innovar healthcare

SSL Plugin v3 Settings

Created: July 24, 2025
Updated: 8/25/2025

Introduction

This plugin enables the use of SSL/TLS encryption across various connectors within BridgeLink. Specifically, it supports secure configurations for the HTTP Listener, HTTP Sender, TCP Listener, TCP Sender, Web Service Listener, and Web Service Sender. It is designed to help ensure secure data exchange across both HTTP and TCP-based interfaces, as well as web service endpoints.

Why SSL?

Encrypting your traffic is essential for transmitting data, especially patient health information. Using the SSL Settings Plugin allows you to enable SSL/TLS both when receiving and sending messages.

Key Features

- **Keystore**
  - Specify the path to your keystore (file system or S3).
  - Contains your certificate (for identity) and private key (for decrypting messages).
- **Truststore**
  - Specify the path to the truststore.
  - Stores certificates of all trusted systems.
  - Required for mutual TLS authentication.
- **Verify Hostname**
  - Option to enable or disable hostname verification when using SSL.
  - Validates that the server's certificate matches the URL.

Getting Started

Before you dive into the documentation, make sure to review the installation prerequisites and check compatibility with your existing BridgeLink setup. The subsequent sections will guide you through the configuration, usage, and optimization of the SSL Settings plugin.

Installation

If you are subscribed to the Open Source Mirth® Connect package from Innovar Healthcare on the AWS Marketplace, the extension should be pre-installed in the 'Advanced with SSL', 'Advanced with SSL Autoscaling'.

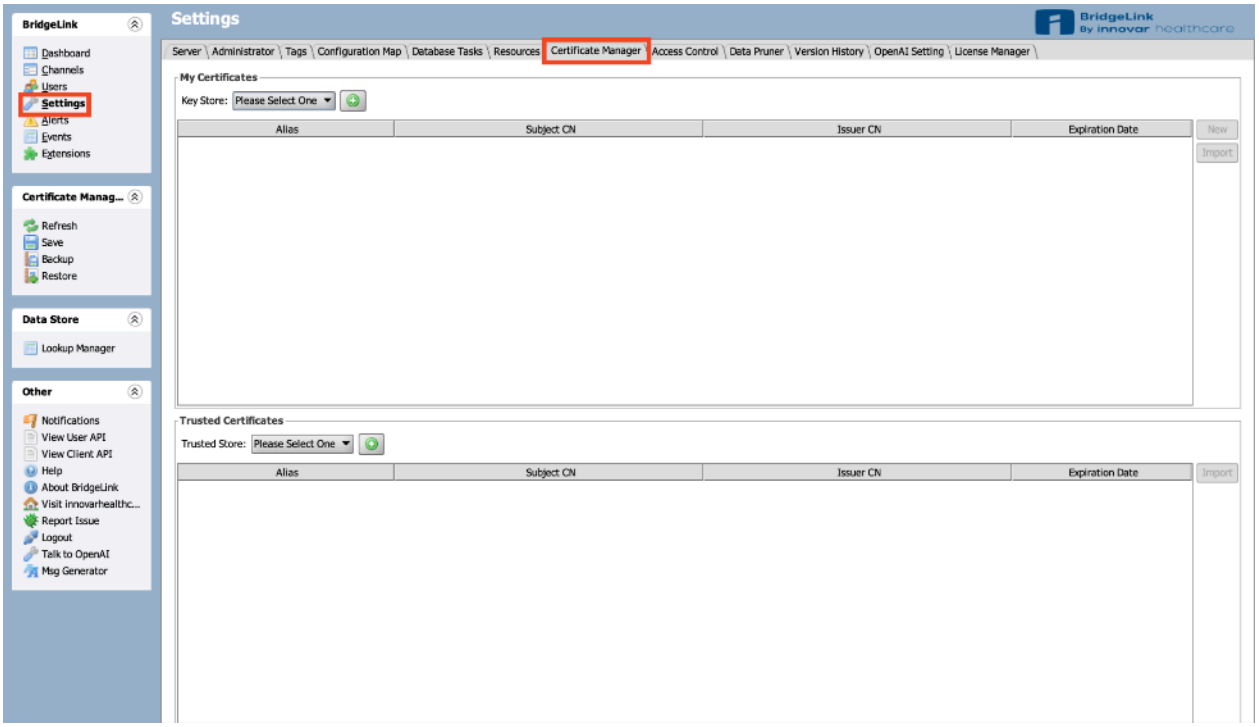
If you are subscribed to the BridgeLink package from Innovar Healthcare on the AWS Marketplace, the extension should be pre-installed in the ‘BridgeLink Standard Edition an Open Source Mirth Connect Fork’, ‘BridgeLink Enterprise Edition an Open Source Mirth Connect Fork versions’.

If you need to reinstall or update the plugin, you can do so from the BridgeLink application:

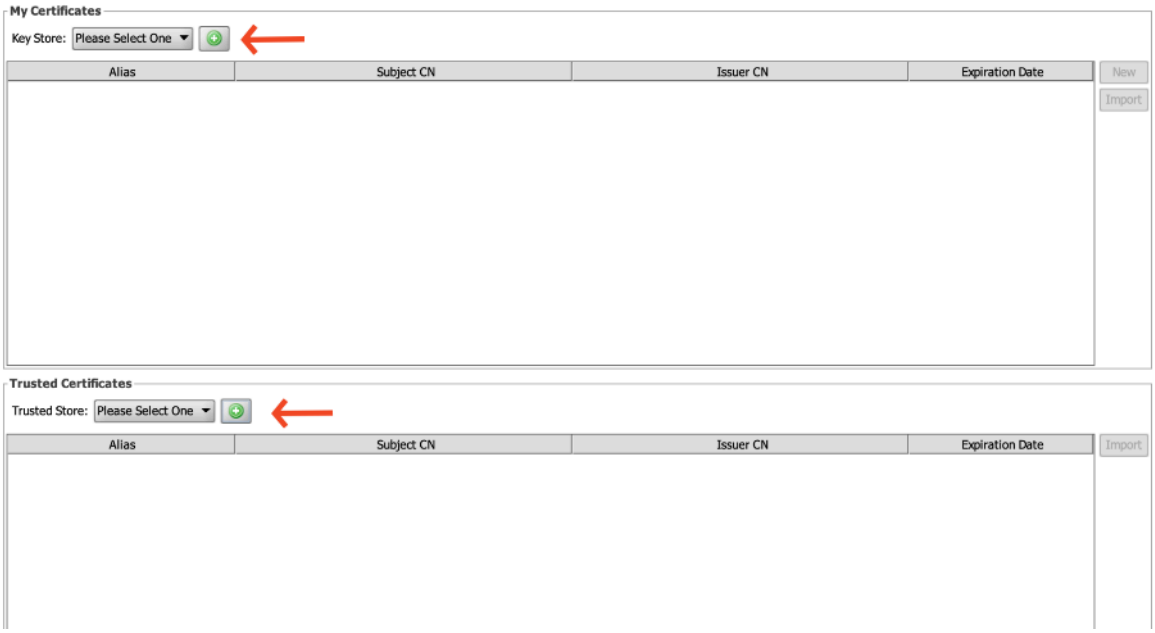
1. Log into BridgeLink.
2. Click on **Extensions**.
3. At the bottom of the screen, click **Browse**.
4. In the pop-up window, locate and select the plugin ZIP file on your local machine.
5. Click **Open** to return to the Extensions screen, where the file path will be populated.
6. Click **Install** to upload the file.
7. Restart the BridgeLink service to complete the installation.

Plugin Configuration

The settings include a 'Certificate Manager' tab, where you can configure both the keystore and truststore certificates. These settings take effect when configuring SSL in a channel. You can either manually enter the information or select 'From System.' The Certificate Manager is where the information is configured for the 'From System' option.

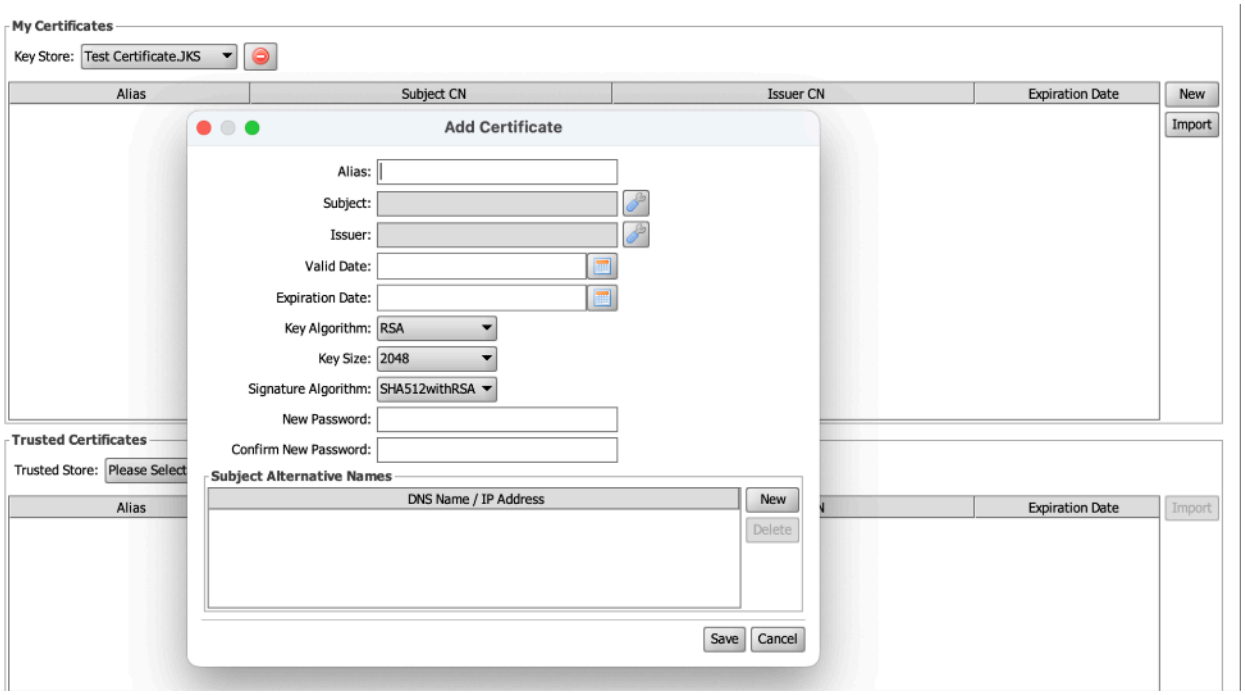


To add a new key store, click the green button.

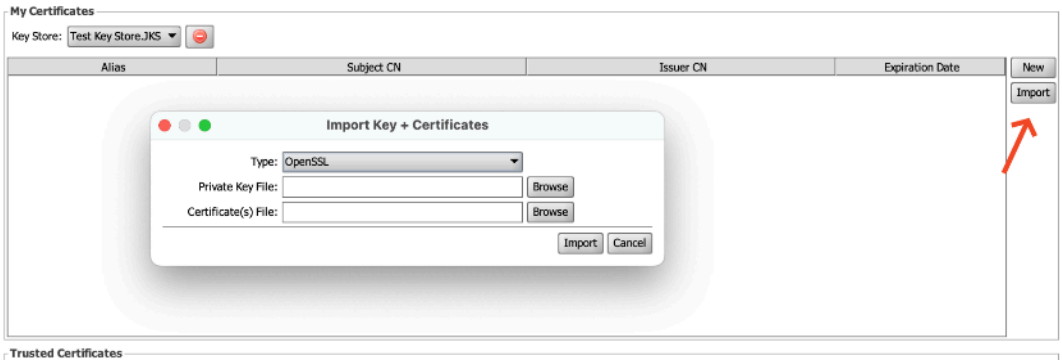


Enter the keystore name, password, and confirm the password.

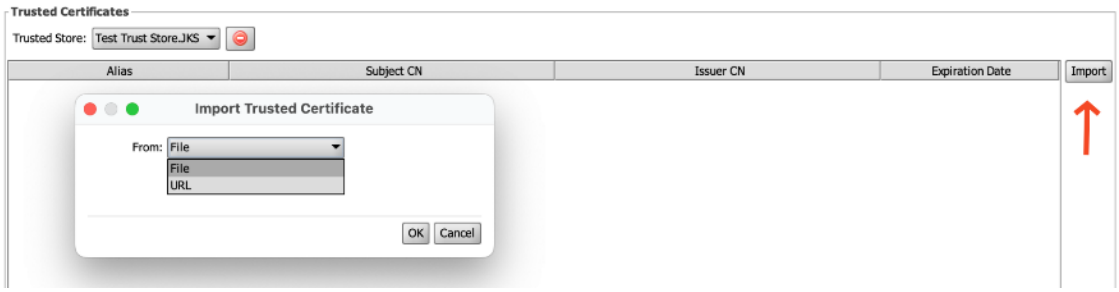
Manually add a key and certificate. Click the 'New' button and enter the required information.

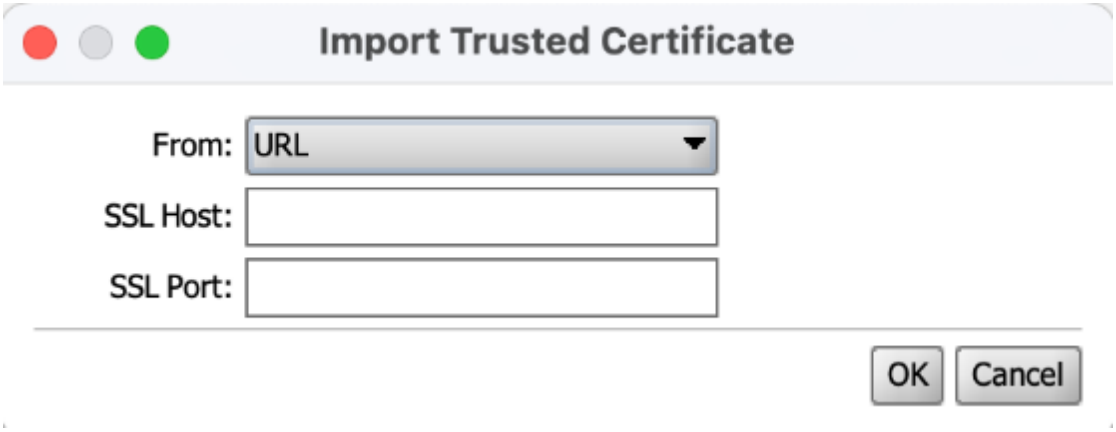


There is also an option to import a key & certificate. Import the private key file and certificate(s) file.

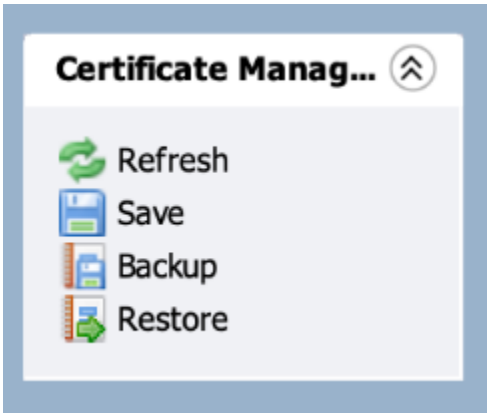


Under the truststore, there is also an option to import a key and certificate.

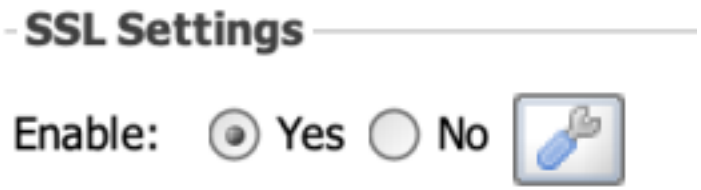




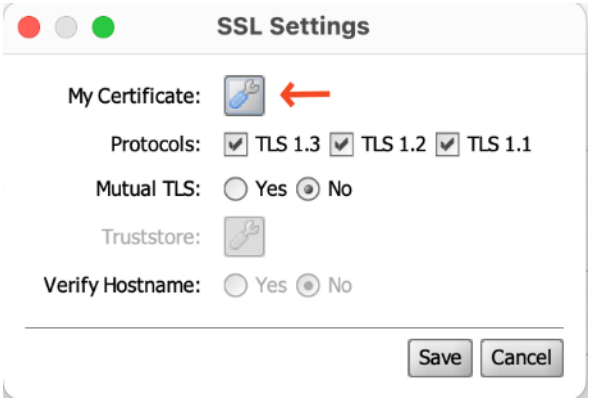
There are options to refresh, save, backup, and restore.



With the SSL Settings Plugin installed, there is an option to enable/disable SSL on HTTP Connector and Web Service Sender. With SSL enabled, the tool button is enabled allowing you to configure the SSL Connection.

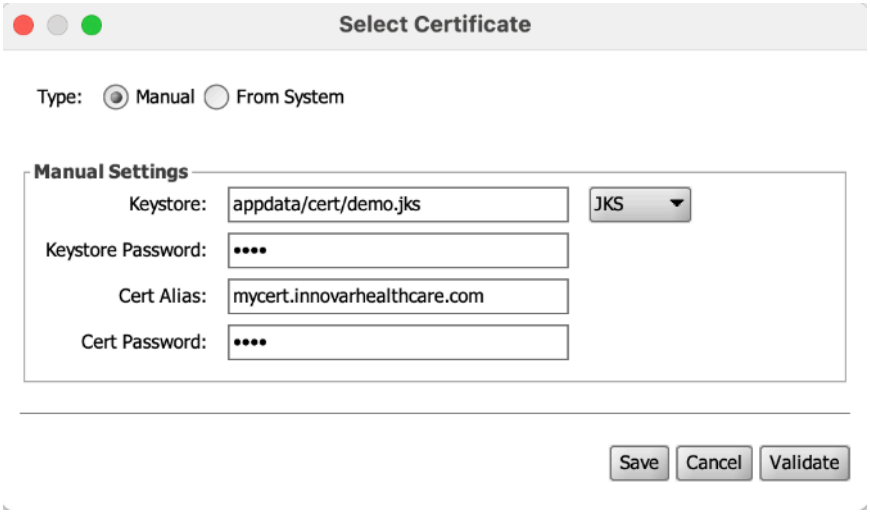


Clicking the tool icon navigates to the section where you can enter the certificate information.



There are two types: Manual and From System. The 'From System' option retrieves its information from the configuration in the settings under Certificate Manager.

Manual:



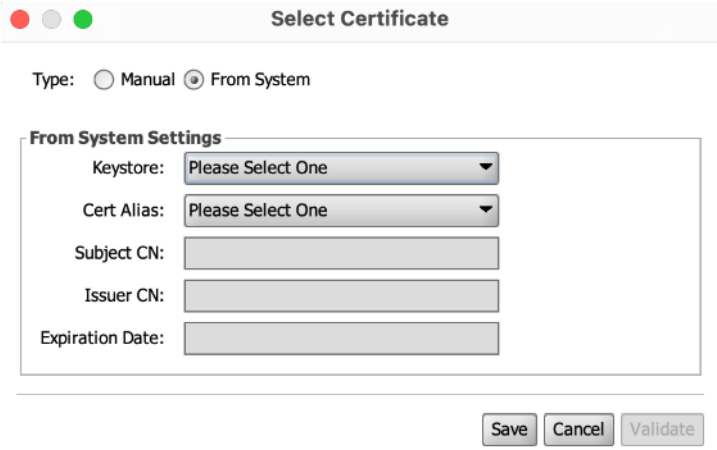
The **Keystore** field is for specifying the path to the keystore. It can be a full, or relative path to the BridgeLink installation directory. You can also specify a S3 path using the following format: s3://<region>.<bucketname>/<objectkey>. Make sure to select the appropriate keystore type in the dropdown field.

In the **Keystore Password** field, enter the password for the keystore.

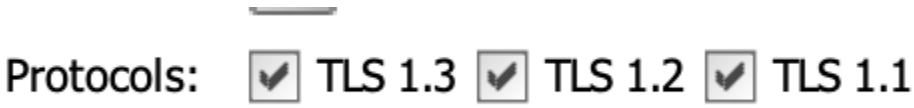
In the **Cert Alias and Cert Password** fields, enter the certificate alias and the password for the associated private key.

From System:

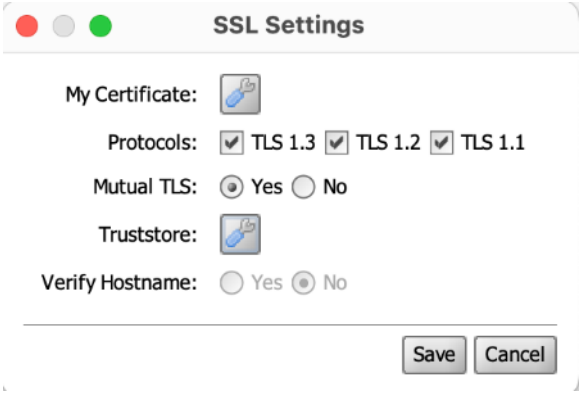
If 'From System' is selected, the information will be pulled from the Certificate Manager in settings.



In **Protocols**, you can select which version of TLS you want to support.



When **mutual TLS (mTLS)** is enabled, there is a tool icon to configure the truststore

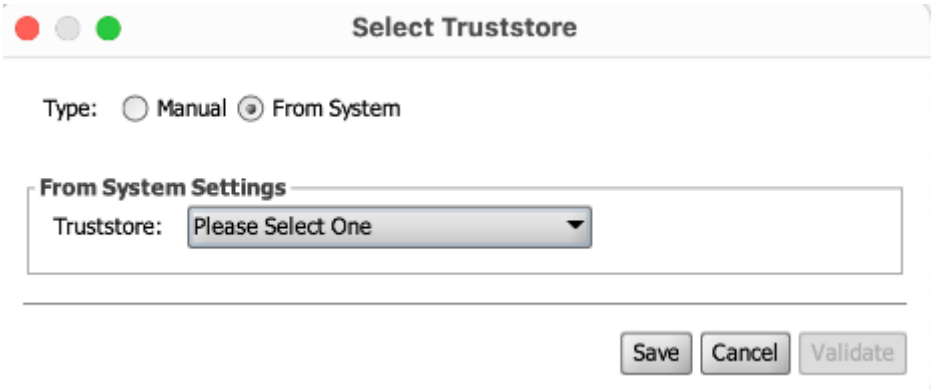


Similar to the certificate, there are two types when configuring the truststore: Manual and From System.

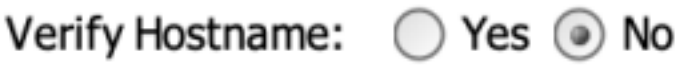
Manual:

Manually configure the truststore and set the truststore password. You can use a relative or absolute file path, or a S3 location using format: s://<region>.<bucketname>/<objectkey>

From System:



When using SSL with the HTTP Listener, HTTP Sender, and the Web Service Sender, you can enable Verify Hostname to validate that the certificate alias matches the URL.



SSL v3 Addendum: Client API

Product: Innovar SSL Plugin v3
Scope for this release: JKS import and export for keystores and truststores.

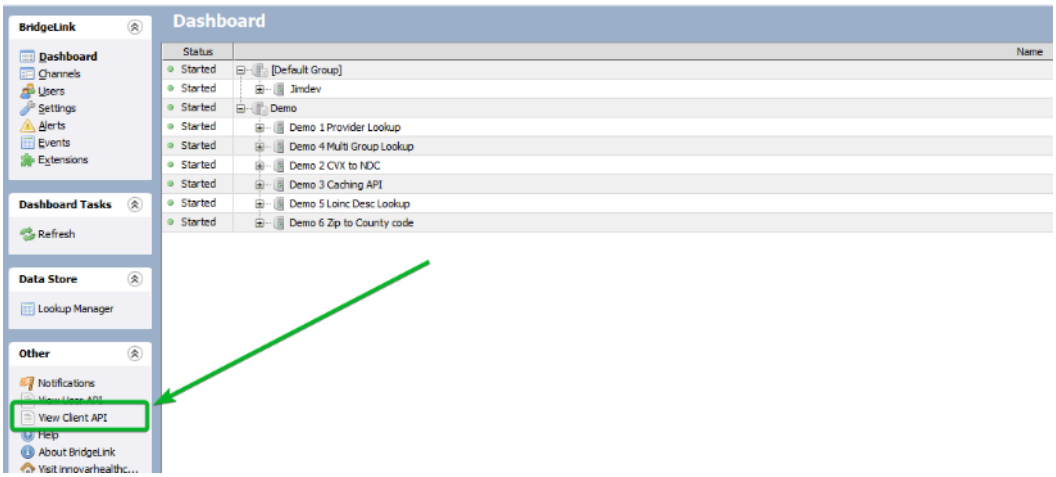
What's new in v3

- Client API to manage **keystores** and **truststores**, including per-certificate actions.
- **Full certificate chain** visibility for keystore entries and a chain export action.
- **JKS** keystore and truststore **import** and **export** from Certificate Manager.

Client API location

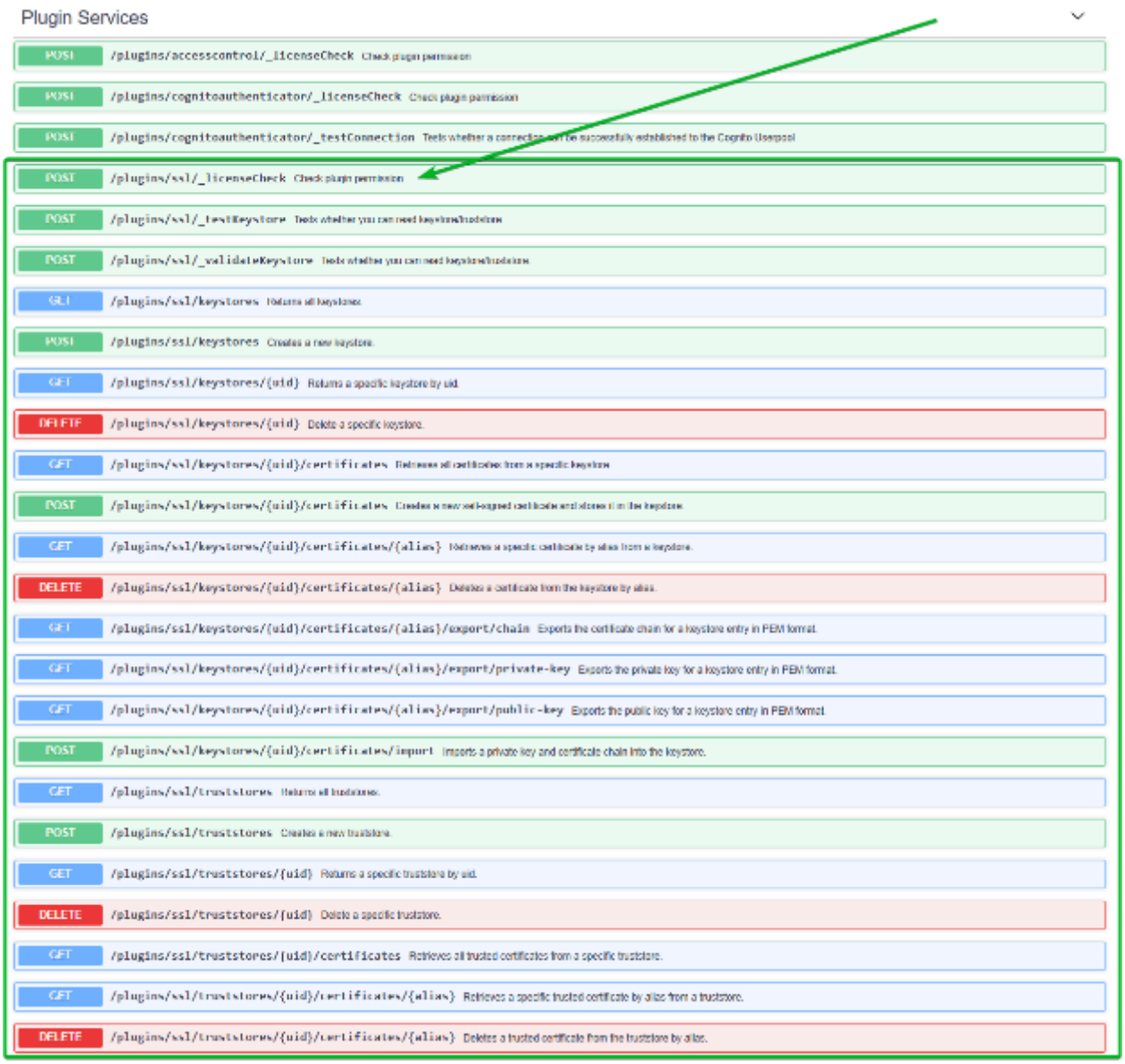
Use this section only to reference live endpoints and schemas.

- Open the Administrator and navigate to **View Client API**.



- Select **Plugin Services**.

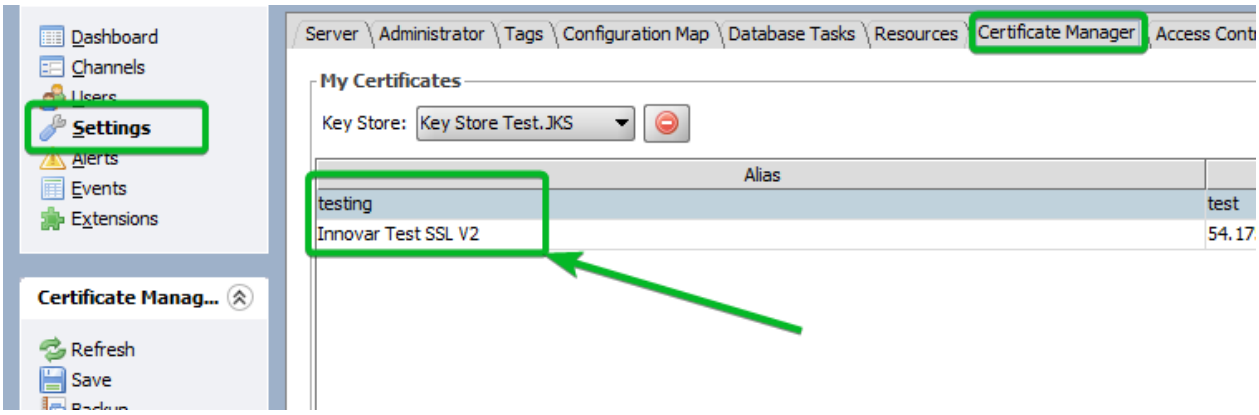
- Expand **ssl** to see the available endpoints.



Certificate Manager overview

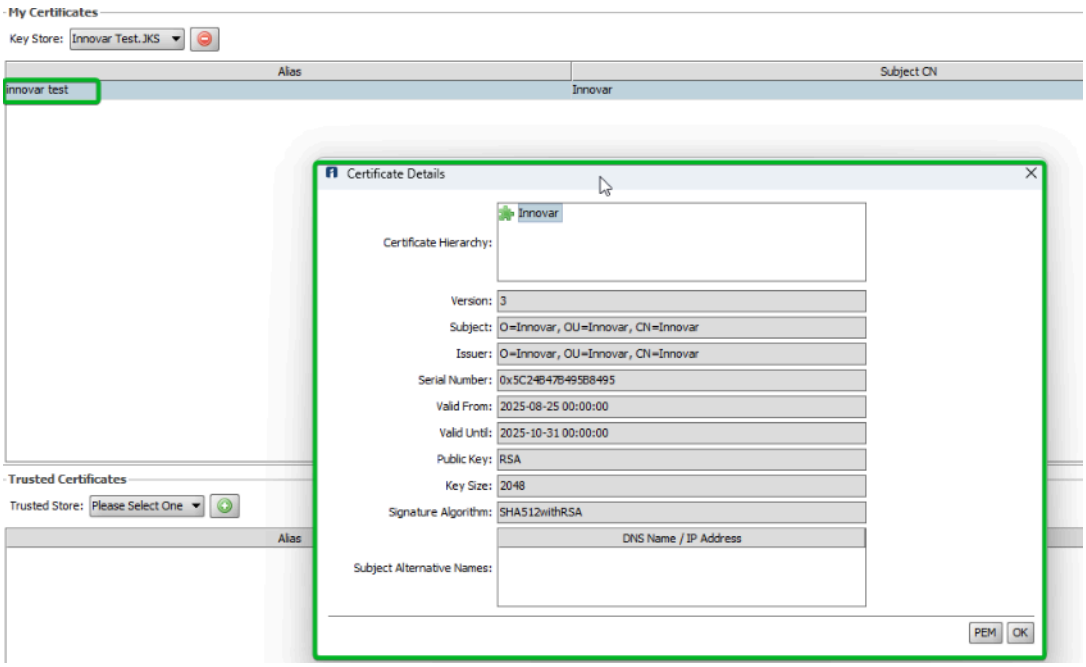
UI path: **Settings** → **Certificate Manager**

- The selector switches between **Key Store** and **Trust Store**.
- The table lists entries by **Alias**.
- Right-click an alias for available actions.



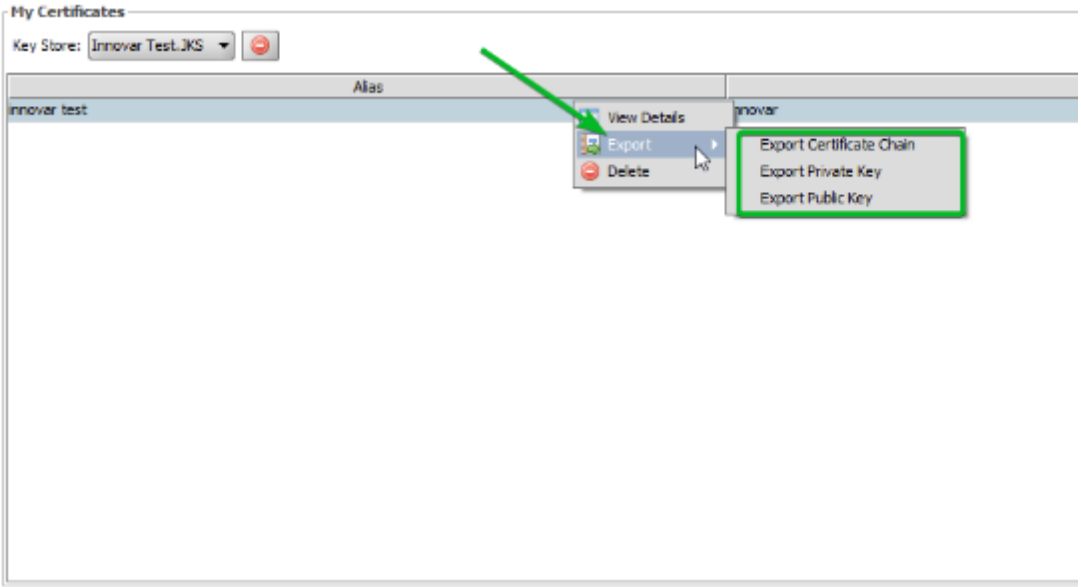
Viewing the Certificate Details

- In **Certificate Manager**, select **Key Store** and highlight the desired alias.
- Right-click and choose **View Details** to see the leaf, intermediate certificates, and root.



Exporting from a keystore entry

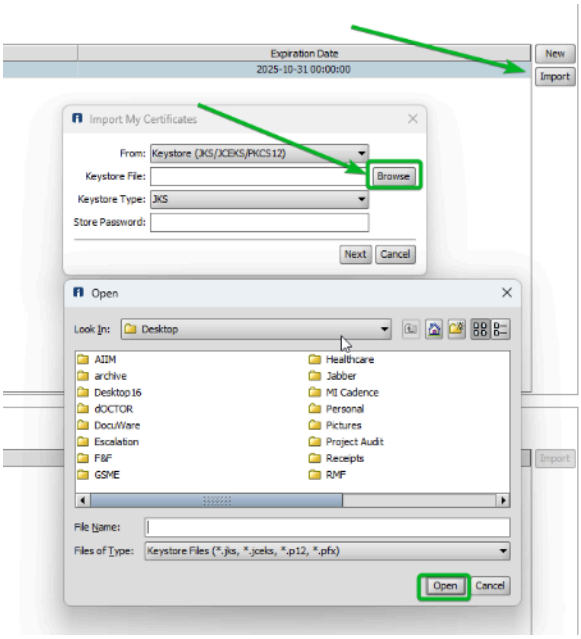
- Right-click an alias and select **Export**, then choose one of the available options:
  - **Export Certificate Chain** (PEM)
  - **Export Private Key** (PEM)
  - **Export Public Key** (PEM)



Security note: Store private keys in a secure vault and restrict access.

Importing an existing keystore file

- In **Certificate Manager**, select **Import**.
- Choose **From: Keystore (JKS/JCEKS/PKCS12)**.
- Select the **Keystore File**.
- Set **Keystore Type** to **JKS** for this release.
- Enter the **Store Password** and continue.



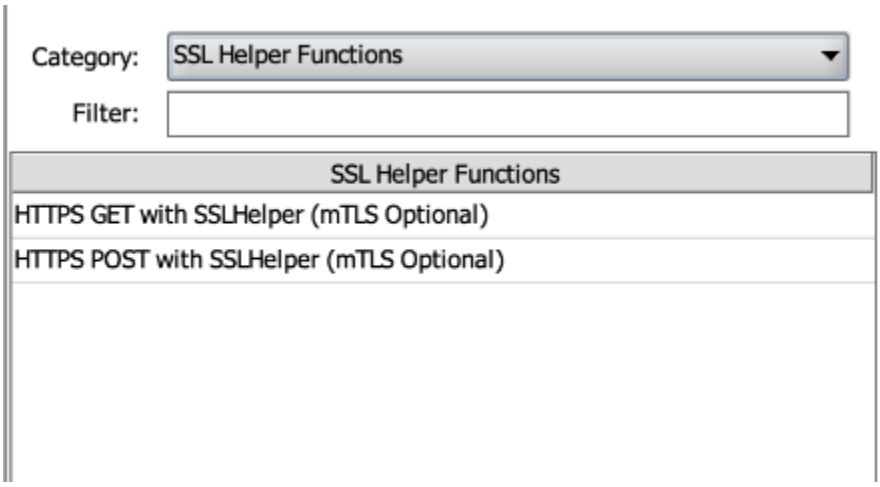
Entries from the JKS appear as aliases in the selected store. You can use **View Details** and **Export** on those entries.

Using keystores and truststores in channels

- In a connector's SSL settings, select the keystore and truststore you configured in Certificate Manager.
- If the connector requires a client certificate, choose the correct **Alias**.
- When hostname verification is applicable, enable it and ensure the certificate CN or SAN contains the target hostname.

SSL Helper Functions

Two Templates that let a channel script make outbound HTTPS requests using your truststore. The templates use is named **SSLHelper**.



Insert the template into a script step

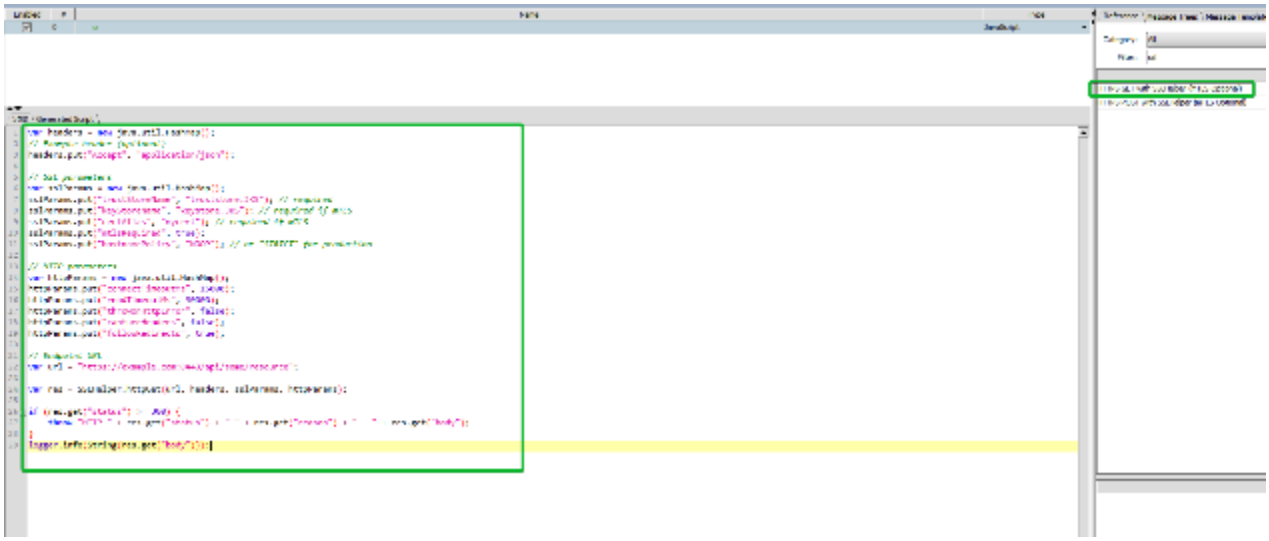
- In the Transformer, click **Add New Step** and choose **JavaScript**.
- The template's code appears in the **Step** editor.

Configure the template

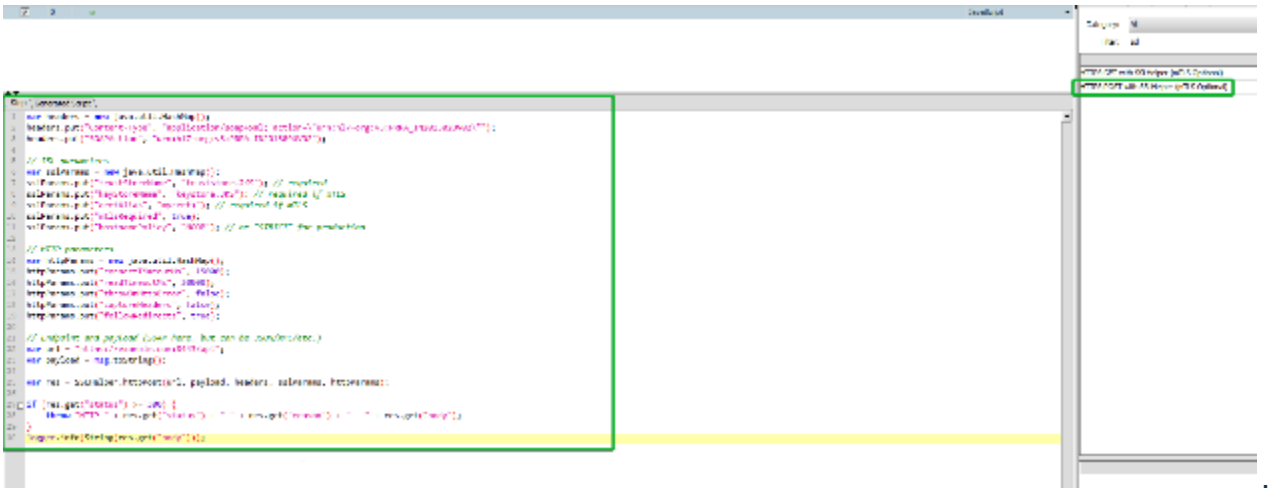
Change only the obvious fields after insertion:

- **Endpoint**: set the HTTPS URL.
- **sslParams**
  - trustStoreName — required
  - keyStoreName — required if mTLS
  - certAlias — required if mTLS
  - mTlsRequired — include when doing mTLS
  - hostnamePolicy — template comments show "NOOP" or "STRICT"
- **httpParams (names shown in the template)**
  - connectTimeoutMs, readTimeoutMs, throwOnHttpError, captureHeaders, followRedirects
- **Headers**: keep the example SOAP/JSON headers or replace them with your target's requirements.

HTTPS GET with SSLHelper



HTTPS POST with SSLHelper



Utilities

Method	Path	Purpose
POST	/plugins/ssl/_testKeystore	Test whether a keystore or truststore can be read.
POST	/plugins/ssl/_validateKeystore	Validate a keystore or truststore and its access details.

Keystores

Method	Path	Purpose
GET	/plugins/ssl/keystores	List keystores.

Method	Path	Purpose
POST	/plugins/ssl/keystores	Create a keystore entry.
GET	/plugins/ssl/keystores/{uid}	Get a specific keystore.
DELETE	/plugins/ssl/keystores/{uid}	Delete a keystore.
GET	/plugins/ssl/keystores/{uid}/certificates	List certificates in a keystore.
POST	/plugins/ssl/keystores/{uid}/certificates	Create a new self-signed certificate in the keystore.
GET	/plugins/ssl/keystores/{uid}/certificates/{alias}	Get a certificate by alias.
DELETE	/plugins/ssl/keystores/{uid}/certificates/{alias}	Delete a certificate by alias.
GET	/plugins/ssl/keystores/{uid}/certificates/{alias}/export/chain	Export the full certificate chain in PEM.
GET	/plugins/ssl/keystores/{uid}/certificates/{alias}/export/private-key	Export the private key in PEM.
GET	/plugins/ssl/keystores/{uid}/certificates/{alias}/export/public-key	Export the public key in PEM.
POST	/plugins/ssl/keystores/{uid}/certificates/import	Import a private key and certificate chain into the keystore.

Truststores

Method	Path	Purpose
GET	/plugins/ssl/truststores	List truststores.
POST	/plugins/ssl/truststores	Create a truststore entry.
GET	/plugins/ssl/truststores/{uid}	Get a specific truststore.
DELETE	/plugins/ssl/truststores/{uid}	Delete a truststore.
GET	/plugins/ssl/truststores/{uid}/certificates	List trusted certificates in a truststore.
GET	/plugins/ssl/truststores/{uid}/certificates/{alias}	Get a trusted certificate by alias.
DELETE	/plugins/ssl/truststores/{uid}/certificates/{alias}	Delete a trusted certificate by alias.