Implementation of Two way SSL interface in Data Power & Implementation of Oauth in CA API Gateway

By

Saikumar Maddula
Data Power Developer
Miracle Software Systems, Inc.

Sreevidya poola
APIM Developer
Miracle Software Systems, Inc.

SSL HANDSHAKE

SSL(Secure Socket Layer)

The Secure Sockets Layer (SSL) is a commonly used protocol for managing the security of a **message transmission** on the Internet.
Trust makes all the difference in the world of online business. Investment in technology to protect customers and earn their trust is a critical success factor for any e-commerce web site. The effective implementation of SSL Certificates and correct placement and use of trust marks are proven tools in the establishment of consumer trust.

1.1 WHAT IS AN SSL CERTIFICATE?

An SSL Certificate is a digital computer file (or small piece of code) that has two specific functions:

1. Authentication and Verification: The SSL Certificate has information about the authenticity of certain details regarding the identity of a person, business or web site, which it will display to visitors on your web site when they click on the browser's padlock symbol or trust mark

2. Data Encryption: The SSL Certificate also enables encryption, which means that the sensitive information exchanged via the web site cannot be intercepted and read by anyone other than the intended recipient.

The Process: Every SSL Certificate is issued for a specific server and web site domain (web site address) for a CA-verified entity. When a person uses their browser to navigate to the address of a web site with an SSL Certificate, an SSL handshake (greeting) occurs between the browser and server. Information is requested from the server—which is then made visible to the person in their browser. You will notice changes in your browser. If you click on the trust mark, you will see additional information such as the validity period of the SSL Certificate, the domain secured, the type of SSL Certificate, and the issuing CA. A secure link is established for that session, with a unique session key, and secure communications can begin.

1. A standard web site **without SSL security** displays "http://" before the web site address in the browser address bar. This moniker stands for "Hypertext Transfer Protocol," and is the conventional way to transmit information over the Internet
2. However, a web site that is **secured with a SSL Certificate** will display "https:// " before the address. This stands for "Secure HTTP."
3. You will also see a padlock symbol on the top or bottom of the Internet browser (depending on which browser you are using).

4. When you click on the VeriSign seal or the padlock symbol on the page, it will display details of the relevant certificate with all of the company information as verified and authenticated by the CA.
5. By clicking the closed padlock in the browser window, or certain SSL trust marks (such as the VeriSign seal), the web site visitor sees the authenticated organization name. In high-security browsers, the authenticated organization name is prominently displayed and the address bar turns green when an Extended Validation (EV) SSL Certificate is detected. If the information does not match, or the certificate has expired, the browser displays an error message or warning.
The above mentioned is a simple example to know the **ssl certificate** and **process of ssl CA .** Each and every certificate should send and verified by Certification authority.
As I had taken example of VeriSign certificates, it is the world's leading provider of SSL Certificates, and over 90,000 domains in 160 countries display the VeriSign seal, the most recognized trust mark on the Internet.

1.2 WHERE WOULD I USE AN SSL CERTIFICATE?

Answer to this question is that you would use an SSL Certificate anywhere that you wish to transmit information securely.

Here are some examples:

- Securing communication between your web site and your customer's Internet browser.
- Securing internal communications on your corporate intranet.
- Securing email communications sent to and from your network (or private email address).
- Securing information between servers (both internal and external).

1.2 **DIFFERENT TYPES OF SSL CERTIFICATES:** There are a number of different SSL Certificates on the market today. Some of them are as followed,

- Self-signed certificate
- Domain Validated Certificate
- Fully authenticated SSL Certificate
- Extended Validation (EV) SSL Certificates ., etc…

When to create SSL Proxy with a SSL Direction as:

1.3.1 REVERSE: Create a Reverse Crypto profile when you need just front side handler. And in this case, create the Identification Credentials with the Crypto Key and Certificate of the DataPower box itself. Validation credential is mostly not required.

1.3.2 TWO-WAY: In this case, create a Front Crypto Profile along with the Reverse Crypto Profile. Forward Crypto Profile is needed to connect to a back-end server. For creating the Forward Crypto Profile, Identification credential are mostly not required. Validation Credentials will contain the actual certificate of the back-end to connect to.

1.3.3 FORWARD: Create a Forward Crypto profile as detailed above. A Forward Crypto profile alone will be created if the Front Side handler does not need to validate the request send from the client to the DataPower box. Mostly you will create a TWO-WAY proxy instead of just the FORWARD proxy.

- Data that is transmitted over SSL connections are encrypted by using session keys that are secured with public key cryptography. Public key cryptography requires a public key (store in the certificate) and a private key.
- DataPower uses a Crypto Identification Credential to associate or match a public key and private key for use in cryptographic operation such as SSL.
- DataPower uses Validation Credentials to validate digital signatures and received signatures. A Validation Credential is a list of certificate objects.
- Crypto Profile associates a Crypto identification with a Crypto Identification credentials.

Implementation of Two way SSL interface in Data Power

1. Introduction

A multiprotocol gateway connects client requests that are transported over one or more protocols to a back-end service by using the same or a different protocol.

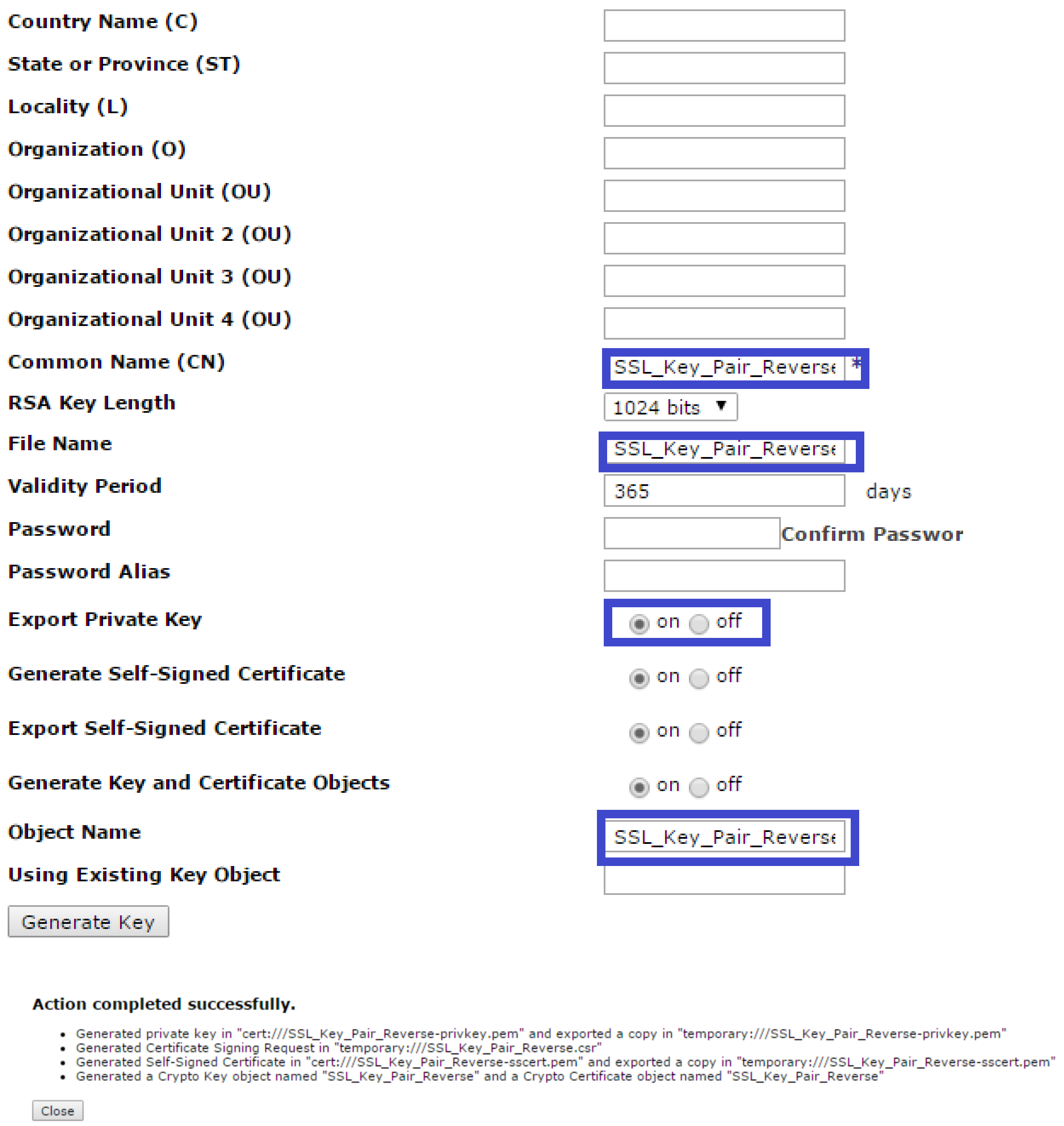
2. Logging into WebGUI

➢ Login into the DataPower WebGUI with user id and password along with the Domain.

3. Creation of Certificate and Key

Search for **Crypto Tools** from Navigation Bar

Click on "**Generate Key**" then you will get the following pop up.



Country Name (C)	
State or Province (ST)	
Locality (L)	
Organization (O)	
Organizational Unit (OU)	
Organizational Unit 2 (OU)	
Organizational Unit 3 (OU)	
Organizational Unit 4 (OU)	
Common Name (CN)	SSL_Key_Pair_Reverse *
RSA Key Length	1024 bits ▼
File Name	SSL_Key_Pair_Reverse
Validity Period	365 days
Password	Confirm Passwor
Password Alias	
Export Private Key	[✓ on] off
Generate Self-Signed Certificate	[✓ on] off
Export Self-Signed Certificate	[✓ on] off
Generate Key and Certificate Objects	[✓ on] off
Object Name	SSL_Key_Pair_Reverse
Using Existing Key Object	

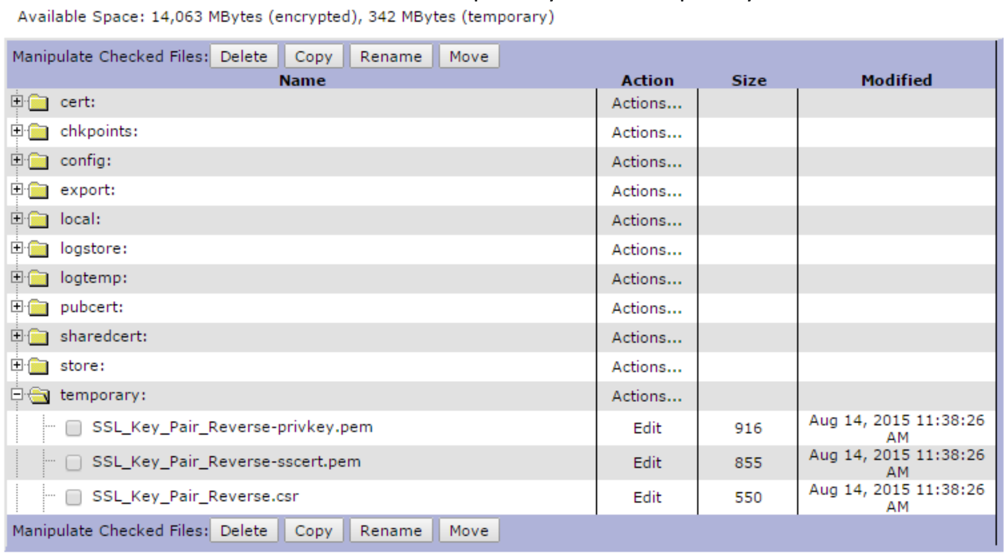
Generate Key

Action completed successfully.

- Generated private key in "cert:///SSL_Key_Pair_Reverse-privkey.pem" and exported a copy in "temporary:///SSL_Key_Pair_Reverse-privkey.pem"
- Generated Certificate Signing Request in "temporary:///SSL_Key_Pair_Reverse.csr"
- Generated Self-Signed Certificate in "cert:///SSL_Key_Pair_Reverse-sscert.pem" and exported a copy in "temporary:///SSL_Key_Pair_Reverse-sscert.pem"
- Generated a Crypto Key object named "SSL_Key_Pair_Reverse" and a Crypto Certificate object named "SSL_Key_Pair_Reverse"

Close

We can download those certificate and privkey from temporary folder from File Management.



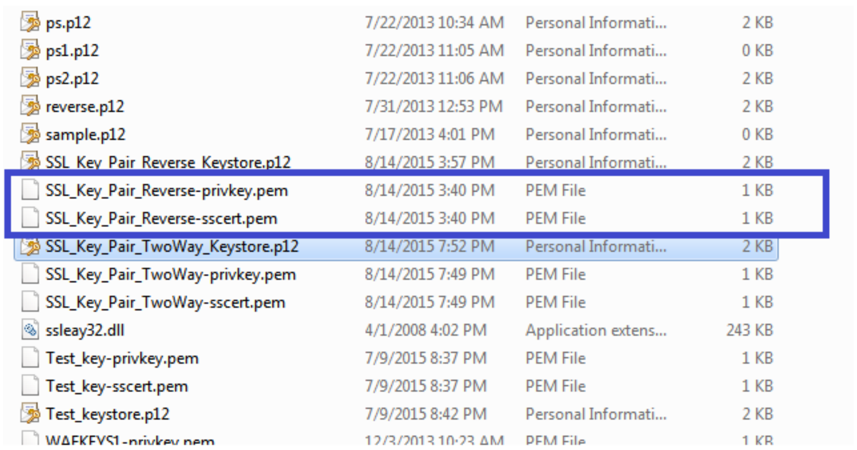
Available Space: 14,063 MBytes (encrypted), 342 MBytes (temporary)

Manipulate Checked Files: Delete Copy Rename Move

Name	Action	Size	Modified
cert:	Actions...		
chkpoints:	Actions...		
config:	Actions...		
export:	Actions...		
local:	Actions...		
logstore:	Actions...		
logtemp:	Actions...		
pubcert:	Actions...		
sharedcert:	Actions...		
store:	Actions...		
temporary:	Actions...		
SSL_Key_Pair_Reverse-privkey.pem	Edit	916	Aug 14, 2015 11:38:26 AM
SSL_Key_Pair_Reverse-sscert.pem	Edit	855	Aug 14, 2015 11:38:26 AM
SSL_Key_Pair_Reverse.csr	Edit	550	Aug 14, 2015 11:38:26 AM

Manipulate Checked Files: Delete Copy Rename Move

Click on Name and it will open in browser. Now right click on the page Click on Save as option and save it in Open SSL--> bin folder as follows

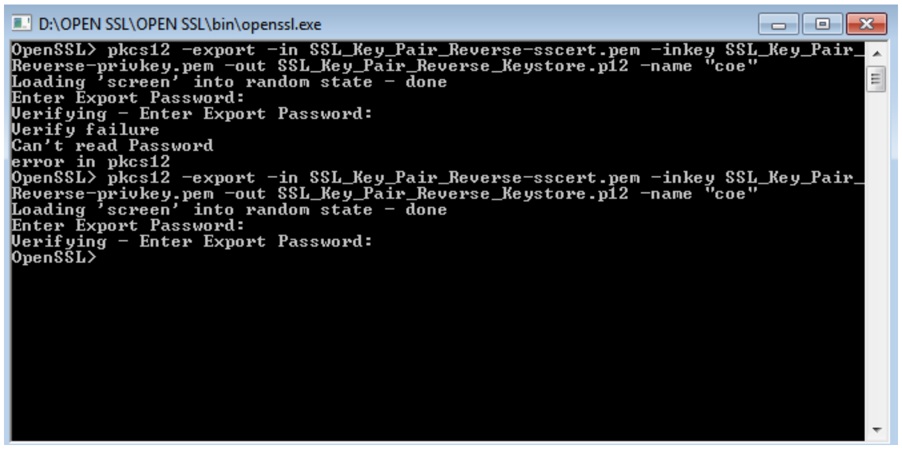


ps.p12	7/22/2013 10:34 AM	Personal Informati...	2 KB
ps1.p12	7/22/2013 11:05 AM	Personal Informati...	0 KB
ps2.p12	7/22/2013 11:06 AM	Personal Informati...	2 KB
reverse.p12	7/31/2013 12:53 PM	Personal Informati...	2 KB
sample.p12	7/17/2013 4:01 PM	Personal Informati...	0 KB
SSL_Key_Pair_Reverse_Keystore.p12	8/14/2015 3:57 PM	Personal Informati...	2 KB
SSL_Key_Pair_Reverse-privkey.pem	8/14/2015 3:40 PM	PEM File	1 KB
SSL_Key_Pair_Reverse-sscert.pem	8/14/2015 3:40 PM	PEM File	1 KB
SSL_Key_Pair_TwoWay_Keystore.p12	8/14/2015 7:52 PM	Personal Informati...	2 KB
SSL_Key_Pair_TwoWay-privkey.pem	8/14/2015 7:49 PM	PEM File	1 KB
SSL_Key_Pair_TwoWay-sscert.pem	8/14/2015 7:49 PM	PEM File	1 KB
ssleay32.dll	4/1/2008 4:02 PM	Application extens...	243 KB
Test_key-privkey.pem	7/9/2015 8:37 PM	PEM File	1 KB
Test_key-sscert.pem	7/9/2015 8:37 PM	PEM File	1 KB
Test_keystore.p12	7/9/2015 8:42 PM	Personal Informati...	2 KB
WAFKEYS1-privkey.pem	12/3/2013 10:23 AM	PEM File	1 KB

4. Creation of the Keystore

Now open for **openssl.exe** file from OpenSSL--> bin--> openssl.exe

Now issue the following command
OpenSSL> **pkcs12 -export -in SSL_Key_Pair_Reverse-sscert.pem -inkey SSL_Key_Pair_Reverse-privkey.pem -out SSL_Key_Pair_Reverse_Keystore.p12 -name "coe"**



D:\OPEN SSL\OPEN SSL\bin\openssl.exe

```
OpenSSL> pkcs12 -export -in SSL_Key_Pair_Reverse-sscert.pem -inkey SSL_Key_Pair_
Reverse-privkey.pem -out SSL_Key_Pair_Reverse_Keystore.p12 -name "coe"
Loading 'screen' into random state - done
Enter Export Password:
Verifying - Enter Export Password:
Verify failure
Can't read Password
error in pkcs12
OpenSSL> pkcs12 -export -in SSL_Key_Pair_Reverse-sscert.pem -inkey SSL_Key_Pair_
Reverse-privkey.pem -out SSL_Key_Pair_Reverse_Keystore.p12 -name "coe"
Loading 'screen' into random state - done
Enter Export Password:
Verifying - Enter Export Password:
OpenSSL>
```

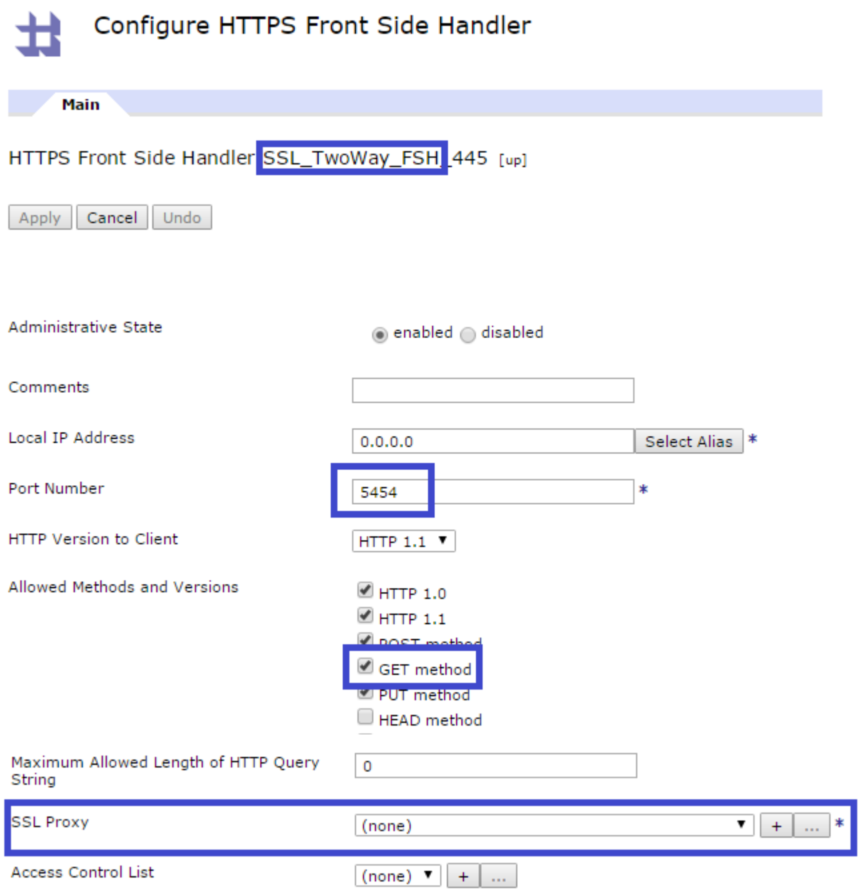
verifying enter export password: keystore(or anything) If you give wrong password it ll show **Can`t read password**
After completion execution of command you ll be create a file in bin folder with the name of **SSL_Key_Pair_Reverse_Keystore.p12** as follows

5. Creating an Multi Protocol Gateway Service ➢ From the list of Service icons (on the top row), click on the ***Multi Protocol Gateway*** icon. The Configure ***Multi Protocol Gateway*** page will be displayed.

5.1Configuring a Multi Protocol Gateway Service.

5.1.5 Configuring HTTPS Front Side handler

- ➢ Since in this scenario, We need to Configure the HTTPS FSH.
- ➢ Click on "+" on the Front Side Protocol tab.



Configure HTTPS Front Side Handler

HTTPS Front Side Handler SSL_TwoWay_FSH_445 [up]

Administrative State: [✓ enabled] disabled

Comments:

Local IP Address: 0.0.0.0 Select Alias *

Port Number: 5454 *

HTTP Version to Client: HTTP 1.1

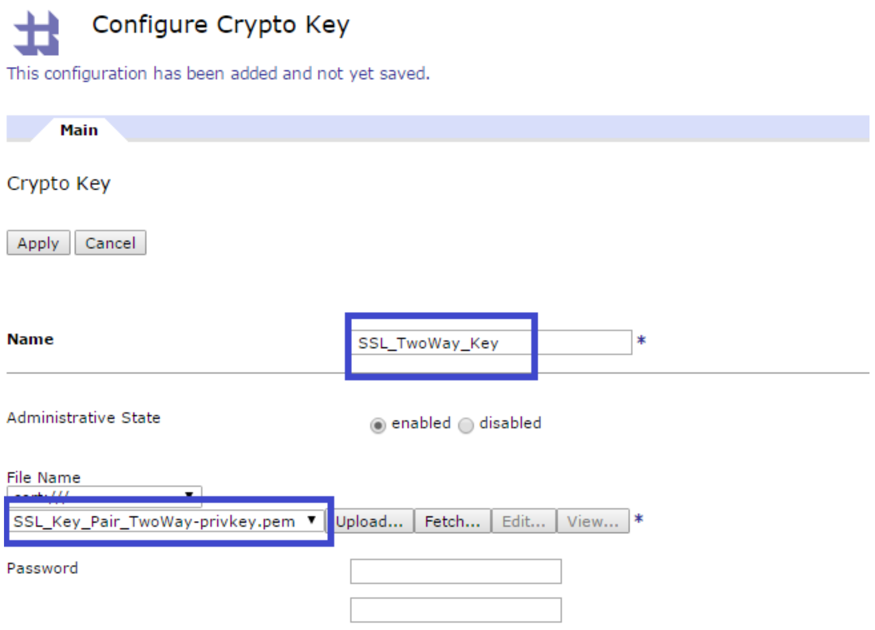
Allowed Methods and Versions:
- ✓ HTTP 1.0
- ✓ HTTP 1.1
- ✓ POST method
- ✓ GET method
- ✓ PUT method
- HEAD method

Maximum Allowed Length of HTTP Query String: 0

SSL Proxy: (none) + ... *

Access Control List: (none) + ...

Click on + symbol to **configure SSL Proxy.**
Specify the **name of SSL Proxy Profile .**
Specify the name for **Crypto Identification Credentials.**
Click on + symbol to **configure Crypto Key.**
Specify the **name for Crypto key** and choose the **certificate** which was created



Configure Crypto Key

This configuration has been added and not yet saved.

Crypto Key

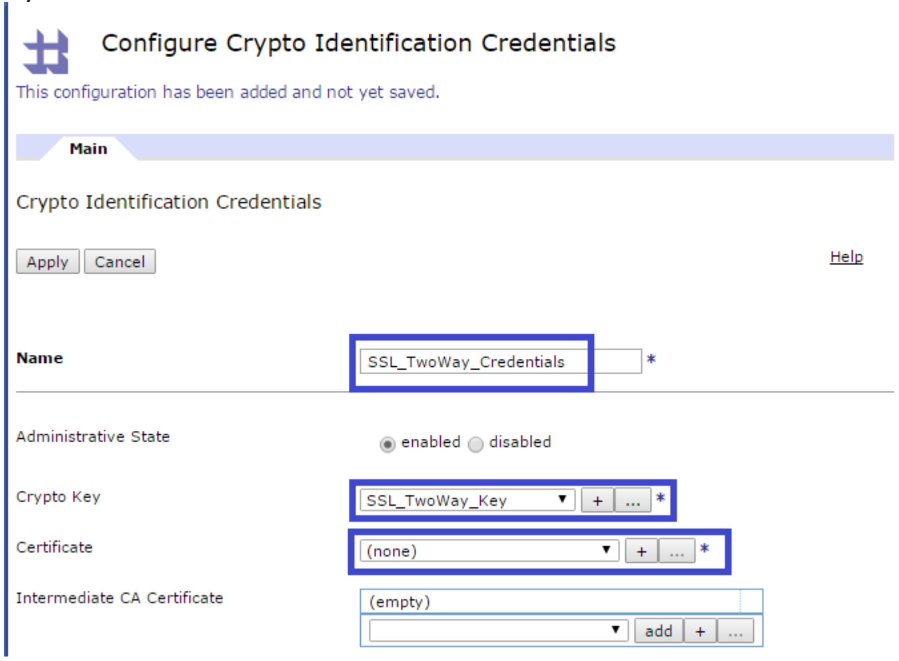
Name SSL_TwoWay_Key *

Administrative State [✓ enabled] disabled

File Name cert:/// SSL_Key_Pair_TwoWay-privkey.pem *

Password

by earlier as follows:



Configure Crypto Identification Credentials

This configuration has been added and not yet saved.

Crypto Identification Credentials

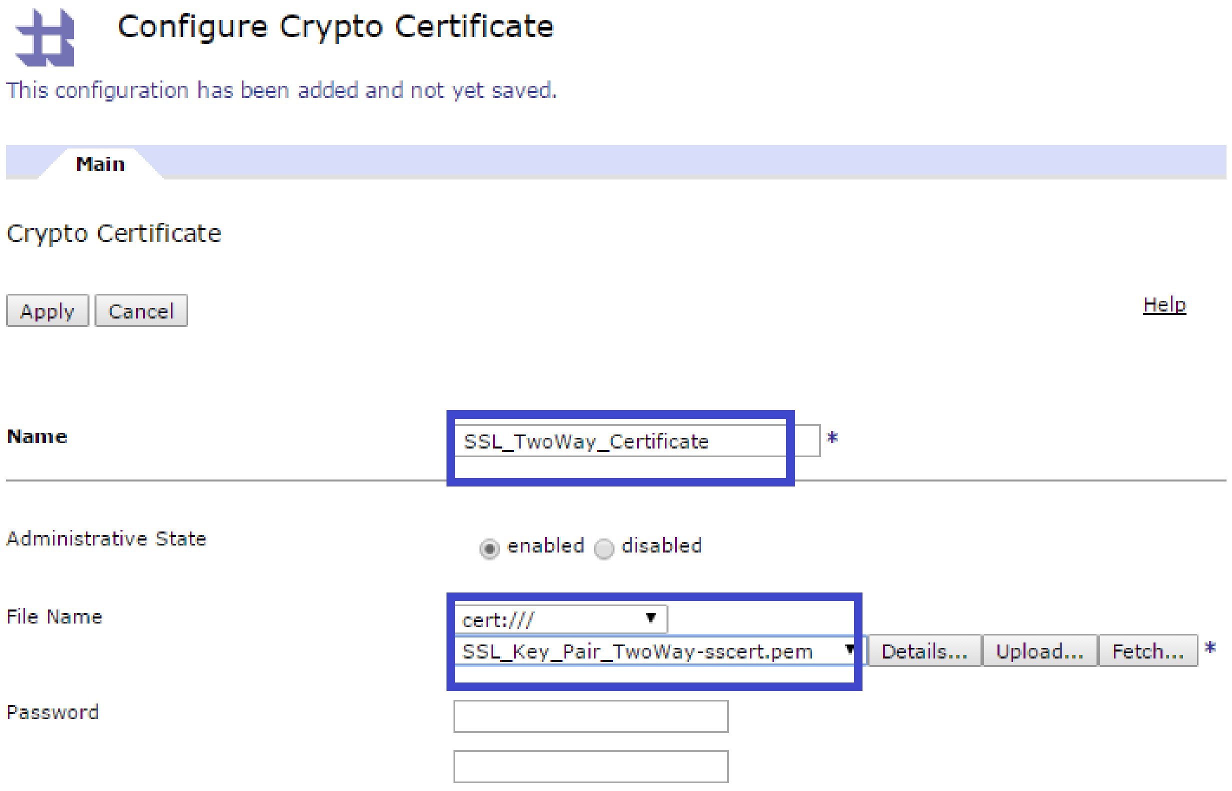
Name SSL_TwoWay_Credentials *

Administrative State [✓ enabled] disabled

Crypto Key SSL_TwoWay_Key *

Certificate (none) *

Intermediate CA Certificate (empty)

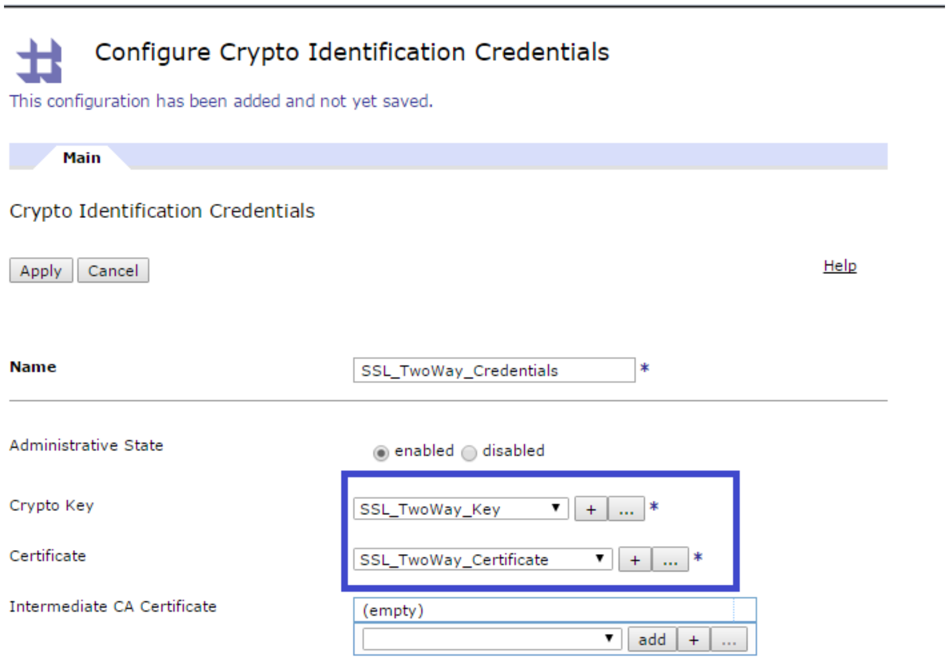


Configure Crypto Certificate

This configuration has been added and not yet saved.

Crypto Certificate

Name	SSL_TwoWay_Certificate *
Administrative State	[✓ enabled] disabled
File Name	cert:/// SSL_Key_Pair_TwoWay-sscert.pem *
Password	


Configure Crypto Identification Credentials

This configuration has been added and not yet saved.

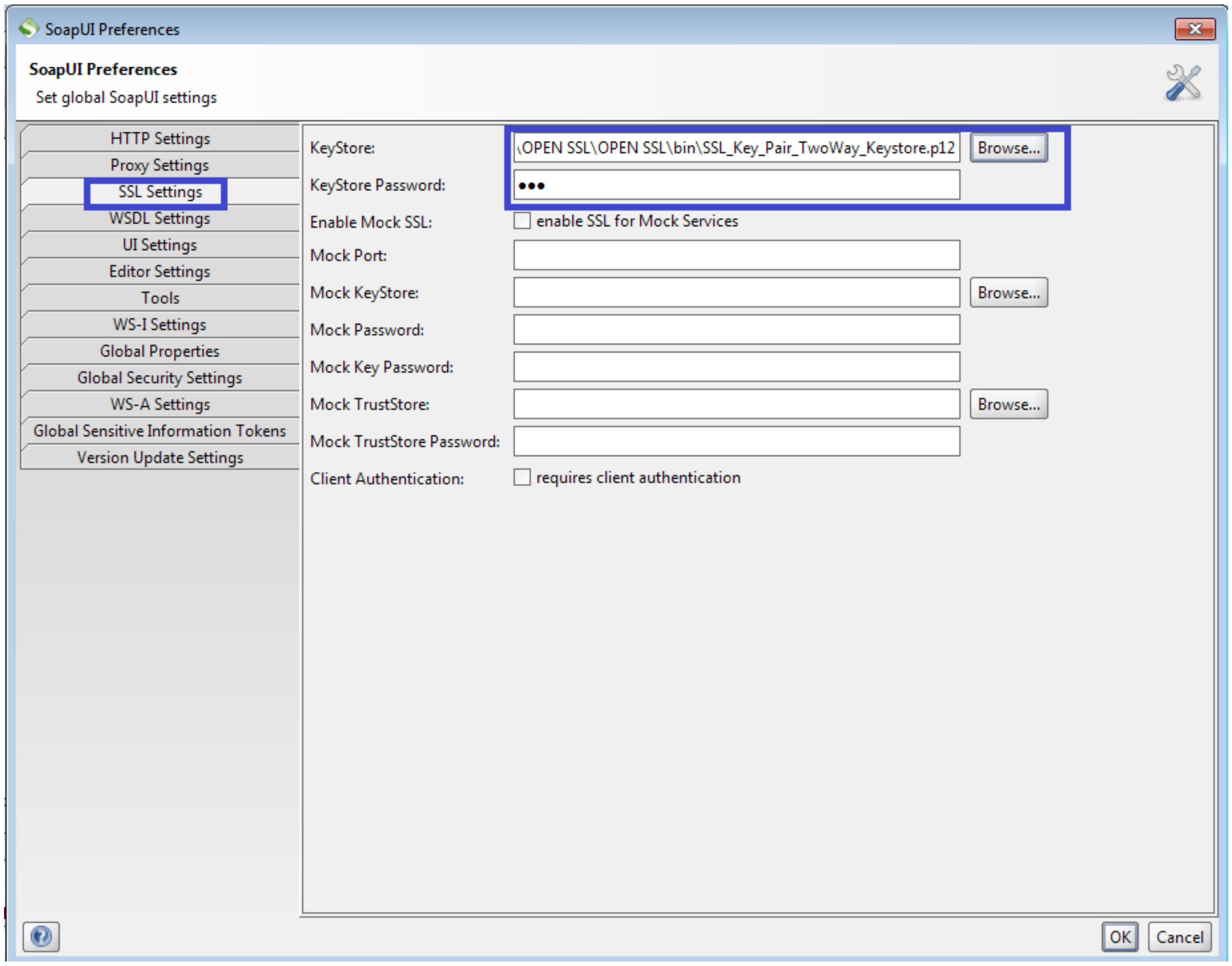
Crypto Identification Credentials

Name	SSL_TwoWay_Credentials *
Administrative State	[✓ enabled] disabled
Crypto Key	SSL_TwoWay_Key + ... *
Certificate	SSL_TwoWay_Certificate + ... *
Intermediate CA Certificate	(empty) add + ...

Click on + symbol to **configure Crypto Certificate.**

Adding Keystore to SoapUI

Open **SOAPUI**


SoapUI Preferences

SoapUI Preferences

Set global SoapUI settings

KeyStore:	\OPEN SSL\OPEN SSL\bin\SSL_Key_Pair_TwoWay_Keystore.p12
KeyStore Password:	•••
Enable Mock SSL:	☐ enable SSL for Mock Services
Mock Port:	
Mock KeyStore:	
Mock Password:	
Mock Key Password:	
Mock TrustStore:	
Mock TrustStore Password:	
Client Authentication:	☐ requires client authentication

Request in SoapUI

```
<?xml version="1.0" encoding="UTF-8"?>
<CATALOG>
<CD>
<TITLE>Empire Burlesque</TITLE>
<ARTIST>Bob Dylan</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>Columbia</COMPANY>
<PRICE>10.90</PRICE>
<YEAR>1985</YEAR>
</CD>
```

```
<CD>
<TITLE>Greatest Hits</TITLE>
<ARTIST>Dolly Parton</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>RCA</COMPANY>
<PRICE>9.90</PRICE>
<YEAR>1982</YEAR>
</CD>
<CD>
<TITLE>Still got the blues</TITLE>
<ARTIST>Gary Moore</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>Virgin records</COMPANY>
<PRICE>10.20</PRICE>
<YEAR>1990</YEAR>
</CD>
<CD>
<TITLE>Eros</TITLE>
<ARTIST>Eros Ramazzotti</ARTIST>
<COUNTRY>EU</COUNTRY>
<COMPANY>BMG</COMPANY>
<PRICE>9.90</PRICE>
<YEAR>1997</YEAR>
</CD>
<CD>
<TITLE>One night only</TITLE>
<ARTIST>Bee Gees</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>Polydor</COMPANY>
<PRICE>10.90</PRICE>
<YEAR>1998</YEAR>
</CD>
</CATALOG>
```

Response in SoapUI

```
<CATALOG>
<CD>
<TITLE>Empire Burlesque</TITLE>
<ARTIST>Bob Dylan</ARTIST>
```

```
<COUNTRY>USA</COUNTRY>
<COMPANY>Columbia</COMPANY>
<PRICE>10.90</PRICE>
<YEAR>1985</YEAR>
</CD>
<CD>
<TITLE>Greatest Hits</TITLE>
<ARTIST>Dolly Parton</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>RCA</COMPANY>
<PRICE>9.90</PRICE>
<YEAR>1982</YEAR>
</CD>
<CD>
<TITLE>Still got the blues</TITLE>
<ARTIST>Gary Moore</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>Virgin records</COMPANY>
<PRICE>10.20</PRICE>
<YEAR>1990</YEAR>
</CD>
<CD>
<TITLE>Eros</TITLE>
<ARTIST>Eros Ramazzotti</ARTIST>
<COUNTRY>EU</COUNTRY>
<COMPANY>BMG</COMPANY>
<PRICE>9.90</PRICE>
<YEAR>1997</YEAR>
</CD>
<CD>
<TITLE>One night only</TITLE>
<ARTIST>Bee Gees</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>Polydor</COMPANY>
<PRICE>10.90</PRICE>
<YEAR>1998</YEAR>
</CD>
</CATALOG>
```

Implementation of Oauth in CA API Gateway

Oauth:

- OAuth is a simple way to publish and interact with protected data.
- It's also a safer and more secure way for people to give you access.
- OAuth is used in a wide variety of applications, including providing mechanisms for user authentication.
- This has led many developers and API providers to incorrectly conclude that OAuth is itself an *authentication* protocol.
- OAuth 2.0 does not support signature, encryption, channel binding, or client verification.
- It relies completely on TLS for some degree of confidentiality and server authentication.
- Because OAuth 2.0 is more of a framework than a defined protocol, one OAuth 2.0 implementation is less likely to be naturally interoperable with another OAuth 2.0 implementation. Further deployment profiling and specification is required for any interoperability.
- This mechanism is used, for example, by Google, Facebook, Microsoft, Twitter, etc to permit the users to share information about their accounts with third party applications or websites.

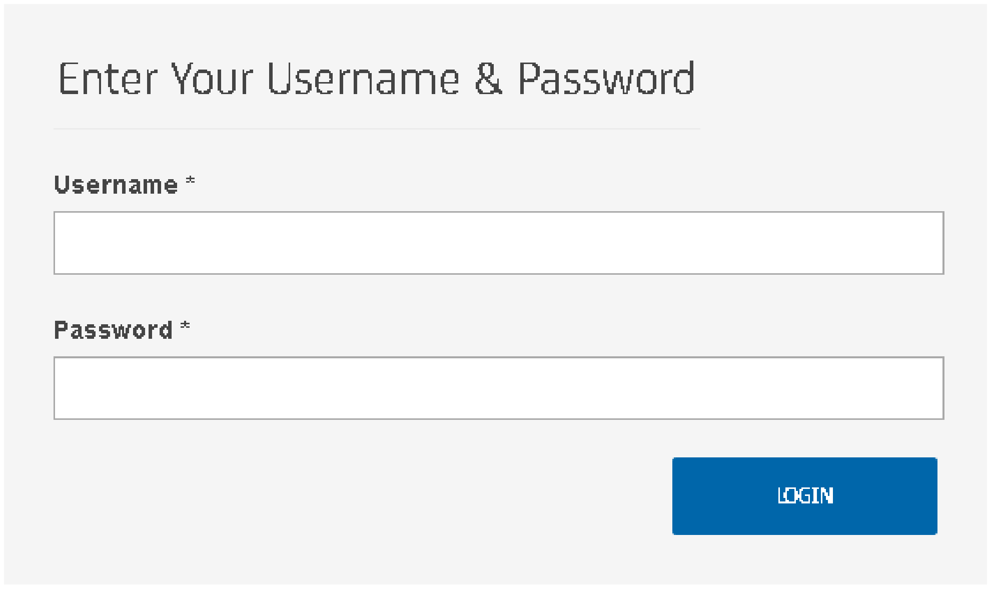
Run the OAuth 2.0 Test Client

The test client is used to verify installation changes and to access OAuth 2.0-secured API endpoints of platforms.

Run the Oauth Manager:

Navigate to the following URL in a browser:

- **https://< Gateway_host >:8443/oauth/manager**



Login to oAuth Manager and click on clients to get an existed clients and follow the follwing steps.

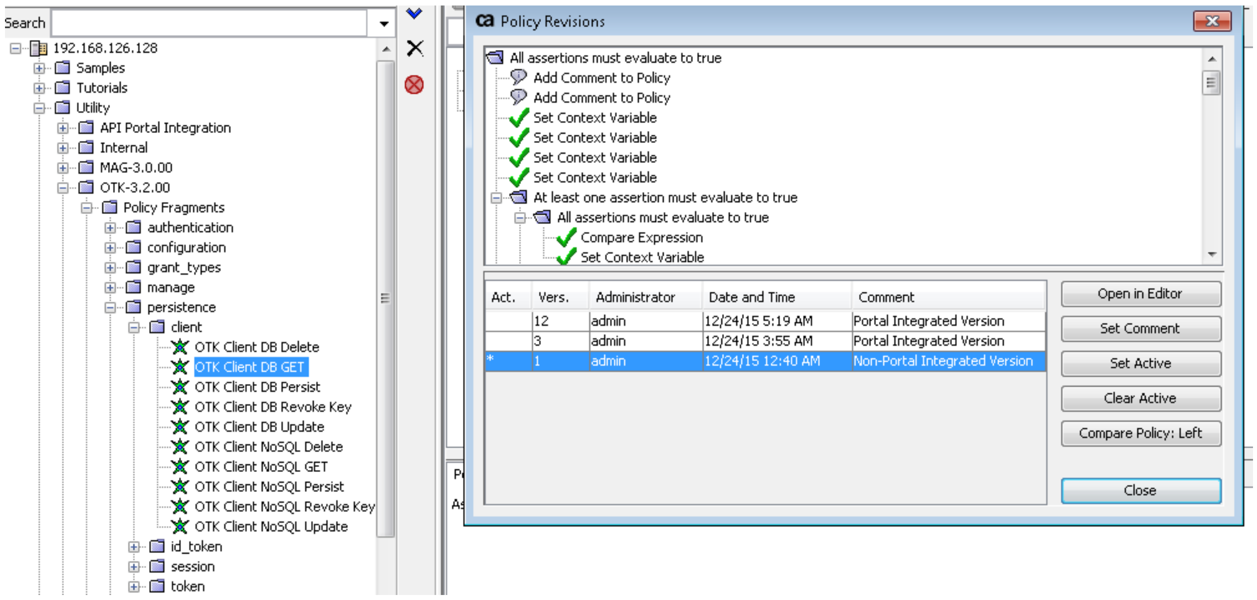
Step 1:

Login to CA API Gateway-Policy Manager 9.0.00

Step 2:

Go to your Gateway IP address, click on Utility->OTK-3.2.00->Policy fragments->persistence->clients->OTK CLIENT GET DB.

Right click on OTK CLIENT GET DB , select revision History and click on version 1 and then click on SET ACTIVE.

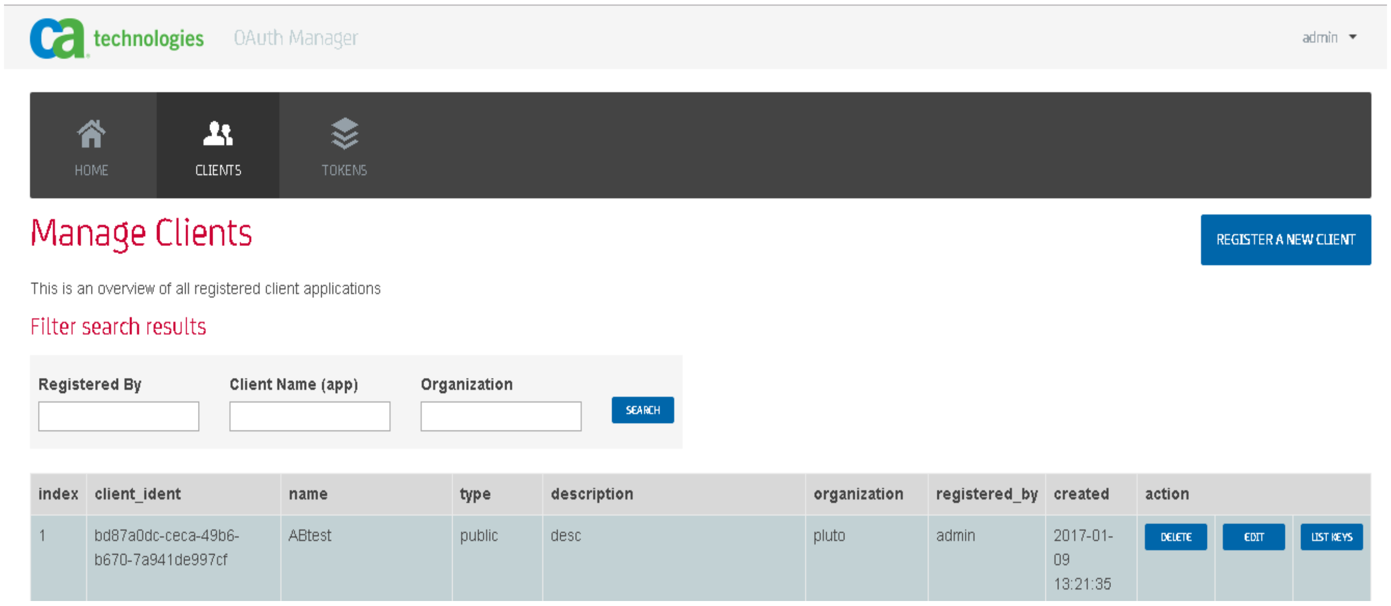


- 192.168.126.128
  - Samples
  - Tutorials
  - Utility
    - API Portal Integration
    - Internal
    - MAG-3.0.00
    - OTK-3.2.00
      - Policy Fragments
        - authentication
        - configuration
        - grant_types
        - manage
        - persistence
          - client
            - OTK Client DB Delete
            - OTK Client DB GET
            - OTK Client DB Persist
            - OTK Client DB Revoke Key
            - OTK Client DB Update
            - OTK Client NoSQL Delete
            - OTK Client NoSQL GET
            - OTK Client NoSQL Persist
            - OTK Client NoSQL Revoke Key
            - OTK Client NoSQL Update
          - id_token
          - session
          - token

- All assertions must evaluate to true
  - Add Comment to Policy
  - Add Comment to Policy
  - Set Context Variable
  - Set Context Variable
  - Set Context Variable
  - Set Context Variable
  - At least one assertion must evaluate to true
    - All assertions must evaluate to true
      - Compare Expression
      - Set Context Variable

Act.	Vers.	Administrator	Date and Time	Comment
	12	admin	12/24/15 5:19 AM	Portal Integrated Version
	3	admin	12/24/15 3:55 AM	Portal Integrated Version
*	1	admin	12/24/15 12:40 AM	Non-Portal Integrated Version

After clicking on SET ACTIVE , all the existed client information will be shown.



Manage Clients

This is an overview of all registered client applications

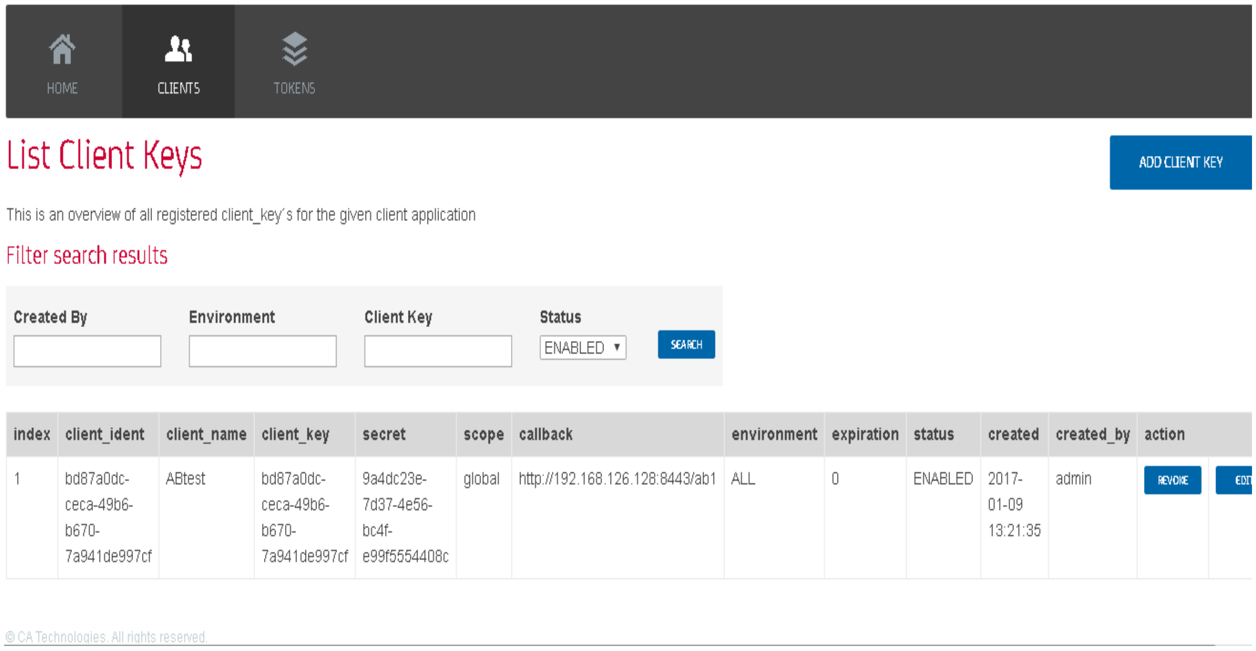
Filter search results

Registered By	Client Name (app)	Organization

index	client_ident	name	type	description	organization	registered_by	created	action
1	bd87a0dc-ceca-49b6-b670-7a941de997cf	ABtest	public	desc	pluto	admin	2017-01-09 13:21:35	DELETE EDIT LIST KEYS

If you want to create new client, click on REGISTER NEW CLIENT and fill the details.

Click on LIST KEYS to check your client identifier, client secret and key.



List Client Keys

This is an overview of all registered client_key´s for the given client application

Filter search results

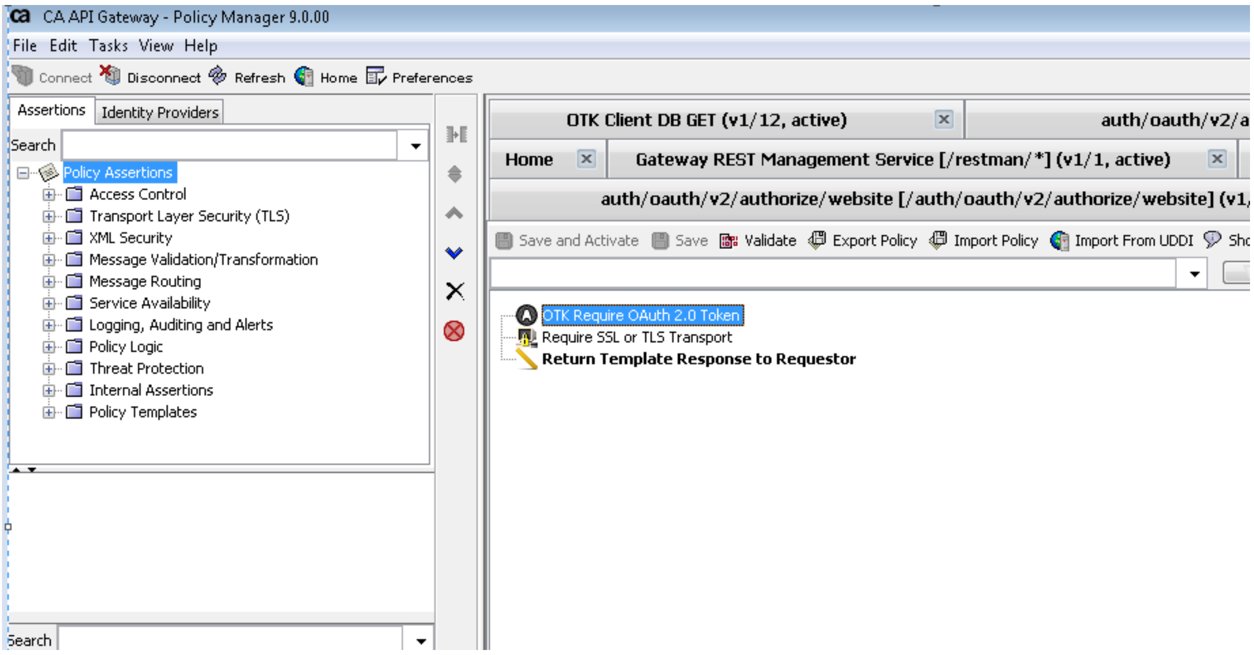
Created By	Environment	Client Key	Status
			ENABLED

index	client_ident	client_name	client_key	secret	scope	callback	environment	expiration	status	created	created_by	action
1	bd87a0dc-ceca-49b6-b670-7a941de997cf	ABtest	bd87a0dc-ceca-49b6-b670-7a941de997cf	9a4dc23e-7d37-4e56-bc4f-e99f5554408c	global	http://192.168.126.128:8443/ab1	ALL	0	ENABLED	2017-01-09 13:21:35	admin	REVOKE EDIT

© CA Technologies. All rights reserved.

To create our own API, login to CA API Gateway-policy manager and right click on the gateway IP address.

Select publish web API and fill the details(name and alias name).


CA API Gateway - Policy Manager 9.0.00
File Edit Tasks View Help
Connect Disconnect Refresh Home Preferences
Assertions Identity Providers
Search
Policy Assertions
Access Control
Transport Layer Security (TLS)
XML Security
Message Validation/Transformation
Message Routing
Service Availability
Logging, Auditing and Alerts
Policy Logic
Threat Protection
Internal Assertions
Policy Templates
OTK Client DB GET (v1/12, active)
Home
Gateway REST Management Service [/restman/*] (v1/1, active)
auth/oauth/v2/authorize/website [/auth/oauth/v2/authorize/website] (v1,
Save and Activate Save Validate Export Policy Import Policy Import From UDDI
OTK Require OAuth 2.0 Token
Require SSL or TLS Transport
Return Template Response to Requestor

Here we used three assertions for the clients.

OTK Require Oauth 2.0 Token: This is the OTK toolkit for Oauth 2.0

Require SSL or TLS Transport： The *Require SSL or TLS Transport /Require SSL or TLS Transport with Client Authentication* assertion allows you to specify the SSL or TLS requirement to ensure transport-level confidentiality and integrity. You can specify whether an SSL/TLS connection is required, optional, or forbidden.

Return Template Response to Requestor Assertion:
The *Return Template Response to Requestor* assertion lets you define a message to be returned to the requestor. This allows you (for example) to create a more descriptive message for a SOAP fault or to elaborate an error condition to aid troubleshooting. For example, you place this assertion in an "At least one assertion must evaluate to true"assertion folder after an Evaluate Response XPath Assertion. If the Evaluate Response Xpath

assertion fails, then the template response message is sent back to the requestor.

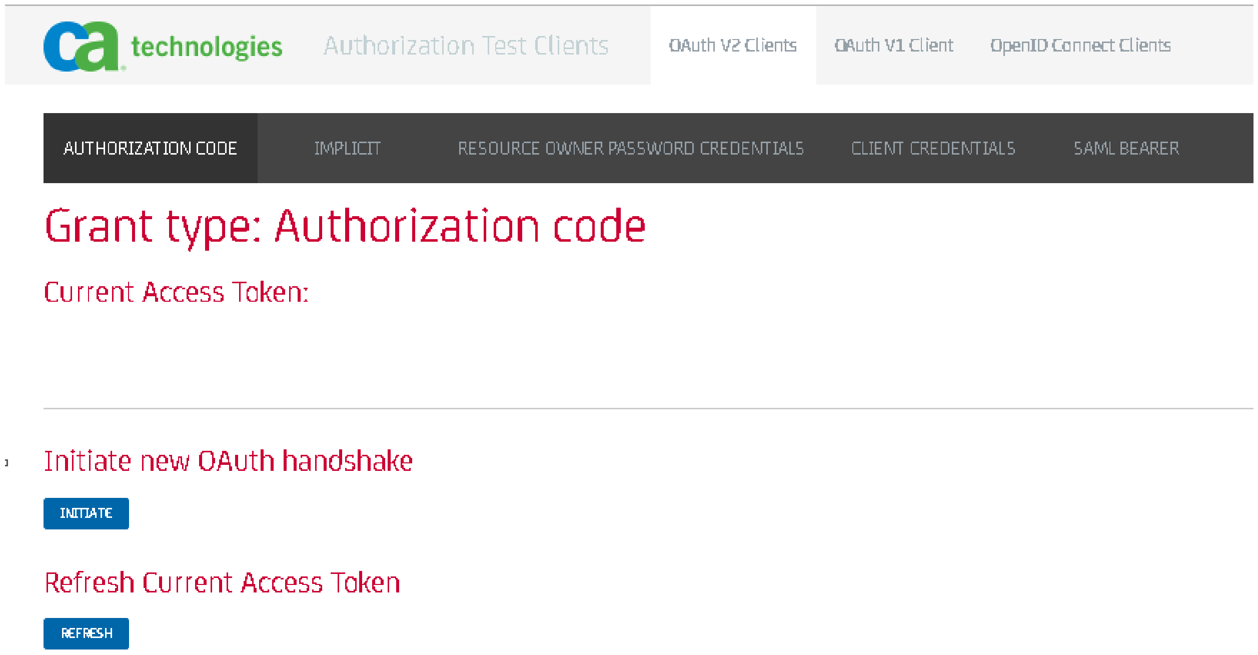
Run the Client:

To run the OAuth 2.0 test client**:**

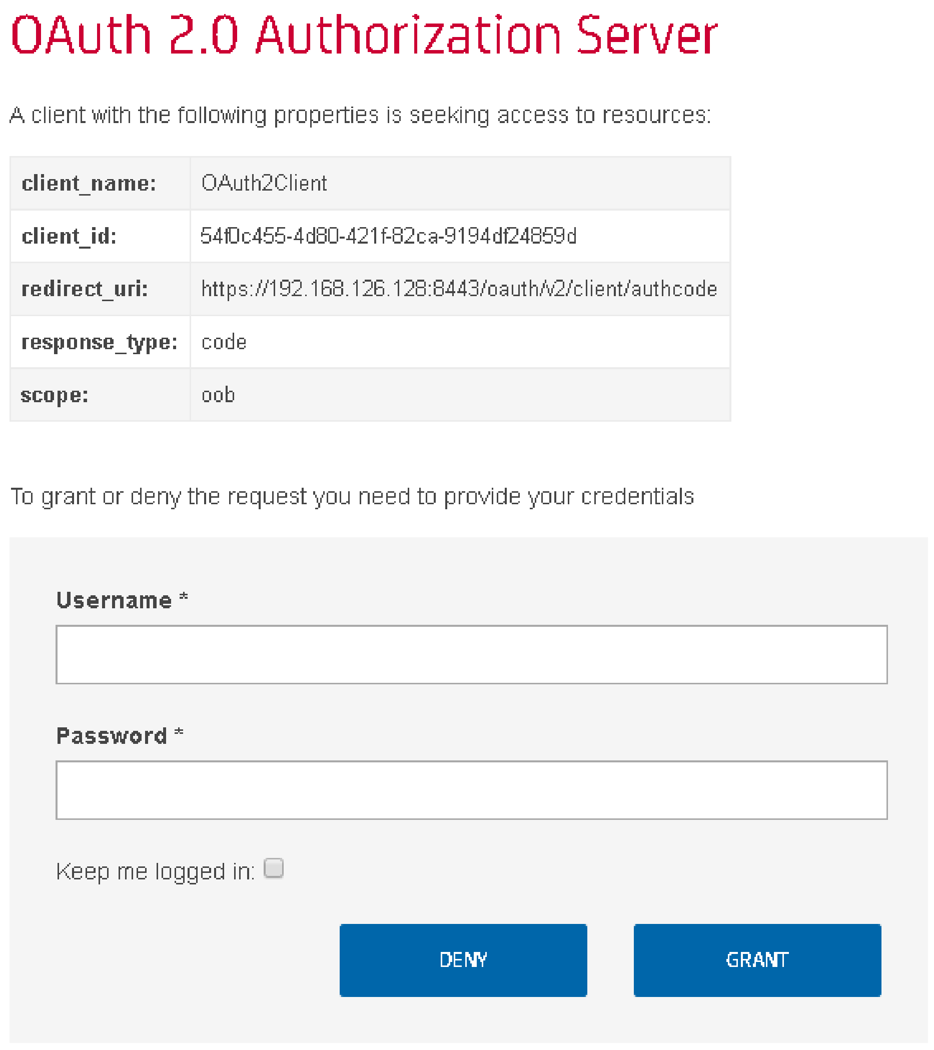
Navigate to the following URL in a browser:

https://<Gateway_host>:8443/oauth/v2/client/authcode

The OAuth Client Test Application screen is displayed.



To get an access token, click on initiate.



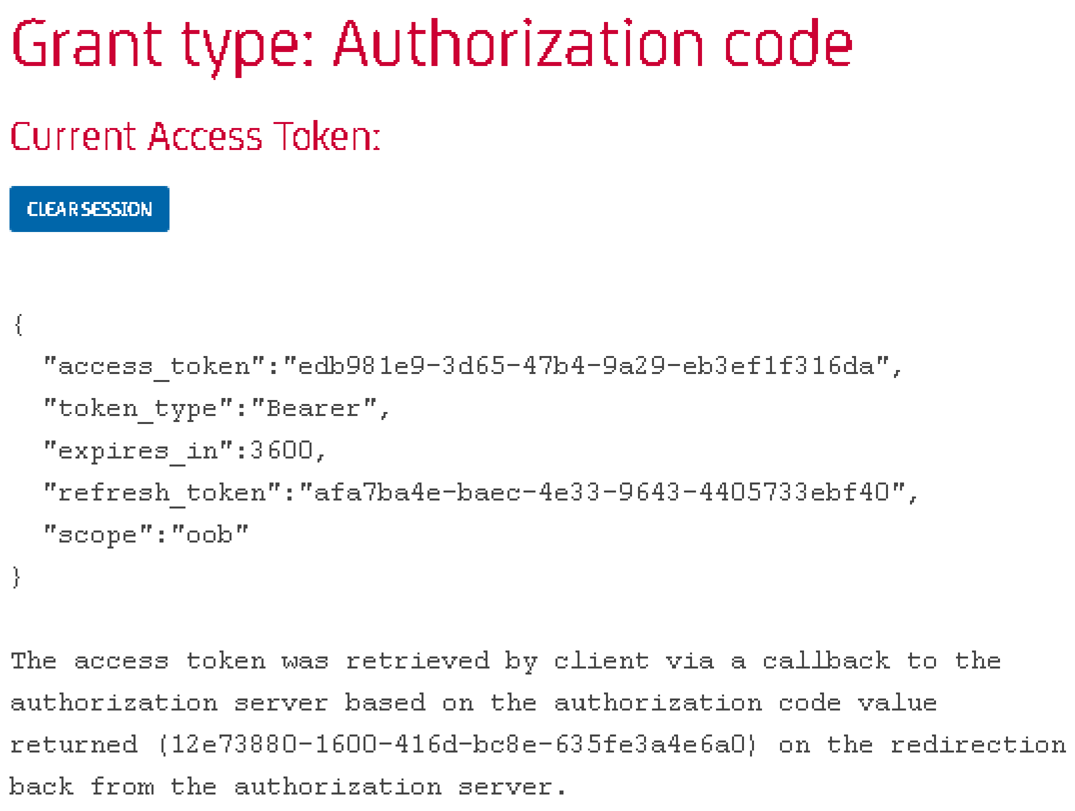
OAuth 2.0 Authorization Server

A client with the following properties is seeking access to resources:

client_name:	OAuth2Client
client_id:	54f0c455-4d80-421f-82ca-9194df24859d
redirect_uri:	https://192.168.126.128:8443/oauth/v2/client/authcode
response_type:	code
scope:	oob

To grant or deny the request you need to provide your credentials

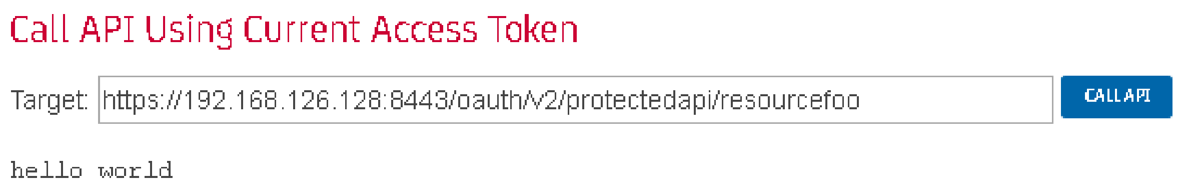
Fill with your credentials, then click in GRANT.



```
{
  "access_token":"edb981e9-3d65-47b4-9a29-eb3ef1f316da",
  "token_type":"Bearer",
  "expires_in":3600,
  "refresh_token":"afa7ba4e-baec-4e33-9643-4405733ebf40",
  "scope":"oob"
}
```

```
The access token was retrieved by client via a callback to the
authorization server based on the authorization code value
returned (12e73880-1600-416d-bc8e-635fe3a4e6a0) on the redirection
back from the authorization server.
```

Give your requested API in the cal API and get the data.



Call API Using Current Access Token

Target: https://192.168.126.128:8443/oauth/v2/protectedapi/resourcefoo CALL API

```
hello world
```