLConnection) conn.openConnection();
                // SSL and Auth
                https.setSSLSocketFactory(config.getSSLContext().getSocketFactory());
                https.setRequestMethod("POST");
                String userPassword = config.getUserName() + ":" + config.getPassword();
                String encoded = Base64.getEncoder().encodeToString(userPassword.getBytes());
                https.setRequestProperty("Authorization", "Basic " + encoded);
                https.setHostnameVerifier(new HostnameVerifier() {
                        @Override
                        public boolean verify(String hostname, SSLSession session) {
                                return true;
                        }
                });
                https.setDoInput(true);
                https.setDoOutput(true);
                return https;
    }
    /**
     * Create new account
     * @return password
     */
    public String accountCreate() throws IOException {
        HttpsURLConnection https = getHttpsURLConnection("AccountCreate");
                AccountCreateRequest request = new AccountCreateRequest();
                request.setNodeName(config.getUserName());
```



AccountCreateResponse response = sendRequest(https, request, AccountCreateResponse.class); return response.getPassword(); } public AccountState accountActivate() throws IOException { HttpsURLConnection https = getHttpsURLConnection("AccountActivate"); AccountActivateRequest request = new AccountActivateRequest(); request.setDescription(config.getDescription()); AccountActivateResponse response = sendRequest(https, request, AccountActivateResponse.class); controllerVersion = response.getVersion(); return response.getAccountState(); } public void registerService(String name, Map<String, String> properties) throws IOException { HttpsURLConnection https = getHttpsURLConnection("ServiceRegister"); ServiceRegisterRequest request = new ServiceRegisterRequest(); request.setName(name); request.setProperties(properties); sendRequest(https, request, ServiceRegisterResponse.class); } public Service[] lookupService(String name) throws IOException { HttpsURLConnection https = getHttpsURLConnection("ServiceLookup"); ServiceLookupRequest request = new ServiceLookupRequest(); request.setName(name); ServiceLookupResponse response = sendRequest(https, request, ServiceLookupResponse.class); return response.getServices(); } public String getAccessSecret(String peerNodeName) throws IOException { HttpsURLConnection https = getHttpsURLConnection("AccessSecret"); AccessSecretRequest request = new AccessSecretRequest(); request.setPeerNodeName(peerNodeName); AccessSecretResponse response = sendRequest(https, request, AccessSecretResponse.class); return response.getSecret(); } public boolean isAuthorized(String requestNodeName, String serviceName, String operation) throws IOException { HttpsURLConnection https = getHttpsURLConnection("Authorization"); AuthorizationRequest request = new AuthorizationRequest(); request.setRequestNodeName(requestNodeName); request.setServiceName(serviceName); request.setServiceOperation(operation); AuthorizationResponse response = sendRequest(https, request, AuthorizationResponse.class); return (response.getAuthorization() == Authorization.PERMIT); } public String getControllerVersion() { return controllerVersion; }

#### PublisherController

}

This code publishes the "assets" topic and looks like the following:
cisco.

Pretty	Raw Preview JSON V
1 <del>-</del> 2 -	assets": [
3 -	{
4	"assetName": "Morello",
5	"assetId": "1",
6	"assetProtocol": "Profinet",
7	"assetIpAddress": "100.100.100.70",
8	"assetMacAddress": "60:88:36:a2:94:44",
9	"assetVendor": "Cisco",
10	"assetProductId": "IE20001",
11	"assetDeviceType": "IPNODE",
12	"assetSwRevision": "9.4",
13	"assetHwRevision": "2.3",
14	"assetSerialNumber": "3492748"
15	},
16 -	{
17	"assetName": "Morello",
18	"assetId": "2",
19	"assetProtocol": "Profinet",
20	"assetIpAddress": "100.100.100.30",
21	"assetMacAddress": "10:55:36:a2:94:99",
22	"assetVendor": "Cisco",
23	"assetProductId": "IE20001",
24	"assetDeviceType": "IPNODE",
25	"assetSwRevision": "9.4",
26	"assetHwkevision": "2.3",
27	"assetSerialNumber": "3492748"
28	J,

```
package com.cisco.pxgrid.samples.ise.http;
import java.util.concurrent.ThreadLocalRandom;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
@RestController
public class PublisherController {
       public Devices DeviceGenerator(int i) {
               return new Devices(Integer.toString(i+1), "Morello",
                               "100.100.100."+10*ThreadLocalRandom.current().nextInt(1, 9 + 1),
(10*ThreadLocalRandom.current().nextInt(1, 9 + 1))+":"+(11*ThreadLocalRandom.current().nextInt(1, 9 +
1))+":36:a2:94:"+
(11*ThreadLocalRandom.current().nextInt(1, 9 + 1)), "Cisco",
"IE20001", "3492748", "IPNODE", "9.4", "2.3", "Profinet");
       }
    @RequestMapping("/getAssets")
    public DeviceList device() {
               Devices[] device_list = new Devices[5];
               for(int i = 0; i < 5; i++) {
                       device_list[i] = DeviceGenerator(i);
               }
        return new DeviceList(device_list);
    }
```

#### **Custom Publisher**

The CustomPublisher Java code publishes the asset device attributes by calling the service name "com.cisco.endpoint.asset", topic path "/topic/com.cisco.endpoint.asset", PUBSERVCE "com.cisco.ise.pubsub.

This will return the wsPubsubServce: "com.cisco.ise.pubsub", restbaseURL <u>http://raghdasa-lnv1:8080</u>, and assetTopic, "/topic/com.cisco.endpoint.asset".

```
package com.cisco.pxgrid.samples.ise.http;
import java.net.URI;
import java.nio.charset.StandardCharsets;
import java.util.HashMap;
import java.util.Map;
import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.SSLSession;
import java.util.Random;
import org.glassfish.tyrus.client.ClientManager;
import org.glassfish.tyrus.client.ClientProperties;
import org.glassfish.tyrus.client.SslEngineConfigurator;
import org.glassfish.tyrus.client.auth.Credentials;
import org.json.simple.JSONArray;
import org.json.simple.JSONObject;
import java.util.concurrent.ThreadLocalRandom;
import com.cisco.pxgrid.model.AccountState;
import com.cisco.pxgrid.model.Service;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
/**
* Sample creation of a Dynamic Service by a client, also publishes. Another Client may subscribe to this
service and receive
* notifications when this service is published to.
* USE CASE for Dynamic Services: My company that requires all devices that connect to our network to be
specially authenticated so they need
 more information. Using pxGrid dynamic services, I can set up a service that broadcasts the requirements
to all devices that connect and tell
* them to send this information over to allow for authentication.
* @author anirvenk
*/
@SpringBootApplication
public class CustomPublisher {
       private static final String SERVICE NAME = "com.cisco.endpoint.asset";
       private static final String TOPIC PATH = "/topic/com.cisco.endpoint.asset";
       private static final String PUBSUBSERVICE = "com.cisco.ise.pubsub";
       public static int counter = 0;
       private static String prefix = "asset";
       private static int counterIncrement = 1;
        * Static class that is used to create a sample Service object.
        */
       public static Service createService() {
               Service service = new Service();
               service.setName(SERVICE_NAME);
               service.setNodeName("dynamic capability");
               Map<String, String> properties = new HashMap<String, String>();
               properties.put("wsPubsubService", PUBSUBSERVICE);
               properties.put("assetTopic", TOPIC_PATH);
properties.put("restBaseURL", "http://raghdasa-lnv1:8080");
               service.setProperties(properties);
               return service;
       }
       public static String getRandomIpAddress(){
               Random r = new Random();
return r.nextInt(256) + "." + r.nextInt(256) + "." + r.nextInt(256) + "." + r.nextInt(256);
   public static String getRandomMacAddress(){
       Random rand = new Random();
       byte[] macAddr = new byte[6];
       rand.nextBytes(macAddr);
```

```
cisco.
```

```
macAddr[0] = (byte)(macAddr[0] & (byte)254); //zeroing last 2 bytes to make it unicast and locally
adminstrated
        StringBuilder sb = new StringBuilder(18);
        for(byte b : macAddr){
                if(sb.length() > 0)
                sb.append(":");
                         sb.append(String.format("%02x", b));
                }
        return sb.toString();
    }
        public static JSONObject createJsonObject() {
                JSONObject object = new JSONObject();
                //change prefix to asset later
                object.put(prefix+"Id",
4"+Integer.toString(ThreadLocalRandom.current().nextInt(1, 9 + 1)));
                object.put(prefix+"IpAddress", getRandomIpAddress());
                 //+Integer.toString(ThreadLocalRandom.current().nextInt(45, 100 + 1)));
                 //object.put(prefix+"MacAddress", "28:63:36:a2:94:"+Integer.toString(30+counterIncrement));
                object.put(prefix+"MacAddress", getRandomMacAddress());
                object.put(prefix+"Vendor", "ACME");
                object.put(prefix+"ProductId", "IE2000");
                object.put(prefix+"Productid, "122000","
object.put(prefix+"SerialNumber", "1212121213243");
object.put(prefix+"DeviceType", "EtherNet/IP Node");
object.put(prefix+"SwRevision", "4.6");
object.put(prefix+"HwRevision", "5.6");
                object.put(prefix+"HwRevision", "5.6"
object.put(prefix+"Protocol", "CIP");
                 JSONArray customAttr = new JSONArray();
                 JSONArray connectedLinks = new JSONArray();
    JSONObject object1 = new JSONObject();
                object1.put("key","CVSS");
object1.put("value","7");
                customAttr.add(object1);
                 JSONObject object2 = new JSONObject();
object2.put("key","assetGroup");
object2.put("value","Root");
                  customAttr.add(object2);
                 JSONObject object3 = new JSONObject();
         object3.put("key","indattr3");
object3.put("value","1");
          customAttr.add(object3);
          connectedLinks.add(object1);
                  connectedLinks.add(object2);
                  connectedLinks.add(object3);
                object.put("assetCustomAttributes",customAttr);
                object.put("assetConnectedLinks",connectedLinks);
                counterIncrement++;
                return object;
        }
        public static void main(String[] args) throws Exception {
                 // setting up the environment from passed in arguments
         SpringApplication.run(CustomPublisher.class, args);
                SampleConfiguration config = new SampleConfiguration();
                //creates PxGridControl object with the environment
                PxgridControl control = new PxgridControl(config);
                // AccountActivate
                while (control.accountActivate() != AccountState.ENABLED) {
                         Thread.sleep(45000);
                Console.log("pxGrid controller version=" + control.getControllerVersion());
```



```
//creating new service.
               Service service = createService();
               //registers the service that we created above
               control.registerService(service.getName(), service.getProperties());
               Service[] list_of_services;
               //below lookup should find main pxGrid server.
               list_of_services = control.lookupService(PUBSUBSERVICE);
               if (list_of_services.length == 0) {
                       Console.log("service isn't there");
                       return;
               }
               // takes the main pxGrid server node
               Service wsPubsubService = list_of_services[0];
               //get wsURL so we can get the URI later from it via REST query.
               String wsURL = wsPubsubService.getProperties().get("wsUrl");
               Console.log("wsUrl=" + wsURL);
               // pxGrid AccessSecret
               String secret = control.getAccessSecret(wsPubsubService.getNodeName());
               //setting up client manager which will use ssl connection for authentication.
               ClientManager client = ClientManager.createClient();
               SslEngineConfigurator sslEngineConfigurator = new
SslEngineConfigurator(config.getSSLContext());
               sslEngineConfigurator.setHostnameVerifier(new HostnameVerifier() {
                       @Override
                       public boolean verify(String hostname, SSLSession session) {
                              return true;
                       }
               });
               client.getProperties().put(ClientProperties.SSL_ENGINE_CONFIGURATOR, sslEngineConfigurator);
               client.getProperties().put(ClientProperties.CREDENTIALS,
                              new Credentials(config.getUserName(), secret.getBytes()));
               // WebSocket connect
               StompPubsubClientEndpoint endpoint = new StompPubsubClientEndpoint();
               //get URI, connect pxGrid client to the pxGrid server so that we can publish to dynamic
service
               URI uri = new URI(wsURL);
               javax.websocket.Session session = client.connectToServer(endpoint, uri);
               // STOMP connect
               endpoint.connect(uri.getHost());
               /*
                * publishing to the dynamic service. This message "dynamic topic publish" will be received by
all subscribers to the service.
                * only triggers when key is pressed. This is to make it so we can subscribe another client to
the service by running
                * DynamicServiceSubscribe.java to see if it receives the published info.
                */
               SampleHelper.prompt("press <enter> to start the publishing...");
               //for multi publishing
               //for(int i = 0; i < 20000; i++) {</pre>
                       JSONArray deviceArr = new JSONArray();
                       JSONObject device object = new JSONObject();
                       JSONObject device = createJsonObject();
                       deviceArr.add(device);
                       device_object.put("asset", device);
device_object.put("opType","UPDATE");
                       byte[] array = device object.toJSONString().getBytes(StandardCharsets.UTF 8);
                       endpoint.publish(TOPIC_PATH, array);
                       Console.log(device.toJSONString());
                       Console.log(Integer.toString(i));
            11
               //}
```

```
/*JSONArray deviceArr = new JSONArray();
         JSONObject device_object = new JSONObject();
         JSONObject device1 = createJsonObject();
         JSONObject device2 = createJsonObject();
         JSONObject device3 = createJsonObject();
         JSONObject device4 = createJsonObject();
        device2.put(prefix+"SwRevision", "7.8");
device3.put(prefix+"SwRevision", "3.3");
device3.put(prefix+"HwRevision", "2.5");
device4.put(prefix+"SwRevision", "3.5");
         deviceArr.add(device1);
         deviceArr.add(device2);
         deviceArr.add(device3);
         deviceArr.add(device4);
        device_object.put("asset", device1);
device_object.put("opType","UPDATE");
         byte[] array = device_object.toJSONString().getBytes(StandardCharsets.UTF_8);
         endpoint.publish(TOPIC PATH, array);
         */
         //SampleHelper.prompt("press <enter> to disconnect...");
         // STOMP disconnect
         //endpoint.disconnect("ID-123");
         // Wait for disconnect receipt
         //Thread.sleep(3000);
         //session.close();
}
```

### CustomSubscriber

}

This code sets up the subscriber to the published topic, in the case, the ISE pxGrid node.

```
package com.cisco.pxgrid.samples.ise.http;
import java.net.URI;
import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.SSLSession;
import org.glassfish.tyrus.client.ClientManager;
import org.glassfish.tyrus.client.ClientProperties;
import org.glassfish.tyrus.client.SslEngineConfigurator;
import org.glassfish.tyrus.client.auth.Credentials;
import com.cisco.pxgrid.model.AccountState;
import com.cisco.pxgrid.model.Service;
/**
* Demonstrates how to subscribe to a Dynamic Service using REST/WS.
* Works hand in hand with Dynamic Service.java class.
*/
public class CustomSubscriber {
       // Subscribe handler class
       private static class SessionHandler implements StompSubscription.Handler {
               @Override
               public void handle(StompFrame message) {
                      System.out.println(new String(message.getContent()));
               }
       }
```

# cisco.

```
public static void main(String [] args) throws Exception {
               // Read environment for config
               SampleConfiguration config = new SampleConfiguration();
               PxgridControl control = new PxgridControl(config);
               // AccountActivate
               while (control.accountActivate() != AccountState.ENABLED) {
                      Thread.sleep(60000);
               Console.log("pxGrid controller version=" + control.getControllerVersion());
               // dynamic service lookup
               Console.log("Looking up service com.custom.ise.dynamic");
               Service[] services = control.lookupService("com.custom.ise.dynamic");
               if (services.length == 0) {
                      Console.log("dynamic service unavailabe");
                      return:
               }
               //gets dynamic service.
               Service service = services[0];
               String wsPubsubServiceName = service.getProperties().get("wsPubsubService");
               //sets topic to be used later as the service to subscribe to.
               String topic = service.getProperties().get("iotTopic");
               Console.log("wsPubsubServiceName=" + wsPubsubServiceName + " iotTopic=" + topic);
               // Pubsub ServiceLookup
               services = control.lookupService(wsPubsubServiceName);
               if (services.length == 0) {
                      Console.log("Pubsub service unavailabe");
                      return;
               }
               // Select first one which ends up being the pxGrid server node. Should cycle through until
connects.
               Service wsPubsubService = services[0];
               String wsURL = wsPubsubService.getProperties().get("wsUrl");
               Console.log("wsUrl=" + wsURL);
               // pxGrid AccessSecret
               String secret = control.getAccessSecret(wsPubsubService.getNodeName());
               //setting up client manager which will use ssl connection for authentication.
               ClientManager client = ClientManager.createClient();
               SslEngineConfigurator sslEngineConfigurator = new
SslEngineConfigurator(config.getSSLContext());
               sslEngineConfigurator.setHostnameVerifier(new HostnameVerifier() {
                      @Override
                      public boolean verify(String hostname, SSLSession session) {
                              return true:
                      }
               });
               client.getProperties().put(ClientProperties.SSL_ENGINE_CONFIGURATOR, sslEngineConfigurator);
               client.getProperties().put(ClientProperties.CREDENTIALS,
                              new Credentials(config.getUserName(), secret.getBytes()));
               // WebSocket connect
               StompPubsubClientEndpoint endpoint = new StompPubsubClientEndpoint();
               URI uri = new URI(wsURL);
               //connects the pxGrid client to the pxGrid server
               javax.websocket.Session session = client.connectToServer(endpoint, uri);
               // STOMP connect
               endpoint.connect(uri.getHost());
               // Subscribes to the specific topic of the dynamic service
               StompSubscription subscription = new StompSubscription(topic, new SessionHandler());
               endpoint.subscribe(subscription);
```



```
SampleHelper.prompt("press <enter> to disconnect...");
// STOMP disconnect
endpoint.disconnect("ID-123");
// Wait for disconnect receipt
Thread.sleep(3000);
session.close();
}
```

## **Device List**

This code returns the list of devices as called in the PublishController Code

```
package com.cisco.pxgrid.samples.ise.http;
import com.fasterxml.jackson.annotation.JsonProperty;
public class DeviceList {
    Devices[] list_of_devices;
    public DeviceList(Devices[] device) {
        this.list_of_devices = device;
    }
    @JsonProperty("assets")
    public Devices[] getDeviceList() {
        return list_of_devices;
    }
}
```

## Devices

This code defines the asset and custom attributes

```
package com.cisco.pxgrid.samples.ise.http;
import com.fasterxml.jackson.annotation.JsonProperty;
import org.json.simple.JSONObject;
public class Devices {
       private String assetId;
       private String assetIpAddress;
       private String assetMacAddress;
       private String assetName;
       private String assetVendor;
       private String assetSerialNumber;
       private String assetProductId;
       private String assetDeviceType;
       private String assetSwRevision;
       private String assetHwRevision;
       private String assetProtocol;
// private JSONObject CustomAttributes;
       public Devices(String assetId, String assetName, String assetIpAddress, String assetMacAddress,
String assetVendor, String assetProductId,
                      String assetSerialNumber, String assetDeviceType, String assetSwRevision, String
assetHwRevision, String assetProtocol) {
               this.assetId = assetId;
               this.assetName = assetName;
```

## cisco.

```
this.assetIpAddress = assetIpAddress;
             this.assetMacAddress = assetMacAddress;
             this.assetVendor = assetVendor;
             this.assetProductId = assetProductId;
             this.assetSerialNumber = assetSerialNumber;
             this.assetDeviceType = assetDeviceType;
             this.assetSwRevision = assetSwRevision;
             this.assetHwRevision = assetHwRevision;
             this.assetProtocol = assetProtocol;
             this.CustomAttributes= new JSONObject();
     /*
             this.CustomAttributes.put("test1","value1");
this.CustomAttributes.put("test2","value2");*/
     }
/*@JsonProperty("CustomAttributes")
     public JSONObject getCustomAttributes() {
       return CustomAttributes;
     }*/
     @JsonProperty("assetId")
     public String getId() {
             return assetId;
     }
     @JsonProperty("assetIpAddress")
     public String getIpAddress() {
             return assetIpAddress;
     }
     @JsonProperty("assetMacAddress")
     public String getMacAddress() {
             return assetMacAddress;
     }
     @JsonProperty("assetName")
     public String getName() {
             return assetName;
     }
     @JsonProperty("assetVendor")
     public String getVendor() {
             return assetVendor;
     }
     @JsonProperty("assetSerialNumber")
     public String getSerialNumber() {
             return assetSerialNumber;
     }
     @JsonProperty("assetProductId")
     public String getProductId() {
             return assetProductId;
     }
     @JsonProperty("assetDeviceType")
     public String getDeviceType() {
             return assetDeviceType;
     }
     @JsonProperty("assetSwRevision")
     public String getSwRevision() {
             return assetSwRevision;
     }
     @JsonProperty("assetHwRevision")
     public String getHwRevision() {
             return assetHwRevision;
     }
     @JsonProperty("assetProtocol")
     public String getProtocol() {
             return assetProtocol;
     }
```

## Console.java

This code prints out the time zone

### Sample Helper. Java

This code helps with the initial WebSockets Connection

```
package com.cisco.pxgrid.samples.ise.http;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;
import java.nio.charset.StandardCharsets;
import java.text.ParseException;
import java.time.OffsetDateTime;
import java.time.format.DateTimeFormatter;
import java.util.Base64;
import java.util.Scanner;
import javax.net.ssl.HttpsURLConnection;
import javax.net.ssl.SSLSocketFactory;
import org.apache.commons.io.IOUtils;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
```

```
cisco.
```

```
import com.google.gson.TypeAdapter;
import com.google.gson.stream.JsonReader;
import com.google.gson.stream.JsonToken;
import com.google.gson.stream.JsonWriter;
public class SampleHelper {
       private static Logger logger = LoggerFactory.getLogger(SampleHelper.class);
       public static HttpsURLConnection createHttpsURLConnection(String url, String user, String password,
SSLSocketFactory sslSocketFactory) throws IOException {
               URL conn = new URL(url);
               HttpsURLConnection https = (HttpsURLConnection) conn.openConnection();
               https.setSSLSocketFactory(sslSocketFactory);
               String userPassword = user + ":" + password;
               String encoded = Base64.getEncoder().encodeToString(userPassword.getBytes());
               https.setRequestProperty("Authorization", "Basic " + encoded);
               return https:
       }
       public static String prompt(String msg) {
        System.out.print(msg);
        @SuppressWarnings("resource")
               Scanner scanner = new Scanner(System.in);
        String value = scanner.nextLine();
if ("".equals(value)) return null;
        return value;
   }
    public static OffsetDateTime promptDate(String msg) throws ParseException {
       String value = prompt(msg);
       if (value == null) return null;
       return OffsetDateTime.parse(value);
       }
       public static void postObjectAndPrint(String url, String user, String password, SSLSocketFactory ssl,
Object postObject) throws IOException {
               Gson gson = new GsonBuilder()
                    .registerTypeAdapter(OffsetDateTime.class, new OffsetDateTimeAdapter())
                    .create();
               postStringAndPrint(url, user, password, ssl, gson.toJson(postObject));
       }
       public static void postStringAndPrint(String url, String user, String password, SSLSocketFactory ssl,
String postData) throws IOException {
               logger.info("postData={}", postData);
               HttpsURLConnection httpsConn = SampleHelper.createHttpsURLConnection(url, user, password,
ssl);
       httpsConn.setRequestMethod("POST");
       httpsConn.setRequestProperty("Content-Type", "application/json");
       httpsConn.setRequestProperty("Accept", "application/json");
       httpsConn.setDoInput(true);
       httpsConn.setDoOutput(true);
               OutputStreamWriter osw = new OutputStreamWriter(httpsConn.getOutputStream());
               osw.write(postData);
               osw.flush();
               int status = httpsConn.getResponseCode();
       logger.info("Response status={}", status);
               if (status < HttpURLConnection.HTTP BAD REQUEST) {</pre>
                       try (InputStream in = httpsConn.getInputStream()) {
                              String content = IOUtils.toString(in, StandardCharsets.UTF_8);
                       System.out.println("Content: " + content);
               }
               }
               else {
                       try (InputStream in = httpsConn.getErrorStream()) {
                               String content = IOUtils.toString(in, StandardCharsets.UTF 8);
                       System.out.println("Content: " + content);
               }
               }
       }
```



```
private static class OffsetDateTimeAdapter extends TypeAdapter<OffsetDateTime> {
       DateTimeFormatter formatter = DateTimeFormatter.ISO OFFSET DATE TIME;
       @Override
       public void write(JsonWriter out, OffsetDateTime value) throws IOException {
               if (value == null) {
                      out.nullValue();
                      return;
               }
               out.value(formatter.format(value));
       }
       @Override
       public OffsetDateTime read(JsonReader in) throws IOException {
               if (in.peek() == JsonToken.NULL) {
                       in.nextNull();
                       return null;
               }
               return formatter.parse(in.nextString(), OffsetDateTime::from);
       }
}
```

#### Stompframe.java

This code handles the STOMP message frame

```
package com.cisco.pxgrid.samples.ise.http;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.text.ParseException;
import java.util.HashMap;
import java.util.Map;
/**
* This follows STOMP 1.2 specification to parse and generate STOMP frames:
*
    https://stomp.github.io/stomp-specification-1.2.html
* This single class is self-sufficient handle all STOMP frames.
* Note for WebSocket:
* If input comes as WebSocket text type, (WS RFC says Text is UTF-8)
 * server side handling code like Spring TextMessage may convert the bytes to String as UTF-8
* which maybe the wrong encoding as STOMP frame itself can use other encoding.
    e.g. A particular encoding may have bytes: FF FF... 10, that is completely out
of range for Unicode.
* Unless STOMP body is also UTF-8, STOMP frame must be sent as binary
* @author Alan Lei
*/
public class StompFrame {
       public enum Command {
               CONNECT, STOMP, CONNECTED, SEND, SUBSCRIBE, UNSUBSCRIBE, ACK, NACK,
               BEGIN, COMMIT, ABORT, DISCONNECT, MESSAGE, RECEIPT, ERROR;
               private static Map<String, Command> mapOfStringToCommand = new HashMap<>();
               static {
                      for (Command command : Command.values()) {
                              mapOfStringToCommand.put(command.name(), command);
                      }
               }
               public static Command get(String value) {
                      return mapOfStringToCommand.get(value);
               }
```



```
}
private Command command;
private Map<String, String> headers = new HashMap<>();
private byte[] content;
private final static int MAX_BUFFER_SIZE = 1024;
public Command getCommand() {
       return command;
}
public void setCommand(Command command) {
       this.command = command;
}
public String getHeader(String name) {
       return headers.get(name);
}
public void setHeader(String name, String value) {
       headers.put(name, value);
}
public Map<String, String> getHeaders() {
       return headers;
}
public byte[] getContent() {
       return content;
}
public void setContent(byte[] content) {
       this.content = content;
}
public void write(OutputStream out) throws IOException {
        out.write(command.name().getBytes());
       out.write('\n');
       for (String name : headers.keySet()) {
               out.write(name.getBytes());
               out.write(':');
               out.write(headers.get(name).getBytes());
               out.write('\n');
       }
       out.write('\n');
       if (content != null) {
               out.write(content);
       }
       out.write(0);
}
private static String readLine(InputStream in) throws IOException, ParseException {
       byte[] line = new byte[MAX_BUFFER_SIZE];
       int index = 0;
       while (index < MAX_BUFFER_SIZE) {</pre>
               int b = in.read();
               if (b != -1) {
                       if (b == '\n') {
                               return new String(line, 0, index);
                       }
                       }
if (b != '\r') {
    line[index] = (byte)b;
                               index++;
                       }
               }
               else {
                       // No line found
                       return null;
               }
       throw new ParseException("Line too long", MAX_BUFFER_SIZE);
}
```



```
* Using InputStream instead of Reader because
 * content-length is octet count instead of character count
 */
public static StompFrame parse(InputStream reader) throws IOException, ParseException {
       StompFrame stomp = new StompFrame();
        // Read Command
       String line = readLine(reader);
       Command command = Command.get(line);
        if (command == null) throw new ParseException("Unknown command: " + line, 0);
       stomp.setCommand(command);
        // Read Headers
        int contentLength = -1;
       while ((line = readLine(reader)) != null) {
               if (line.equals("")) break;
               int colon = line.indexOf(':');
               String name = line.substring(0, colon);
               String value = line.substring(colon + 1);
               stomp.setHeader(name, value);
               if (name.equals("content-length")) {
                       contentLength = Integer.parseInt(value);
               }
       }
        // Read Content
        if (contentLength != -1) {
               // content-length is in octets
               byte[] content = new byte[contentLength];
               reader.read(content);
               stomp.setContent(content);
               if (reader.read() != 0) {
                       throw new ParseException("Byte after STOMP Body not NULL", -1);
       }
        else {
                // No content-length. Look for ending NULL byte.
               byte[] buffer = new byte[MAX_BUFFER_SIZE];
               int length = 0;
               while (length < MAX_BUFFER_SIZE) {</pre>
                       int b = reader.read();
                       if (b == -1) {
                               throw new ParseException("Premature end of stream", -1);
                       if (b == 0) {
                               if (length > 0) {
                                       byte[] content = new byte[length];
                                       System.arraycopy(buffer, 0, content, 0, length);
                                       stomp.setContent(content);
                               // More EOLs may follow, but ignored.
                               return stomp;
                       buffer[length] = (byte)b;
                       length++;
               throw new ParseException("Frame too long", -1);
       }
       return stomp;
}
@Override
public String toString() {
       StringBuilder sb = new StringBuilder();
       sb.append("command=" + command);
       sb.append(", headers={");
       for (String name : headers.keySet()) {
    sb.append("'" + name + "':");
               sb.append("'" + headers.get(name) + "',");
        }
        sb.append("}");
       sb.append(", content.length=" + content.length);
```

return }

return sb.toString();

## StompSubscription

This code handles subscription to the STOMP Messaging protocol.

```
package com.cisco.pxgrid.samples.ise;
import java.util.concurrent.atomic.AtomicInteger;
public class StompSubscription {
       public static interface Handler {
               void handle(StompFrame message);
       }
       private static AtomicInteger currentSubscriptionID = new AtomicInteger();
       private String id = Integer.toString(currentSubscriptionID.getAndIncrement());
       private String topic;
       private Handler handler;
       public StompSubscription(String topic, Handler handler) {
               this.topic = topic;
               this.handler = handler;
       }
       public String getId() {
               return id;
       }
       public String getTopic() {
               return topic;
       }
       public Handler getHandler() {
               return handler;
       }
       public StompFrame getSubscribeMessage() {
               StompFrame message = new StompFrame();
               message.setCommand(StompFrame.Command.SUBSCRIBE);
               message.setHeader("destination", topic);
               message.setHeader("id", id);
               return message;
       }
```

### StompPubSubClientEndpoint

This code handles endpoint client subscription to the STOMP messaging protocol.

```
package com.cisco.pxgrid.samples.ise.http;
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.nio.ByteBuffer;
import java.text.ParseException;
import java.util.Map;
import java.util.concurrent.ConcurrentHashMap;
```



```
import javax.websocket.ClientEndpoint;
import javax.websocket.CloseReason;
import javax.websocket.Endpoint;
import javax.websocket.EndpointConfig;
import javax.websocket.MessageHandler;
import javax.websocket.Session;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import com.cisco.pxgrid.samples.ise.http.StompSubscription.Handler;
@ClientEndpoint
public class StompPubsubClientEndpoint extends Endpoint {
       private static Logger logger = LoggerFactory.getLogger(StompPubsubClientEndpoint.class);
       private volatile Session session;
       private Map<String, StompSubscription> mapOfIdToSubscription = new ConcurrentHashMap<>();
       public void connect(String hostname) throws IOException {
       logger.info("STOMP CONNECT host=" + hostname);
               StompFrame message = new StompFrame();
       message.setCommand(StompFrame.Command.CONNECT);
       message.setHeader("accept-version", "1.2");
message.setHeader("host", hostname);
       send(message);
       }
       public void disconnect(String receipt) throws IOException {
               logger.info("STOMP DISCONNECT receipt=" + receipt);
               StompFrame message = new StompFrame();
       message.setCommand(StompFrame.Command.DISCONNECT);
       if (receipt != null) {
               message.setHeader("receipt", receipt);
       }
       send(message);
        }
       public void subscribe(StompSubscription subscription) throws IOException {
               logger.info("STOMP SUBSCRIBE topic=" + subscription.getTopic());
               mapOfIdToSubscription.put(subscription.getId(), subscription);
               if (session != null) {
                       StompFrame message = subscription.getSubscribeMessage();
               send(message);
               }
       }
       public void publish(String topic, byte[] content) throws IOException {
               logger.info("STOMP SEND topic=" + topic);
               StompFrame message = new StompFrame();
       message.setCommand(StompFrame.Command.SEND);
       message.setHeader("destination", topic);
message.setHeader("content-length", Integer.toString(content.length));
       message.setContent(content);
       System.out.println(message);
       send(message);
       }
       private void send(StompFrame message) throws IOException {
               if (session != null) {
               ByteArrayOutputStream baos = new ByteArrayOutputStream();
               message.write(baos);
               // Send as binary
               session.getBasicRemote().sendBinary(ByteBuffer.wrap(baos.toByteArray()));
       }
       public void waitForOpen() throws InterruptedException {
               synchronized (this) {
                       while (session == null) {
                               this.wait();
                       }
               }
```



```
}
private void onStompMessage(StompFrame stomp) {
       switch (stomp.getCommand()) {
       case CONNECTED:
               String version = stomp.getHeader("version");
               logger.info("STOMP CONNECTED version={}", version);
               break;
       case RECEIPT:
               String receiptId = stomp.getHeader("receipt-id");
               logger.info("STOMP RECEIPT id={}", receiptId);
               break;
       case MESSAGE:
               String id = stomp.getHeader("subscription");
               StompSubscription subscription = mapOfIdToSubscription.get(id);
               Handler handler = subscription.getHandler();
               if (handler != null) {
                      handler.handle(stomp);
               }
               break;
       case ERROR:
               // Server will close connect on ERROR according to STOMP specification
               logger.info("STOMP ERROR stomp={}", stomp);
               break;
       default:
               // Ignore others
               break;
       }
}
private class TextHandler implements MessageHandler.Whole<String> {
       @Override
       public void onMessage(String message) {
       try {
               StompFrame stomp = StompFrame.parse(new ByteArrayInputStream(message.getBytes()));
               onStompMessage(stomp);
       } catch (IOException | ParseException e) {
               logger.error("onMessage", e);
       }
       }
}
private class BinaryHandler implements MessageHandler.Whole<InputStream> {
        @Override
       public void onMessage(InputStream in) {
       try {
                       StompFrame stomp = StompFrame.parse(in);
                       onStompMessage(stomp);
               } catch (IOException | ParseException e) {
               logger.error("onMessage", e);
               }
       }
}
@Override
public void onOpen(Session session, EndpointConfig cfg) {
       logger.info("WS onOpen");
       this.session = session;
       try {
       session.addMessageHandler(new TextHandler());
       session.addMessageHandler(new BinaryHandler());
       for (StompSubscription subscription : mapOfIdToSubscription.values()) {
               StompFrame message = subscription.getSubscribeMessage();
                              send(message);
               }
       } catch (IOException e) {
               logger.error("onOpen", e);
       }
       synchronized (this) {
               this.notifyAll();
       }
}
```



```
@Override
public void onClose(Session session, CloseReason closeReason) {
        logger.info("WS onClose closeReason code={} phrase={}", closeReason.getCloseCode(),
closeReason.getReasonPhrase());
        this.session = null;
    }
    @Override
    public void onError(Session session, Throwable thr) {
        logger.info("WS onError thr={}", thr.getMessage());
        this.session = null;
    }
}
```



## **Endpoint Asset Configuration**

#### Service: com.cisco.endpoint.asset

This is endpoint asset service topic

#### Service properties

Name	Description	Example
restBaseUrl	The base URL for REST APIs	https://[ind-host1]:8910/pxgrid/ind/asset
wsPubsubService	The WebSocket Pubsub service name	com.cisco.ise.pubsub
assetTopic	Topic for asset events	/topic/com.cisco.endpoint.asset

#### assetTopic WS Stomp message:



"opType" will be one of the following strings: CREATE/UPDATE/DELETE

Sample Response:





#### POST [restBaseUrl]/getAssets

This is used to get all Endpoint devices. Request URL: [restBaseUrl]/getAssets Request Method: POST Content-Type: application/json Accept: application/json

Authorization: Basic [nodeName]:[secret]

Label	Description
[restBaseUrl]	Obtain by ServiceLookup of com.cisco.endpoint.asset
[nodeName]	pxGrid node name
[secret]	Obtain via AccessSecret

#### Request:



Parameter	Description	Default Value
limit	Number of records per page	500
offset	Zero based page index	0

Response:

{
 "assets": [

array of Asset objects
]
}

#### "Asset" Object

Name	Туре	Description
assetId	Long	IND DB id for the asset
assetName	String	Asset name
assetIpAddress	String	IP address
assetMacAddress	String	MAC address
assetVendor	String	Manufacturer
assetProductId	String	Product Code
assetSerialNumber	String	Serial Number
assetDeviceType	String	Device Type
assetSwRevision	String	S/W Revision number
assetHwRevision	String	H/W Revision number
assetProtocol	String	
assetConnectedLinks		Array of Link object
assetCustomAttributes		Array of name-value pair in the form of Pair object

#### Asset Custom Attributes

Key Name	Value Type	Description	Supported IND Version
assetGroup	String	Fully qualified group name assigned to the Asset	1.3, 1.4
assetTag	String	Security Tag assigned to the Asset	1.4

#### "Link" Object

Name	Туре	Description
assetId	Long	IND DB id for the connected asset
assetName	String	Connected Asset name
assetIpAddress	String	Connected Asset IP address
assetPortName	String	Name of the interface that the asset is connected to
assetDeviceType	String	Device Type

#### "Pair" Object

Name	Туре	Description
key	String	Custom attribute name
value	String	Custom attribute value

Sample Response:



=	
"assets	s": [🖃
{ 🗆	
	"assetId": "118",
	"assetName": "172.27.162.154",
	"assetIpAddress": "172.27.162.154",
	"assetMacAddress": "ec:e5:55:7d:4a:2c",
	"assetVendor": "Hirschmann, a Belden brand",
	"assetProductId": "Hirschmann RSR30-L2",
	"assetSerialNumber": "8210988",
	"assetDeviceType": "Switch",
	"assetSwRevision": "",
	"assetHwRevision": "",
	"assetProtocol": "CIP",
	"assetConnectedLinks": [ ],
	"assetCustomAttributes": [ ]
· .	
10	
•	"assetId": "106".
	"assetName": "172.27.162.171"
	"assetIpAddress": "172.27.162.171"
	"assetMacAddress": "e4:90:69:a1:14:27"
	"assetVendor": "Rockwell Automation/Allen-Bradley"
	"assetProductId": "1734-AENTR/B Ethernet Adapter".
	"assetSerialNumber": "1615510289"
	"assetDeviceType": "EtherNet/IP Node"
	"assetSwRevision": ""
	"assetHwRevision": ""
	"assetProtocol": "CTP"
	"assetConnectedLinks": []
	18
	"assetId": "109",
	"assetName": "IE5k-162-165.cisco.com"
	"assetIpAddress": "172.27.162.165".
	"assetPortName": "GigabitEthernet1/9".
	"assetDeviceType": "Switch"
	}
	1.
	"assetCustomAttributes": [ ]
1	
1 1	

#### POST [restBaseUrl]/getSyncStatus

This is used to get the sync status of the IND Pxgrid Service.

Request URL: [restBaseUrl]/getSyncStatus

Request Method: POST

Content-Type: application/json

Accept: application/json

Authorization: Basic [nodeName]:[secret]

Label	Description
[restBaseUrl]	Obtain by ServiceLookup of com.cisco.endpoint.asset
[nodeName]	pxGrid node name
[secret]	Obtain via AccessSecret

Request:

{

}

Response:



Response will be one of the following strings: DISABLED/IN\_SYNC/OUT\_OF\_SYNC