**Pxgrid Notes**

Cisco pxGrid allows for bi-directional any-to-any partner platform integrations.

Cisco pxGrid uses a pub/sub model and publishes Cisco Identity Services Engine (ISE) contextual information.

In addition, pxGrid publishes this session directory topic and other ISE topics of information for ecosystem partners to consume.

Initially pxGrid was released with ISE 1.3 and was XMPP-based, requiring an SDK containing the Java and C libraries, and sample code. This is now named pxGrid 1.0 and supported in ISE versions 1.3 and higher.

\*\* Extensible Messaging and Presence Protocol (XMPP) is a communication protocol for message-oriented middleware based on XML. It enables the near-real-time exchange of structured yet extensible data between any two or more network entities.

\*\* A software development kit (SDK) is a collection of software development tools in one installable package. They ease creation of applications by having compiler, debugger and perhaps a software framework. They are normally specific to a hardware platform and operating systems combination.

pxGrid 2.0 officially released in ISE 2.4 uses WebSocket and REST API over STOMP (Simple Text Oriented Message Protocol) 1.2 messaging protocol. WebSockets and STOMP are widely used in the industry. WebSockets allow the client and server to keep an open connection, so there is bi-directional communication as opposed to using long polling to keep the connection as was used in hacks prior to using WebSockets. STOMP is used over WebSockets to communicate with any STOMP message broker.

This provides the developer with the following benefits:

-- NO SDK Dependency

-- ease of clientless approach ??

-- horizontal scalability of ISE pxGrid servers.

\*\* A message broker is an intermediary computer program module that translates a message from the formal messaging protocol of the sender to the formal messaging protocol of the receiver.

\*\* Vertical scalability is the ability to increase the capacity of existing hardware or software by adding resources - for example, adding processing power to a server to make it faster.

\*\* Horizontal scalability is the ability to connect multiple entities so that they work as a single logical unit.

Some use cases are:

Consuming Session Topic information from ISE

Cisco Identity Services Engine (ISE) contains session information that other systems can use in their security policy information. This session information provides user identity information and includes connection type, endpoint device and compliant attributes and more.

Session={

ip=[192.168.1.15],

Audit Session Id=0A000001000000170001b0ab,

UerName=pxgrid1,

ADUserDNSDomain=lab10.com,

ADUserNetBIOSName=LAB10,

ADUserResolvedIdentities=pxGrid1@lab10.com,

ADUserResolvedDNs=CN=pxgrid1, CN=User, DC=lab10, DC=com,

MacAddress=[00:51:56:86:C9:92],

State=STARTED,

ANCstatus=ANC\_Quarantine,

Security Group=Quarantined\_System,

EndpointProfile=VMWare-Device,

NAS IP=192.168.1.3,

NAS Port=GigabitEthernet1/0/11,

RADIUSAVPairs=[Acct-Session-Id=0000002E],

Posture Status=null,

Posture Timestamp=,

LastUpdateTime=Sat Jan 21 11:49:04 EST 2017,

Session attributeName=Authorization\_Profiles,

Session attributeValue=Quarantined\_Systems,

Providers=[None],

EndpointcheckResult=none,

IdentitySourceFirstPort=0,

IdentityPortStart=0

}

Dynamic Topics

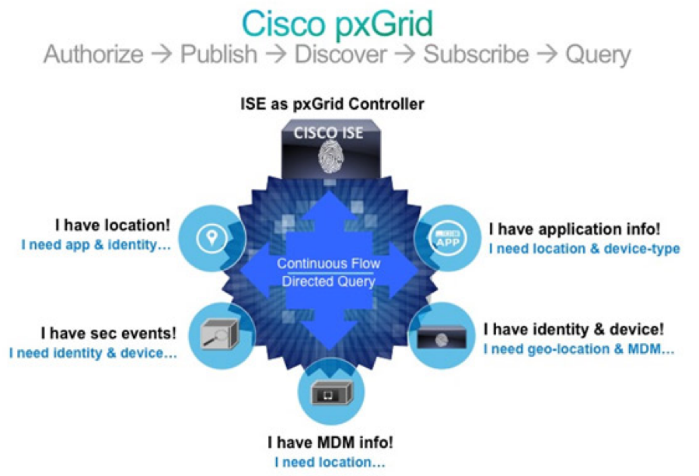
The Cisco Platform Exchange Grid (pxGrid) allows you to integrate your application into the pxGrid.

you can use pxGrid to exchange context with Cisco products, such as the Cisco Identity Service Engine (ISE), or any other Cisco partner that supports pxGrid. Cisco pxGrid will provide you with an API that will open up a unified framework that will enable you to integrate to pxGrid once, then share context with any other platform that supports pxGrid.

This hub and spoke architecture means that you integrate once to pxGrid and there is no need for you to adopt a host of individual, platform-specific

• Context Sharing Control - Because pxGrid is customizable, you can “publish” only the specific information (context) that you want to share, and you can control which other pxGrid partner platforms that it gets shared with.

• Bidirectional context sharing – pxGrid enables partner platforms such as yours and others to either publish context or to subscribe to context; you orchestrate and secure what is published and what is subscribed through the pxGrid controller which resides on Cisco Identity Service Engine (ISE). • Share context data in native formats – you share contextual information in pxGrid using the native data format of your platform - pxGrid does the rest. • Connect to multiple platforms simultaneously – pxGrid enables you to publish only the context data that is relevant to pxGrid partner subscribers. You can customize numerous context “topics” for a variety of partner platforms, yet always shared via the same reusable pxGrid framework. By sharing only relevant data both publishing and subscribing platforms are able to scale by eliminating irrelevant data.



**From Sandbox**

pxGrid 2.o Overview

**Cisco Platform Exchange Grid 2.0**  
  
The Cisco Platform Exchange Grid (pxGrid) allows you to integrate your application into the pxGrid, a multivendor, cross-platform network system that pulls together different parts of an IT infrastructure such as security monitoring and detection systems, network policy platforms, asset and configuration management, identity and access management platforms, to name a few.   
  
Cisco Platform Exchange Grid (pxGrid) 2.0 no longer uses the C or Java SDK as in pxGrid 1.0. Instead pxGrid 2.0 uses WebSocket and REST API over the STOMP messaging protocol for querying and subscribing to topics.   
  
This Sandbox focuses on seeing what context is available on ISE via pxGrid subscribing to the Session Directory and RADIUS failure topics.   
  
This sandbox contains the following:

* Identity Services Engine 2.4 (10.10.20.70)
* Ubuntu DevBox (10.10.20.50)
* Windows Directory and DNS Server (10.10.20.100)
* Windows 7 Server (10.10.20.90)
* **Server Access Credentails**
*  Identity Services Engine 2.4 (admin/C1sco12345!)
*  Ubuntu DevBox (administrator/Cisco1234!)
*  Windows Directory (ABC\Administrator - 1vtG@lw@y)
*  Windows 7 Server (pxgrid/Cisco1234!)  
  The attributes tab of each resource also shows credential information. Resources can also be opened using the Gwac interface. Connect to the VPN, hover over the resource and select the connection type.

**More information:**

<https://developer.cisco.com/site/pxgrid/>

<https://developer.cisco.com/docs/pxgrid/#!learning-pxgrid/welcome-to-learning-cisco-platform-exchange-grid-pxgrid>

**Additional Information:**

https://communities.cisco.com/community/developer/sandbox

Sandbox Documentation

**Using this Sandbox**  
  
This document is for readers to accessing the Cisco Platform Exchange Grid (pxGrid) 2.0 sandbox environment and using their development platform. Eclipse is used in this document as the development platform and uses the pxGrid 2.0 java coding examples from

https://developer.cisco.com/site/pxgrid/   
  
This sandbox focuses on seeing what context is available on ISE via pxGrid subscribing to the Session Directory and RADIUS failure topics. Topics such as Adaptive Network Control (ANC) mitigation actions, and pxGrid Context-In, dynamic topics, and subscription to other TrustSec topics is not available in this release. To learn more about these topics please visit

https://developer.cisco.com/site/pxgrid/

to learn more information on these topics. These topics will be added in a later release.  
  
Security solutions that are going down the pxgrid certification path, they still would need to download the ISE 2.4 code and use RADIUS Simulator if there is no Cisco Catalyst Switch that supporting IEEE-802.1X and Change of Authorization (CoA) for ANC mitigation actions. First, you must register and login to Cisco Devnet   
  
To start using this sandbox and exploring pxGrid, see the

https://devnetsandbox.cisco.com/sandbox-instructions/PxGrid\_Revn/pxgrid\_sandbox20\_updated2.pdf

DevBox

**DevBox Overview:**  
  
The sandbox also comes with an Ubuntu server, named devbox, that comes with a ready-to-use Node.js development environment, including: the Node.js v6+ runtime and npm package, a git client, and the ngrok tunnelling client. This can be handy to use if you are willing to experiment with the jsxapi and do not have a Node.js environment on your laptop.   
  
**DevBox Credentials**  
  
IP Address: 10.10.20.150   
Username: administrator   
Password: Cisco1234!

**Testing process**

Chrome POSTMAN will be used to illustrate Cisco pxGrid Client Flow using REST API.

Enabling pxGrid on the ISE Node

Step 1 - Enable pxGrid

**Administration > System Deployment > Edit Node > Enable pxGrid**



Step 2 - Select Save

Step 3 - Verify that the ISE published nodes appear

**Administration > pxGrid Services**

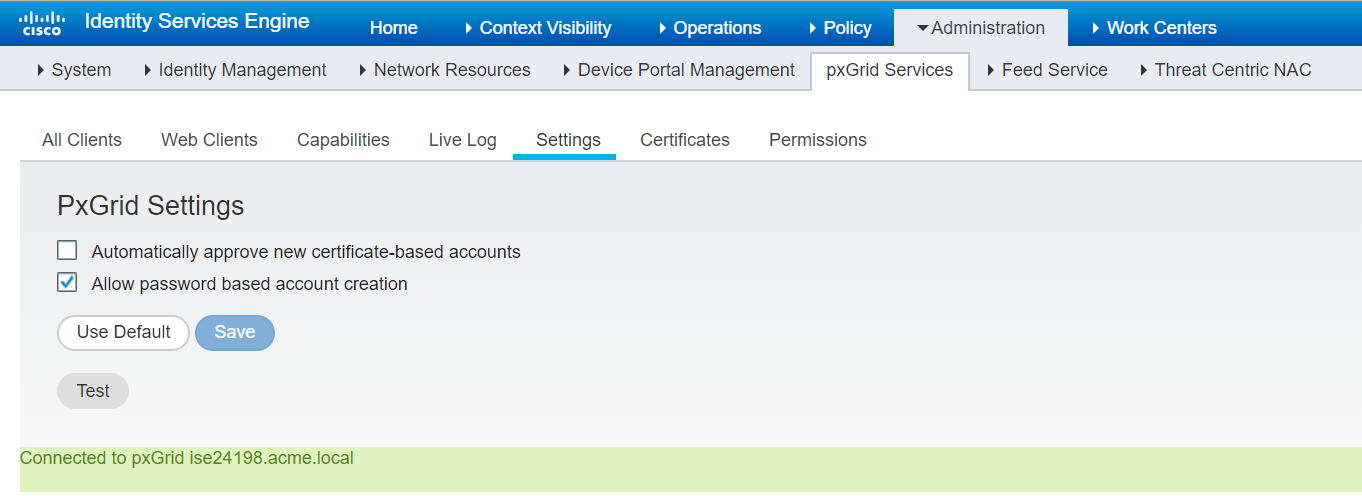
Verify that you have connectivity

Import the ISE identity certificates into your browser

Step 1 - On the workstation setup trust of Cisco ISE. Created trust with ISE Server

Step 2 - Ensure that you have “Allowed password based account creation” enabled under

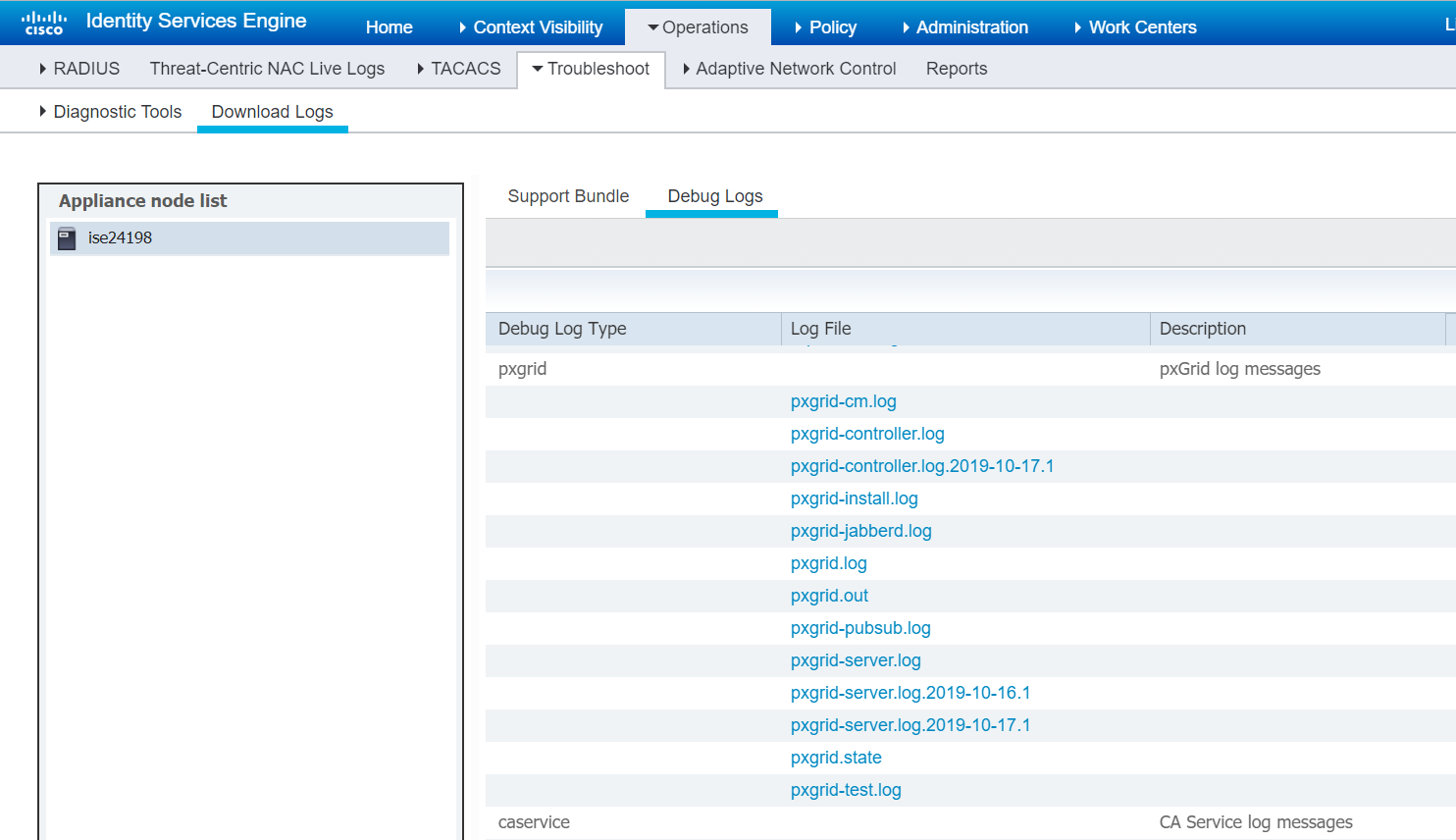
**Administration > pxGrid Services > Settings > pxGrid Settings**



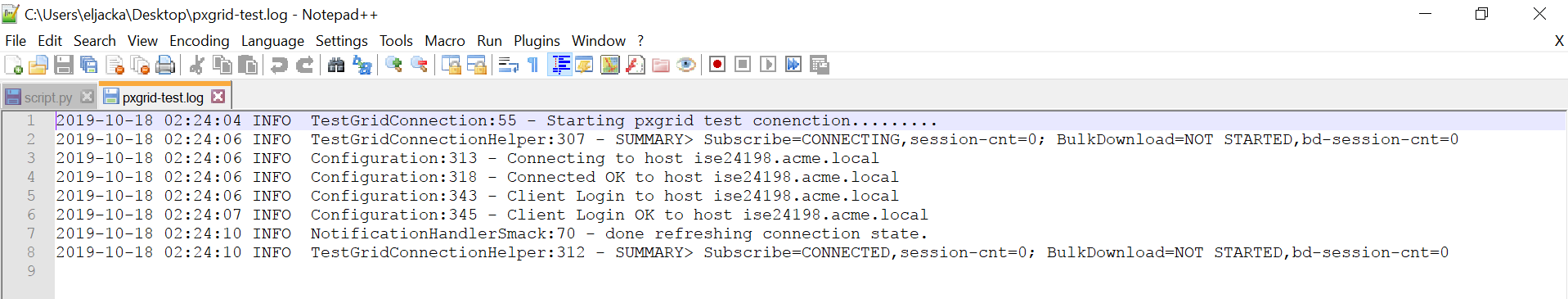
Test ….



View Log



View Log



What exactly does this test do?

Step 3 - Close out the browser and log back in again

\*\* What are we trusting the Certificate and Allowing password based Authentication.

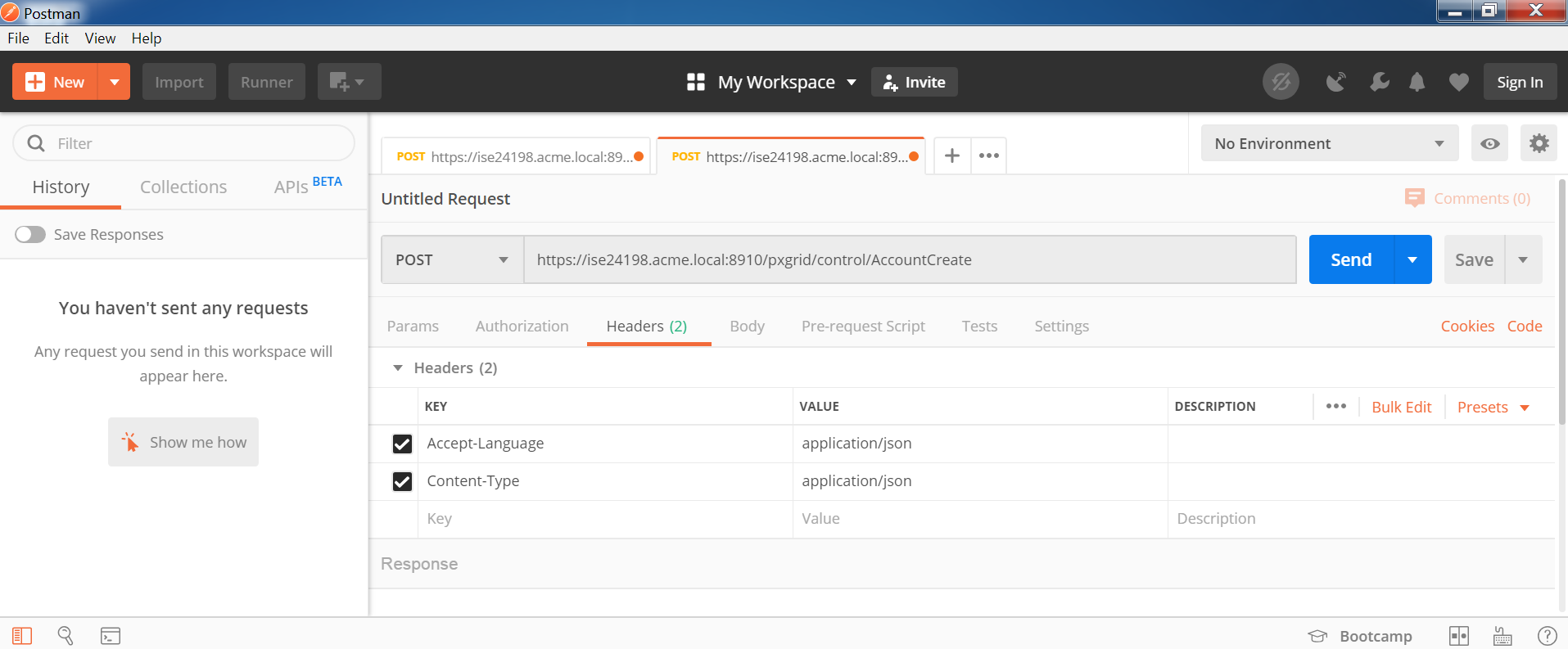
Creating the pxGrid client Account

The initial pxGrid client account will be created. The username and password returned 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.

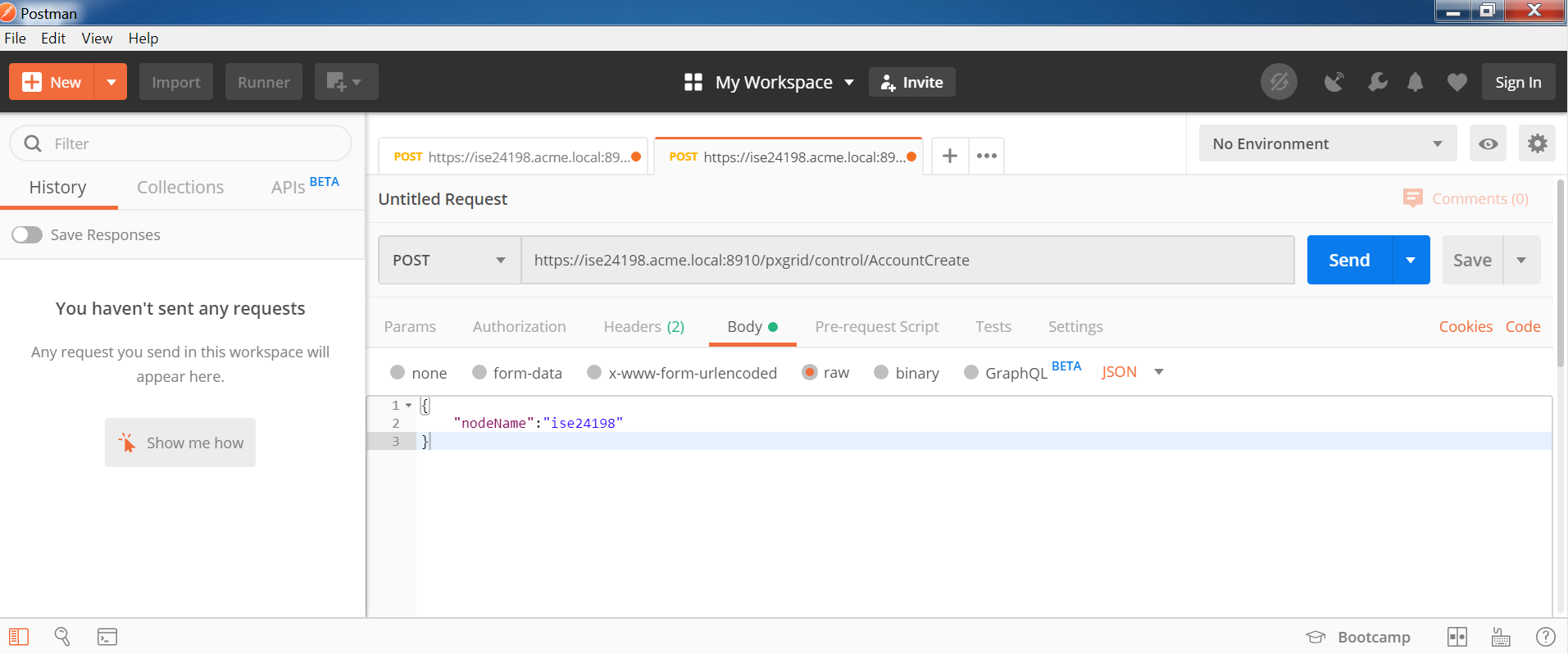
**https://ise24198.acme.local:8910/pxgrid/control/AccountCreate**

Step 2 - Define the following Headers:

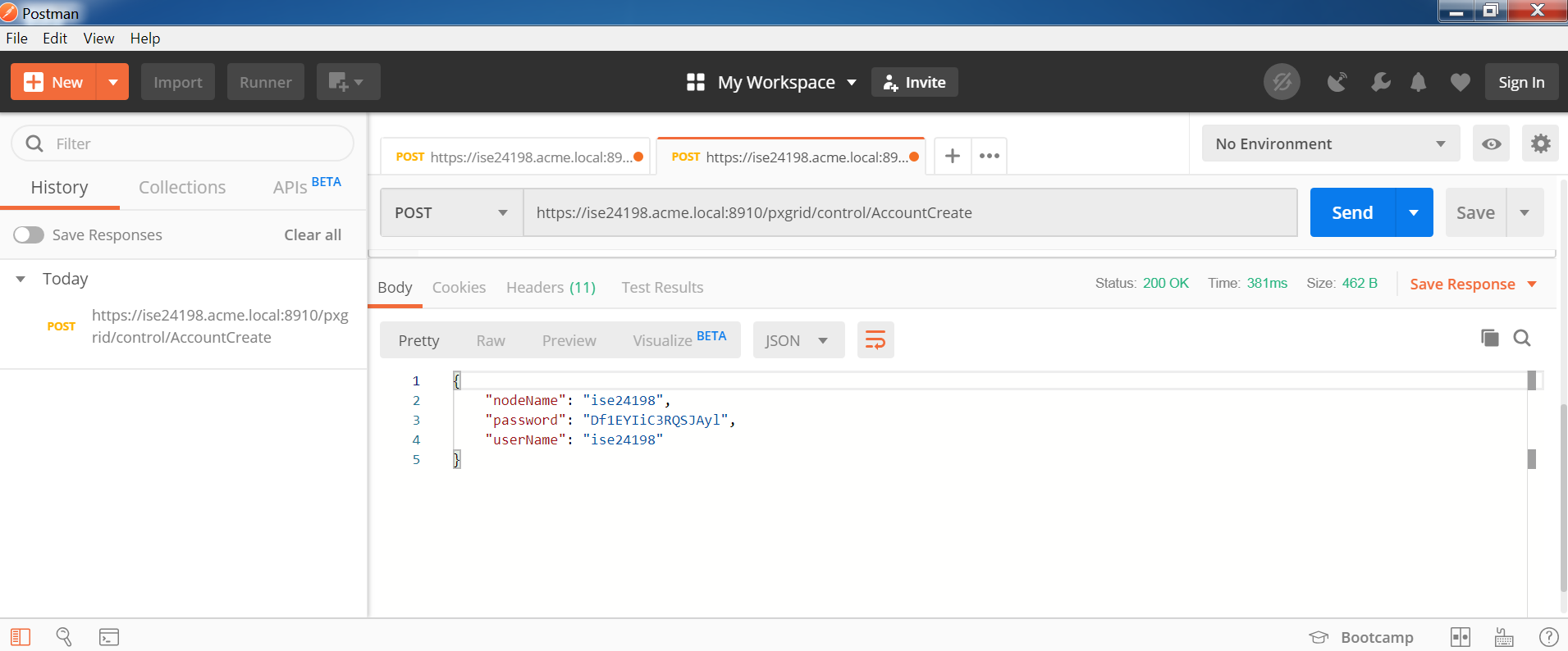


Step 3 - The body should read:

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



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.



Enabling the Account <<< This is in the list of things to be done but no instructions????

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

**https://ise24198.acme.local:8910/pxgrid/control/AccountActivate**

Step 2 - Add the following authorization settings

Note: The username and password are from step 6 above, or the results of the account creation.

**{**

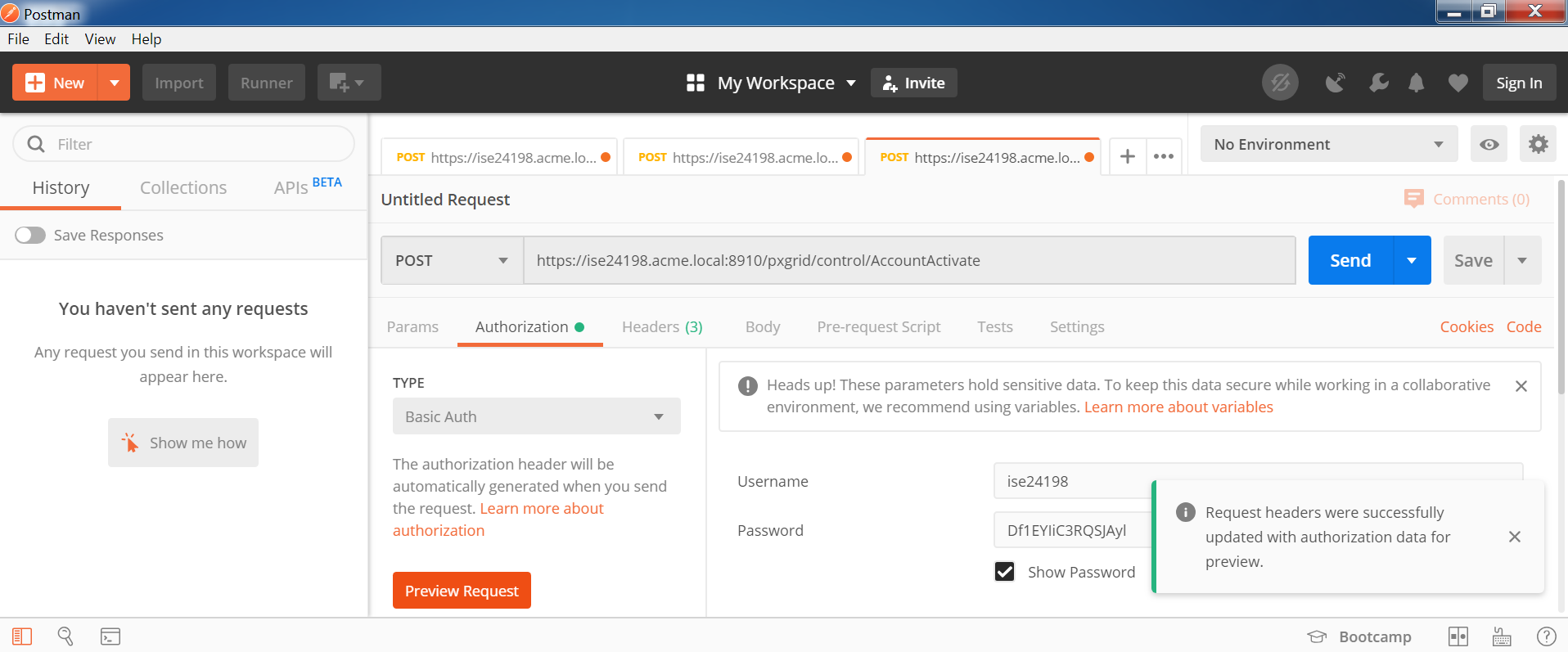
**"nodeName": "ise24198",**

**"password": "Df1EYIiC3RQSJAyl",**

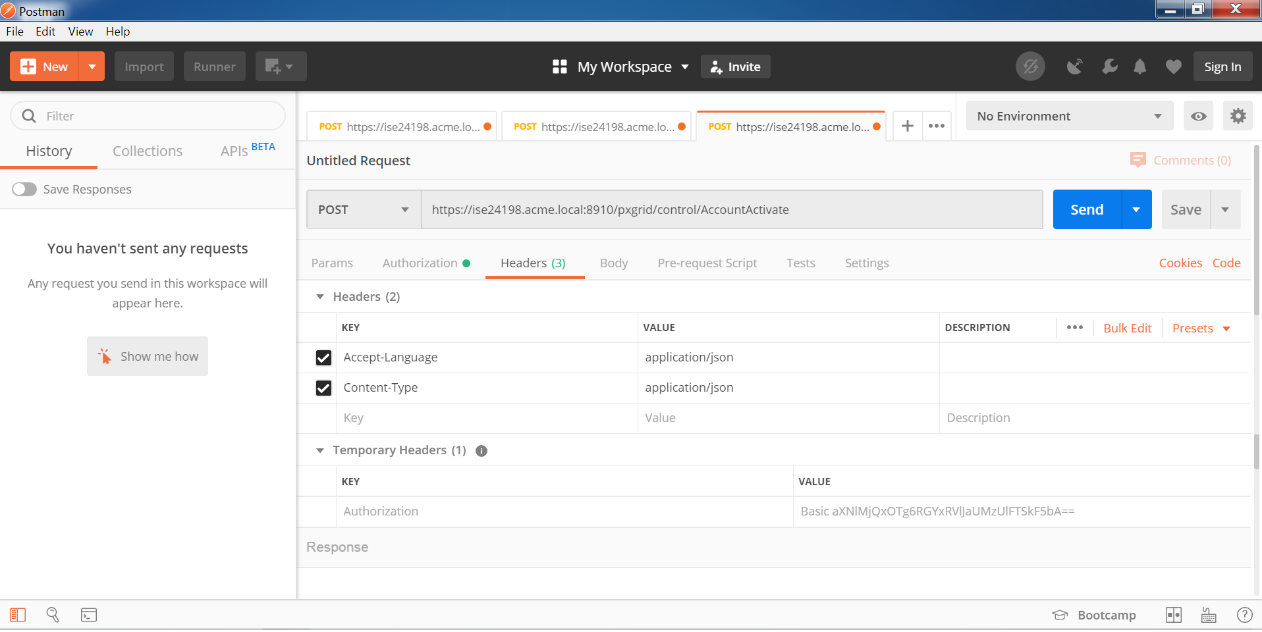
**"userName": "ise24198"**

**}**

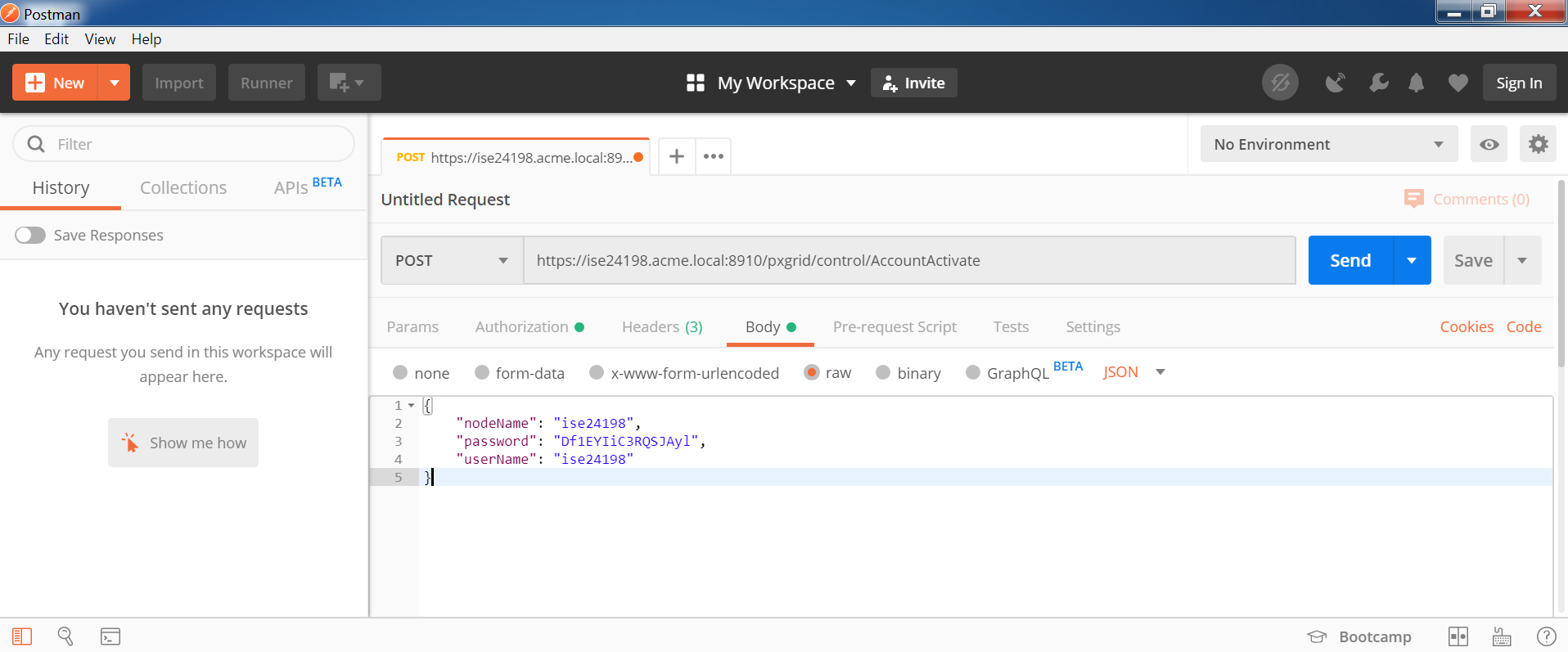
Line



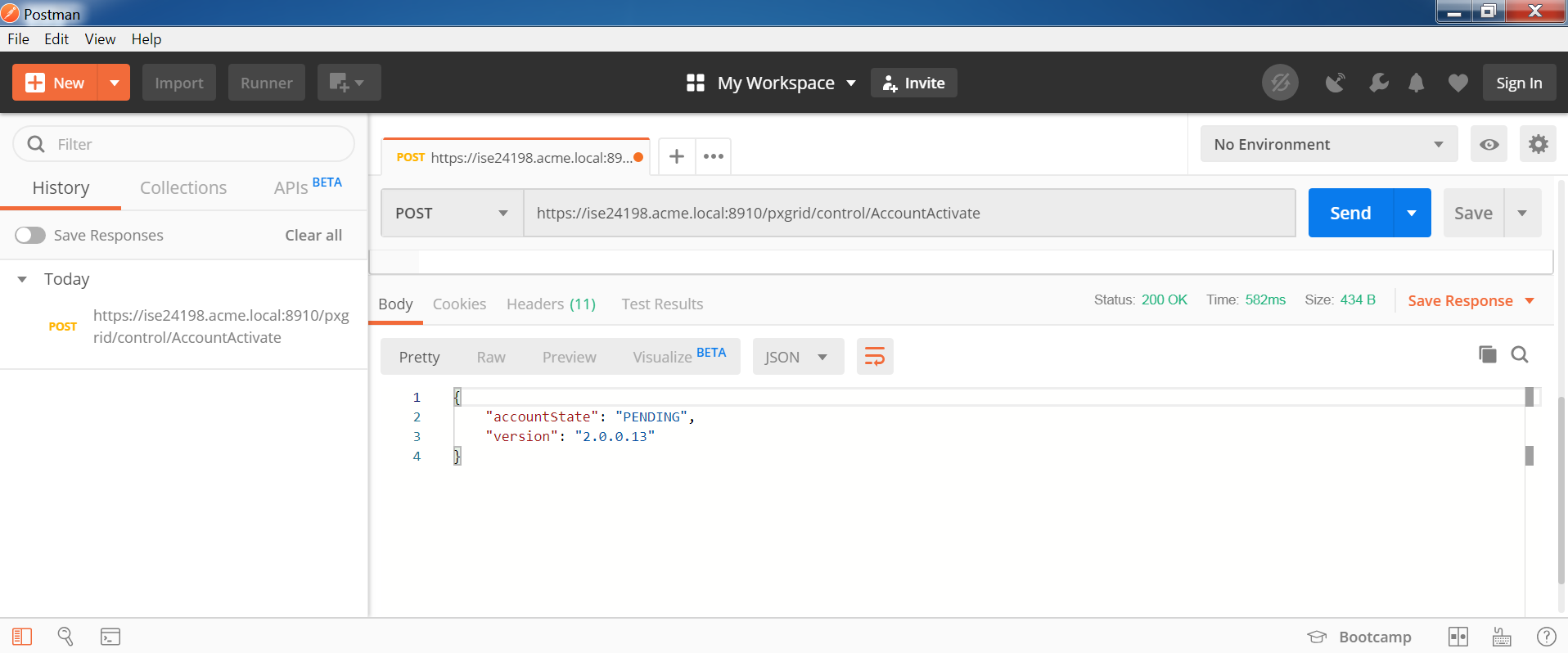
Step 3 Add the following headers, the authorization header will appear after the username and password have been configured.



Line



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

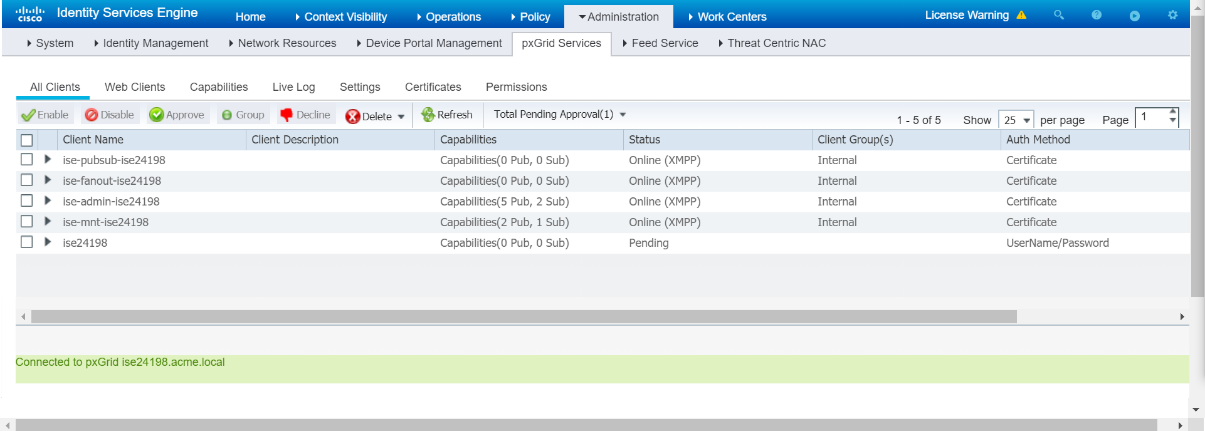


On ISE see the account

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

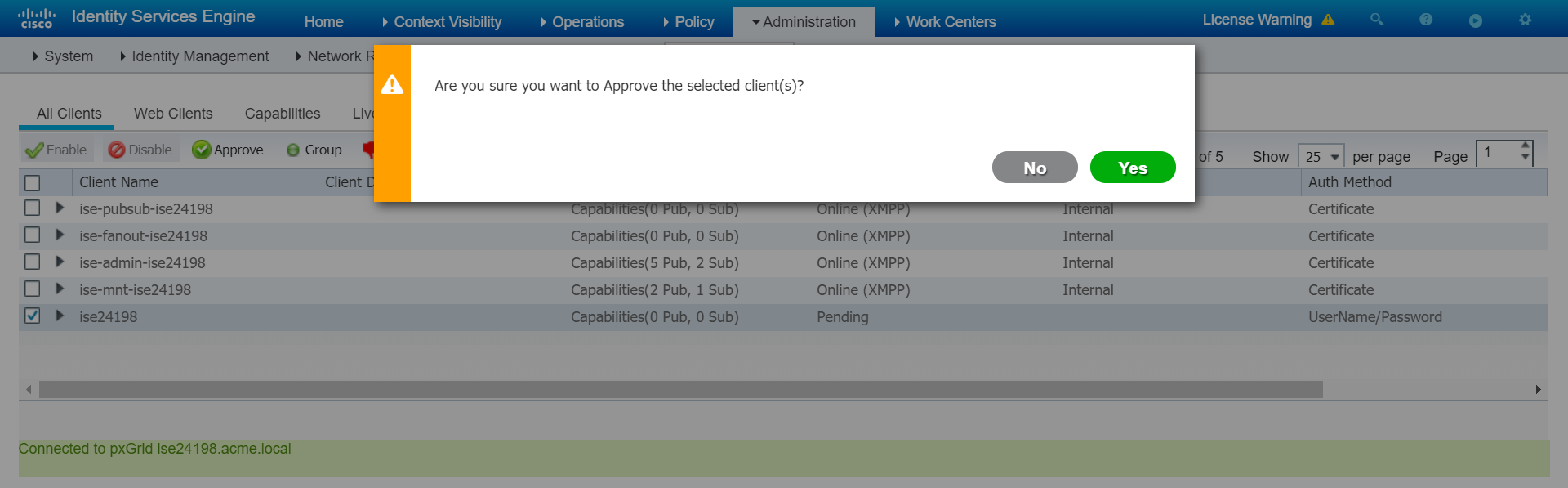
Note: Regardless of enabled password-based authentication for client settings, the admin will still have to approve the pending client request.

Step 6 - Select “ise24198” and “Approve”

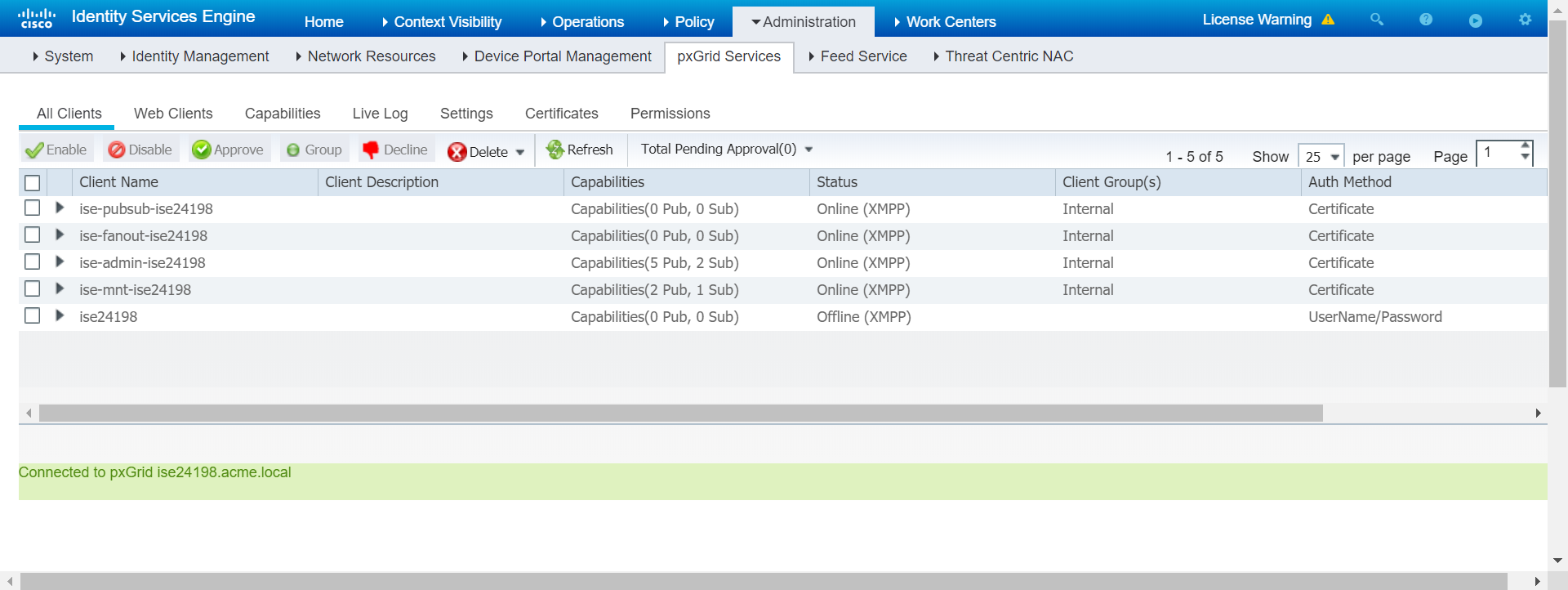


Step 7 - You will see the following message:

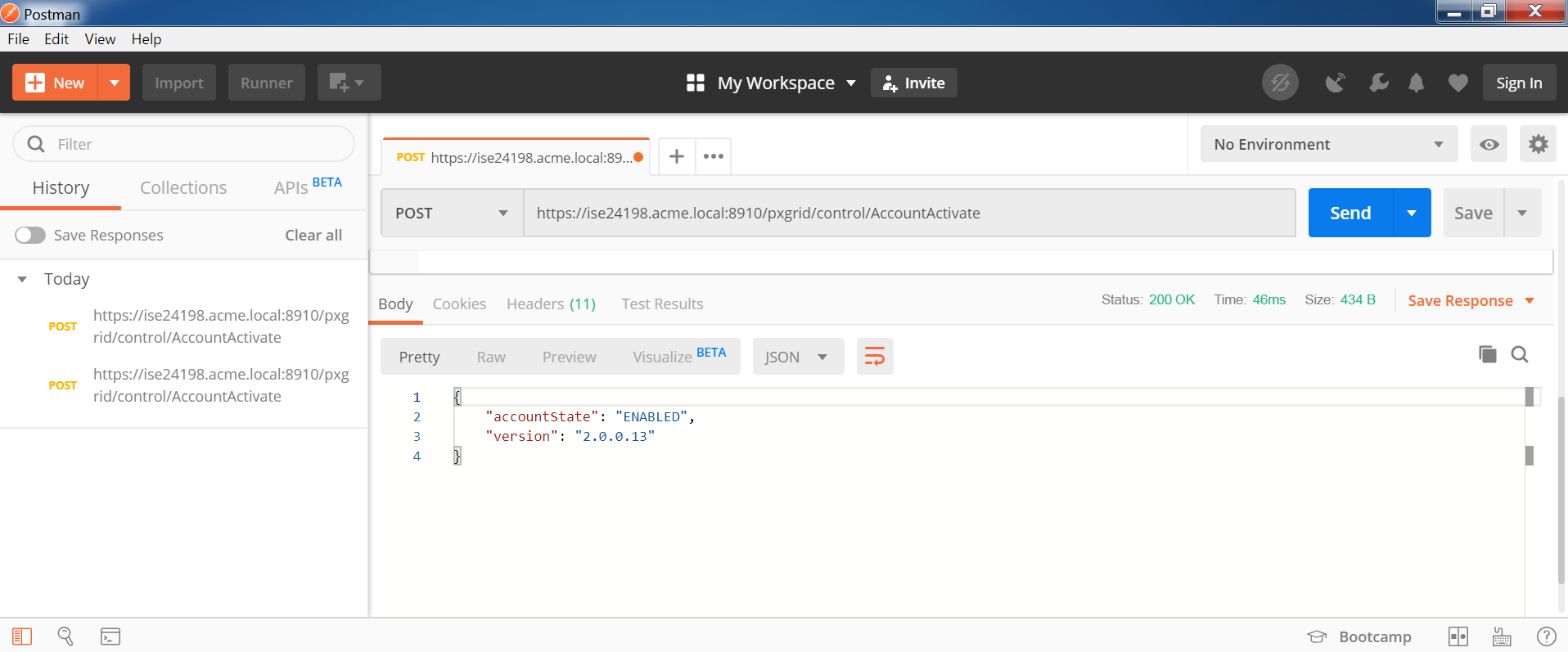
Step 8 Select “Yes”



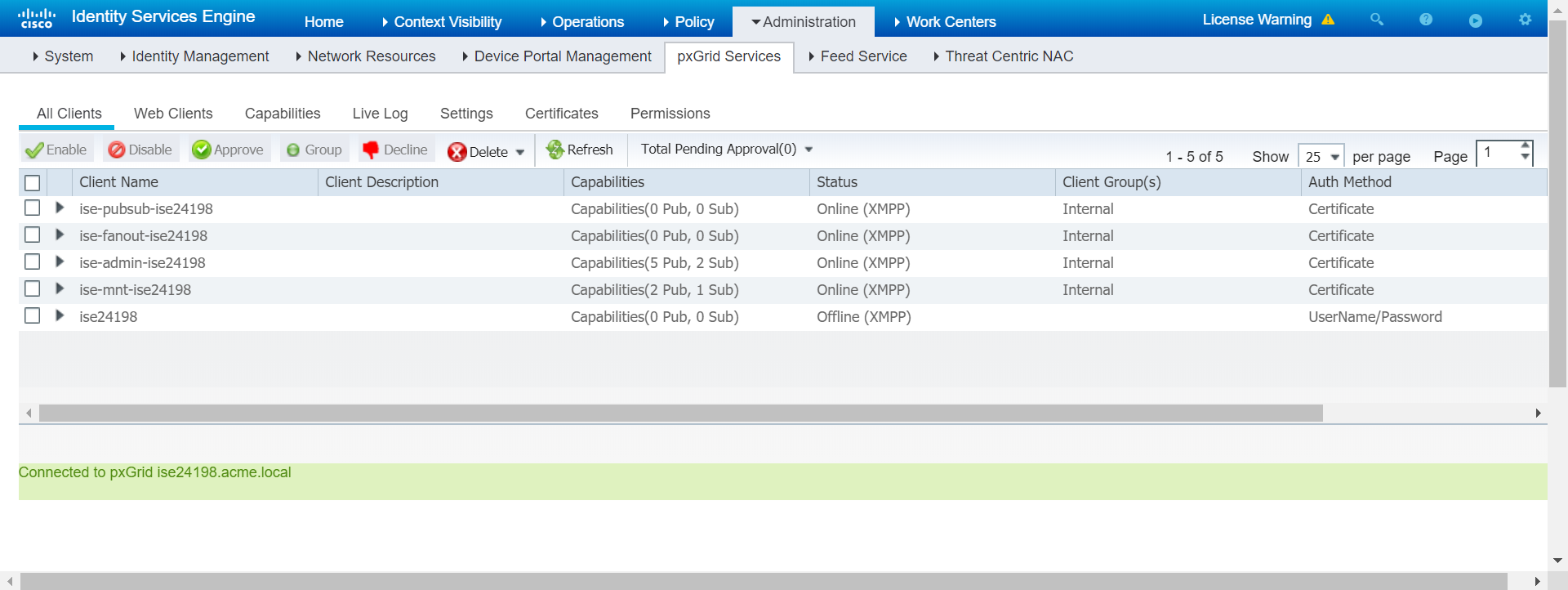
Step 9 You should see:



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



\*\*\* Why do you see the client as offline – was under the impression that the client had bidirectional communication once registered and a connection is established.



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.

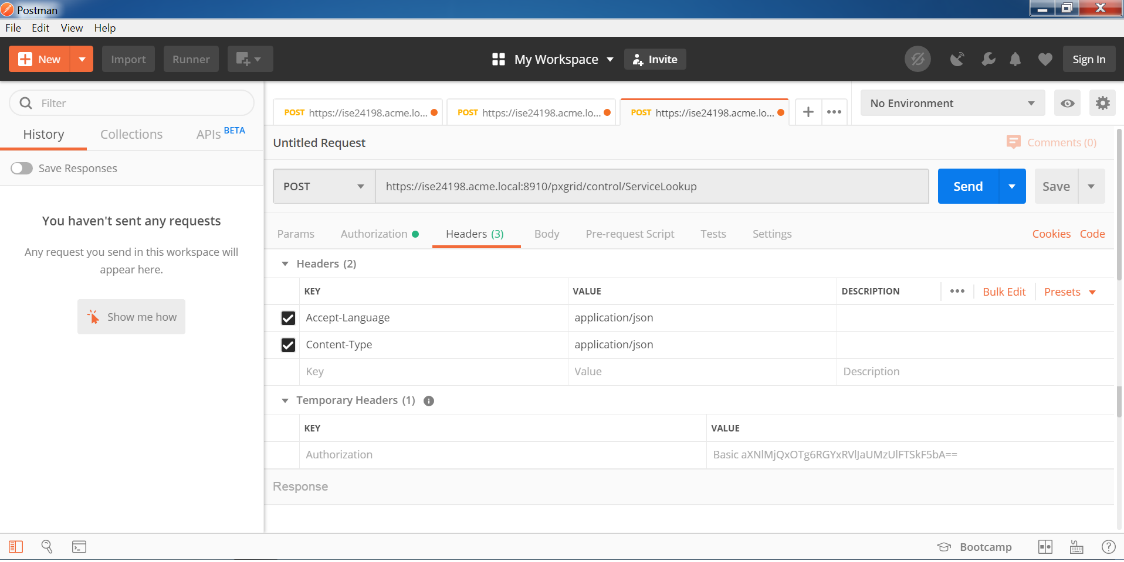
For more information see: https://github.com/cisco-pxgrid/pxgrid-rest-ws/wiki/pxGrid-Consumer

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

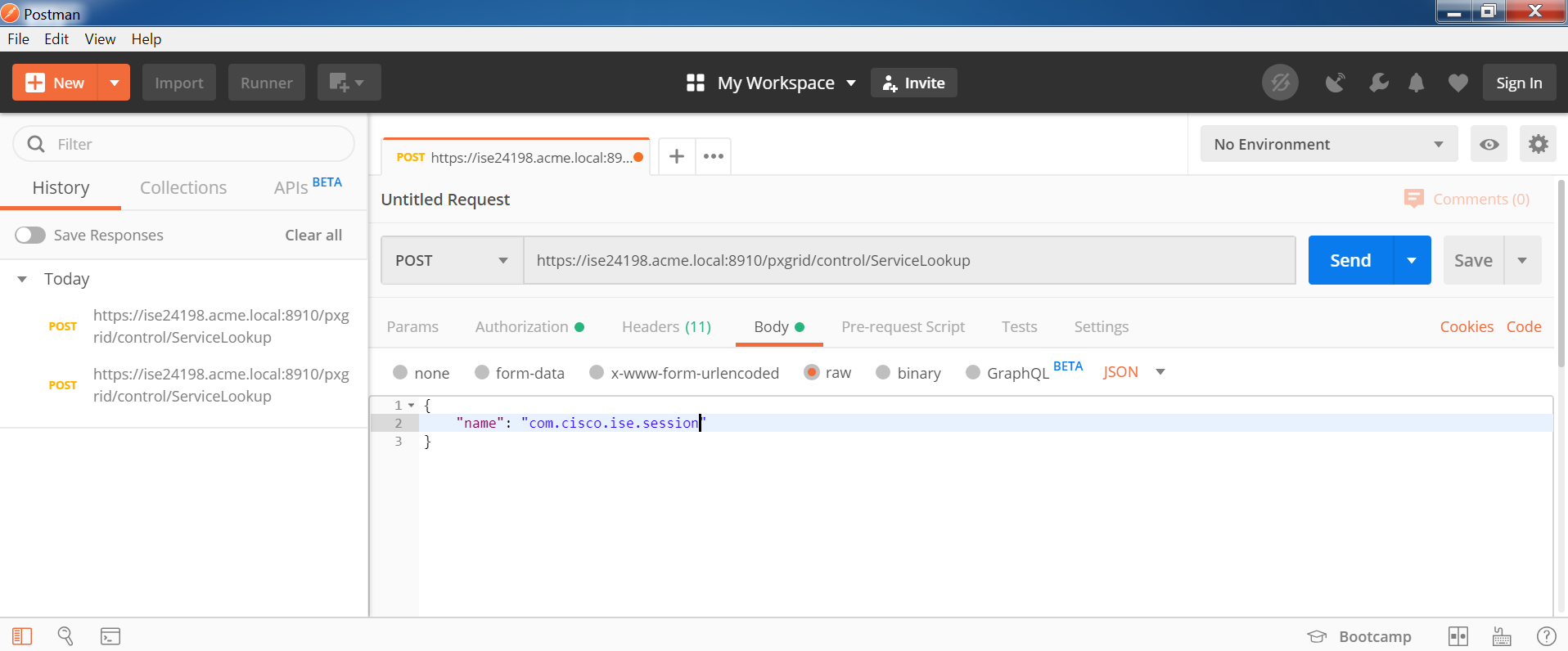
**https://ise24198.acme.local:8910/pxgrid/control/ServiceLookup**

Step 2 - The Headers should read:

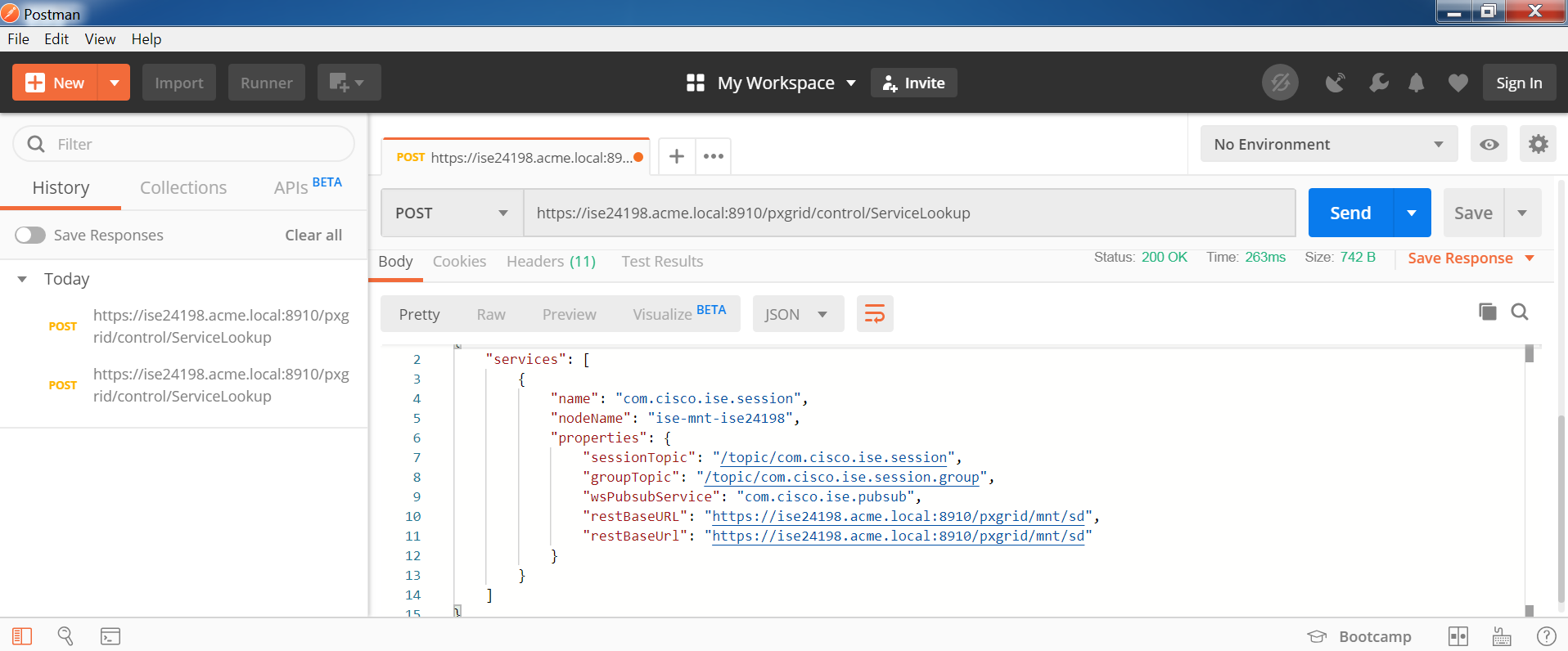


Step 3 - We request session topic which can be found here:

**https://github.com/cisco-pxgrid/pxgrid-restws/wiki/Session-Directory**



Result



Obtaining the Access Secret

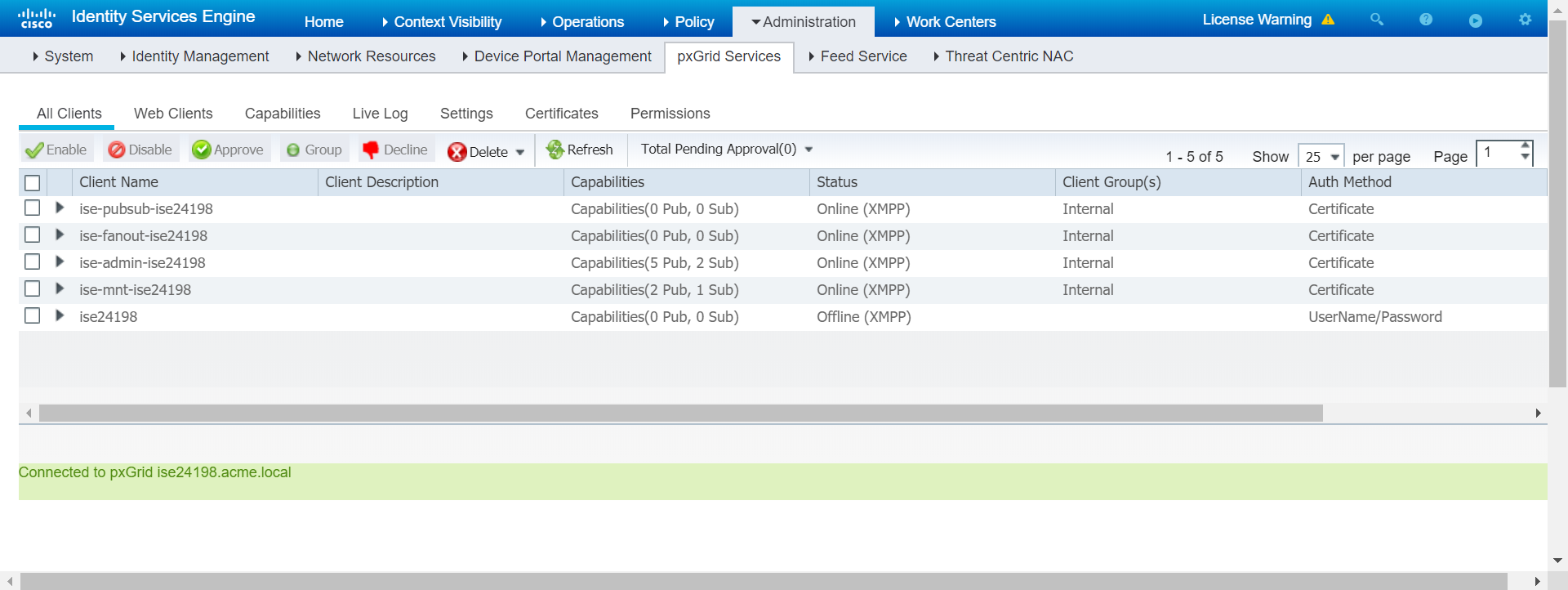
Run GetSessionByIPAddress WebSockets REST API script

**Questions:**

I had a certificate generated

In the intro there was a discussion about pxGrid 1.0 used XMPP – the image below us from ISE 2.4.

Initially pxGrid was released with ISE 1.3 and was XMPP-based, requiring an SDK containing the Java and C libraries, and sample code. This is now named pxGrid 1.0 and supported in ISE versions 1.3 and higher.



\*\* Extensible Messaging and Presence Protocol (XMPP) is a communication protocol for message-oriented middleware based on XML. It enables the near-real-time exchange of structured yet extensible data between any two or more network entities.