SSL Settings Plugin for BridgeLink

User Manual

Table of Contents

1 Introduction

1.1 What Is This Plugin?

The SSL Settings Plugin adds support for SSL/TLS encryption to your BridgeLink integration channels. It allows you to secure the connections used to receive and send messages across a range of commonly used connectors, including:

- **HTTP Listener** and **HTTP Sender**
- **TCP Listener** and **TCP Sender**
- **Web Service Listener** and **Web Service Sender**

Once configured, this plugin ensures that data transmitted through these connectors is encrypted in transit, meaning it cannot be read or intercepted by unauthorized parties.

1.2 Why Does This Matter?

Healthcare data, including patient health information (PHI), is among the most sensitive data your organization handles. Regulations such as HIPAA require that this information be protected whenever it is transmitted over a network.

Without encryption, messages sent between systems are vulnerable to interception. The SSL Settings Plugin helps close this gap by enabling SSL/TLS encryption at the connector level, both when BridgeLink is receiving messages and when it is sending them.

In short: This plugin helps your organization transmit data securely, protect patient privacy, and meet compliance requirements.

1.3 Who Should Use This Plugin?

This plugin is intended for BridgeLink administrators responsible for configuring integration channels. No deep knowledge of SSL/TLS is required; this guide will walk you through each step.

2 Getting Started

Before proceeding with this guide, review the installation prerequisites and verify compatibility with your existing BridgeLink setup. The sections that follow will walk you through configuration, usage, and best practices for the SSL Settings Plugin.

3 Installation

3.1 Pre-Installed Packages

Depending on your subscription, the SSL Settings Plugin may already be included. The table below shows which packages come with the plugin pre-installed:

Provider	Package	Pre-Installed?
Innovar Healthcare (AWS Marketplace)	Open Source Mirth® Connect: Advanced with SSL	Yes
Innovar Healthcare (AWS Marketplace)	Open Source Mirth® Connect: Advanced with SSL Autoscaling	Yes
Innovar Healthcare (AWS Marketplace)	BridgeLink Standard Edition (Open Source Mirth Connect Fork)	Yes
Innovar Healthcare (AWS Marketplace)	BridgeLink Enterprise Edition (Open Source Mirth Connect Fork)	Yes

If you are on one of the packages above, you can skip ahead to Section 4: Configuration.

3.2 Manual Installation (Reinstall or Update)

If you need to install, reinstall, or update the plugin manually, follow the steps below.

> **Important:** Make sure you have the plugin ZIP file downloaded to your local machine and that you have administrator access to BridgeLink.

Steps:

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

> **Tip:** After restarting, navigate back to the Extensions screen to confirm the SSL Settings Plugin appears in the list and shows as enabled.

4 Configuration

4.1 Certificate Manager

The Certificate Manager is the central place where you manage all SSL/TLS certificates used by BridgeLink. You can access it by navigating to **Settings → Certificate Manager**.

There are two sections on this screen:

- **My Certificates:** Certificates that identify your BridgeLink server to other systems. These are stored in a Key Store.
- **Trusted Certificates:** Certificates from external systems that you want to trust. These are stored in a Trusted Store.

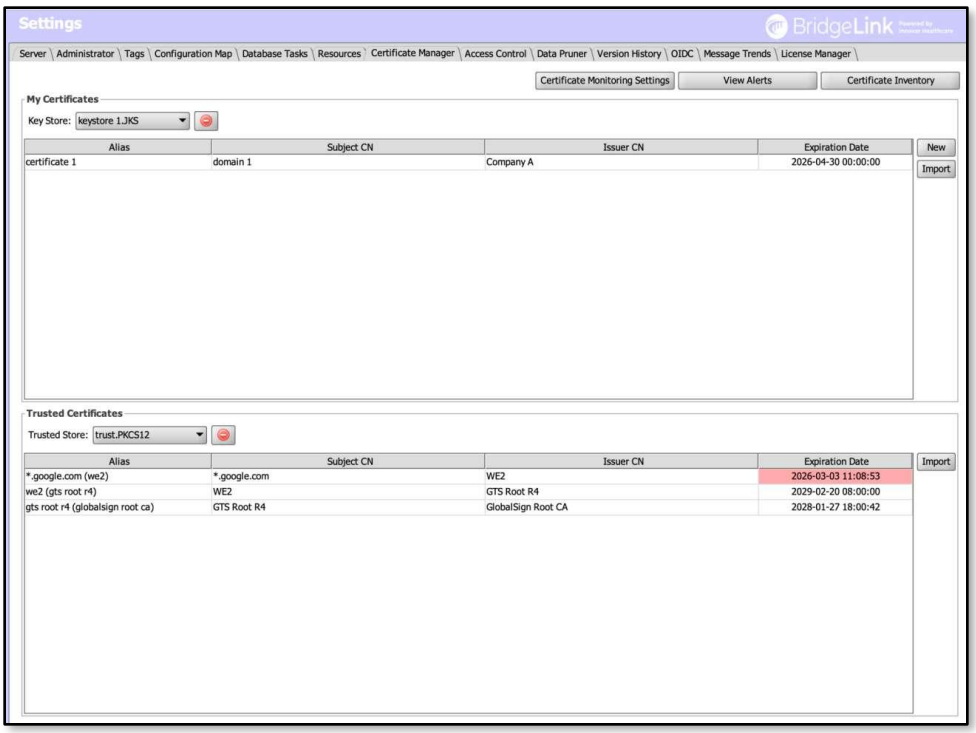

Figure 1: *Certificate Manager screen showing My Certificates and Trusted Certificates sections*

My Certificates

This section manages the certificates your BridgeLink server uses to identify itself when establishing a secure connection.

To create a new Key Store:

1. Click the **+** button next to the **Key Store** dropdown.
2. In the **Add Key Store** pop-up, fill in the following fields:

Field	Description
Name	A friendly name for this key store (e.g., "keystore 1")
Type	Select the key store format: JKS, JCEKS, or PKCS12
New Password	Set a password to protect this key store
Confirm New Password	Re-enter the password to confirm

3. Click **Save**. The new key store will appear in the **Key Store** dropdown.

Figure 2: *Add Key Store dialog*

Tip: If you are unsure which type to choose, JKS is the most common format used with Java-based systems like BridgeLink. Use PKCS12 if you are working with certificates from external systems or browsers.

To create a new certificate:

1. In the **Key Store** dropdown, select the key store you want to use.
2. Click **New**. The **Add Certificate** pop-up will appear.
3. Fill in the following fields:

Field	Description
Alias	A friendly name to identify this certificate (e.g., "my-server-cert")
Subject	The entity this certificate represents (e.g., your server). Click the wrench icon to fill in details like Common Name, Organization, etc.
Issuer	The authority issuing this certificate. For self-signed certificates, this is the same as the Subject. Click the wrench icon to fill in details.
Valid Date	The date from which the certificate is valid. Click the calendar icon to pick a date.
Expiration Date	The date the certificate expires. Click the calendar icon to pick a date.
Key Algorithm	The encryption algorithm. Default is RSA; recommended for most use cases.
Key Size	The strength of the encryption key. Default is 2048; recommended minimum for security.
Signature Algorithm	The algorithm used to sign the certificate. Default is SHA512withRSA; recommended for strong security.
New Password	Set a password to protect this certificate's private key.
Confirm New Password	Re-enter the password to confirm.

4. Optionally, under **Subject Alternative Names**, click **New** to add one or more DNS Names or IP Addresses that this certificate should also be valid for.

5. Click **Save**.

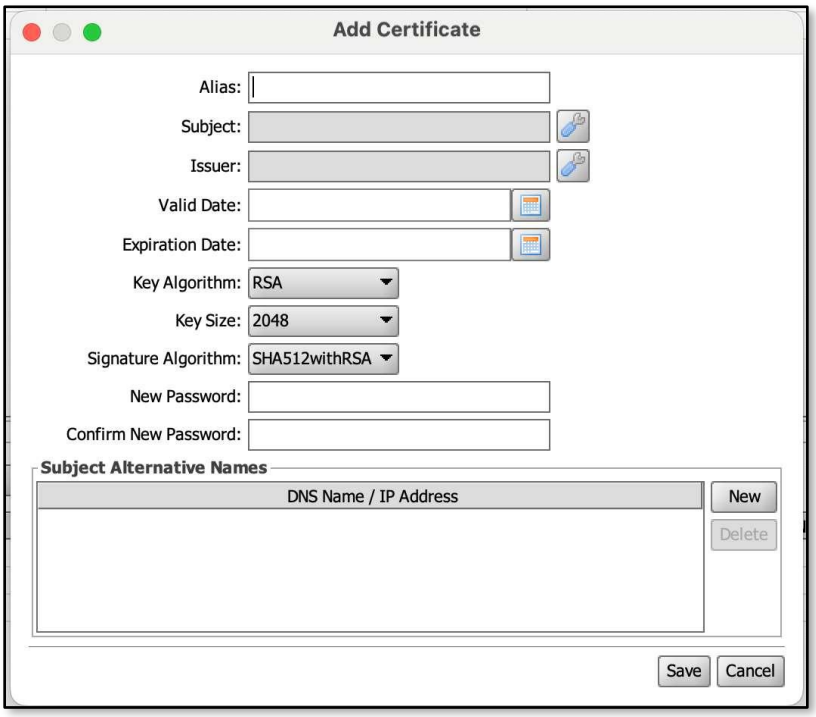

Figure 3: *Add Certificate dialog*

> **Tip:** For most BridgeLink deployments, the default values for Key Algorithm, Key Size, and Signature Algorithm are sufficient. Only change these if you have a specific security requirement from your IT or compliance team.

To import an existing certificate:

1. In the **Key Store** dropdown, select the target key store.
2. Click **Import**. The **Import My Certificates** pop-up will appear.
3. In the **From** dropdown, select the type of certificate you are importing:

Option	When to use
Keystore (JKS/JCEKS/PKCS12)	You have a keystore file (e.g., exported from another server or tool)
Private Key + Certificate (PEM/PKCS8/DER)	You have separate certificate and private key files (e.g., from a Certificate Authority)

If importing a Keystore file:

1. Click **Browse** next to **Keystore File** and select your keystore file.
2. In the **Keystore Type** dropdown, select the matching type: **JKS**, **JCEKS**, or **PKCS12**.
3. Enter the **Store Password** used to protect the keystore file.
4. Click **Next**. The **Select Certificate to Import** window will appear, listing all certificates found in the keystore file.
5. Use the **Search** box to filter by name if needed.
6. Select the certificate you want to import from the list.

7. Click **Import** to complete.

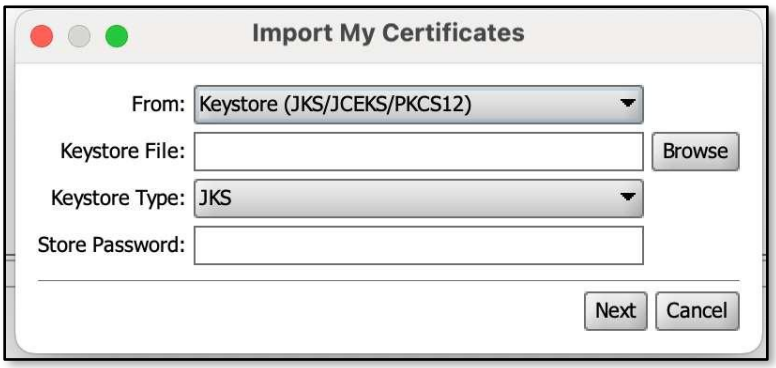

Figure 4: *Import My Certificates: Keystore file*

If importing a Private Key + Certificate:

1. Follow the prompts to browse and select your certificate file and private key file separately.
2. Click **Next**. The **Select Certificate to Import** window will appear.
3. Select the certificate from the list and click **Import**.

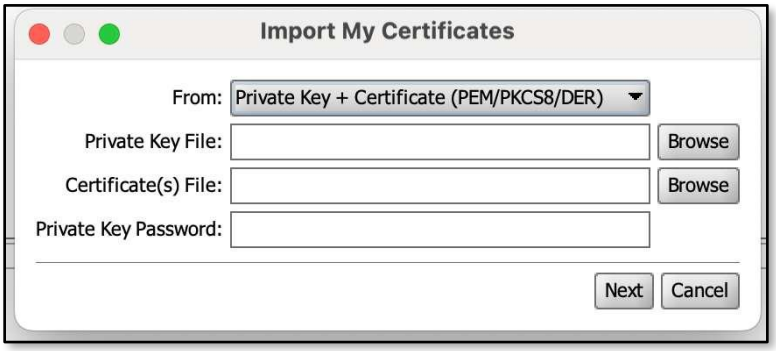

Figure 5: *Import My Certificates: Private Key + Certificate*

The imported certificate will appear in the **My Certificates** table with its Alias, Subject CN, Issuer CN, and Expiration Date.

> **Tip:** If you have multiple certificates in the keystore file, use the Search box to quickly find the one you need by Alias, Subject CN, or Issuer CN.
>
> **Important:** Keep track of your certificate's Expiration Date. An expired certificate will cause SSL connections to fail. Consider setting up Certificate Monitoring (see the Certificate Monitoring Settings button at the top of the screen) to receive alerts before a certificate expires.

Trusted Certificates

This section manages certificates from external systems that your BridgeLink channels will connect to or receive connections from. Adding a certificate here tells BridgeLink to trust that external system.

To manage an existing trusted certificate, right-click on it in the table. A context menu will appear with the following options:

Option	Description
View Details	Opens the Certificate Details window to inspect the certificate
Delete	Removes the certificate from the Trusted Store

> **Note:** Unlike My Certificates, trusted certificates cannot be exported; they belong to external systems and do not contain a private key.

To import a trusted certificate:

1. In the **Trusted Store** dropdown, select the trusted store you want to use (or create a new one by clicking the **+** button).
2. Click **Import**. The **Import Trusted Certificate** pop-up will appear.
3. In the **From** dropdown, select the source type:

Option	When to use
Keystore (JKS/JCEKS/PKCS12)	You have a keystore file containing the external system's certificate
Certificate (PEM)	You have a standalone PEM certificate file from the external system
URL	You want to fetch the certificate directly from a remote server URL

If importing from a Keystore file:

1. Click **Browse** and select your keystore file.
2. Select the **Keystore Type** (JKS, JCEKS, or PKCS12) and enter the **Store Password**.
3. Click **Next**, then select the certificate from the list and click **Import**.

Figure 6: *Import Trusted Certificate: Keystore*

If importing from a PEM Certificate file:

1. Click **Browse** and select the .pem certificate file.
2. Click **Next**, then select the certificate from the list and click **Import**.

Figure 7: *Import Trusted Certificate: PEM*

If importing from a URL:

1. Enter the URL of the remote server.
2. Click **Next**; BridgeLink will retrieve the certificate automatically.
3. Select the certificate from the list and click **Import**.

Figure 8: *Import Trusted Certificate: URL*

The imported certificate will appear in the **Trusted Certificates** table with its Alias, Subject CN, Issuer CN, and Expiration Date.

Managing Certificates

To manage an existing certificate, right-click on it in the table. A context menu will appear with the following options:

Option	Description
View Details	Opens the Certificate Details window to inspect the certificate
Export → Export Certificate Chain	Exports the full certificate chain (root CA + intermediate + certificate) to a file. Supports PEM, JKS, JCEKS, or PKCS12 format.
Export → Export Private Key	Exports the private key associated with this certificate
Export → Export Public Key	Exports only the public key to a file on your local machine
Delete	Removes the certificate from the store

To export a certificate:

1. Right-click the certificate and hover over **Export**, then select the desired option.
2. The **Export Certificate** pop-up will appear.
3. In the **Certificate Format** dropdown, select the format:

Format	What it exports
PEM (Certificate only)	Certificate file only; no private key. Browse to select a save location, then click Export.
JKS (Certificate + Private Key)	Certificate and private key bundled in a JKS keystore. Requires a Keystore Password.
JCEKS (Certificate + Private Key)	Same as JKS but using the JCEKS format. Requires a Keystore Password.
PKCS12 (Certificate + Private Key)	Certificate and private key bundled in a PKCS12 file. Requires a Keystore Password.

Figure 9: *Export Certificate dialog*

4. Click **Browse** to choose the export location on your machine.
5. If the selected format includes a Private Key, enter and confirm a **Keystore Password** to protect the exported file.
6. Click **Export**.

To export a private key:

Right-click the certificate and select Export → Export Private Key. In the Export Private Key pop-up, select the Key Format:

Format	Description
PEM (PKCS1)	Standard PEM format, widely supported
PEM (PKCS8)	Newer PEM format, commonly used with modern systems
PEM (PKCS8 Encrypted)	PKCS8 format with additional encryption for extra security
DER (PKCS8)	Binary format, used by some Java and enterprise systems

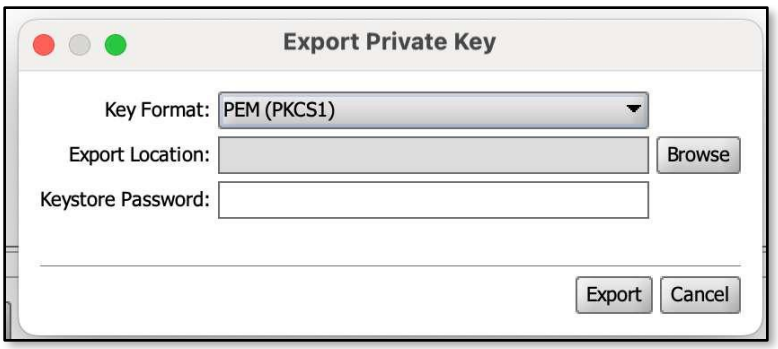

Figure 10: *Export Private Key dialog*

> **Tip:** If you are unsure which format to use, PEM (PKCS1) is the most widely compatible option for most systems.

To export a public key:

Right-click the certificate and select Export → Export Public Key. In the pop-up, click Browse to choose the save location, then click Export.

> **Warning:** Keep your Private Key secure. Never share it with unauthorized parties; it is used to decrypt data and authenticate your server. Only export it when absolutely necessary, such as when migrating to a new server.

Viewing Certificate Details

To view the full details of a certificate, right-click on it and select View Details. The Certificate Details window will open, showing the following information:

Field	Description
Certificate Hierarchy	A visual tree showing the chain of trust, from the Root CA down to the certificate itself
Version	The certificate format version
Subject	The entity this certificate belongs to (e.g., CN=*.google.com)
Issuer	The authority that issued this certificate
Serial Number	A unique identifier for this certificate
Valid From	The date the certificate became valid
Valid Until	The date the certificate expires
Public Key	The encryption algorithm used for the public key (e.g., RSA, EC)
Key Size	The size of the encryption key in bits
Signature Algorithm	The algorithm used to sign the certificate (e.g., SHA256withECDSA)

Subject Alternative Names	Additional DNS names or IP addresses this certificate is valid for

Figure 11: *Certificate Details window*

From this window, click **PEM** to view or export the certificate in PEM format, or click **OK** to close.

> **Note:** When viewing Certificate Details via right-click, the Import button is not available. To import a certificate, use the Import button on the main Certificate Manager screen instead.

Certificate Manager Overview

Column	Description
Alias	A friendly name to identify the certificate
Subject CN	The Common Name of the entity the certificate belongs to
Issuer CN	The Common Name of the authority that issued the certificate
Expiration Date	The date the certificate expires; monitor this closely

Certificate Monitoring Settings

To avoid unexpected SSL failures due to expired certificates, BridgeLink can automatically monitor your certificates and send alerts before they expire. Click Certificate Monitoring Settings at the top of the Certificate Manager screen to configure this.

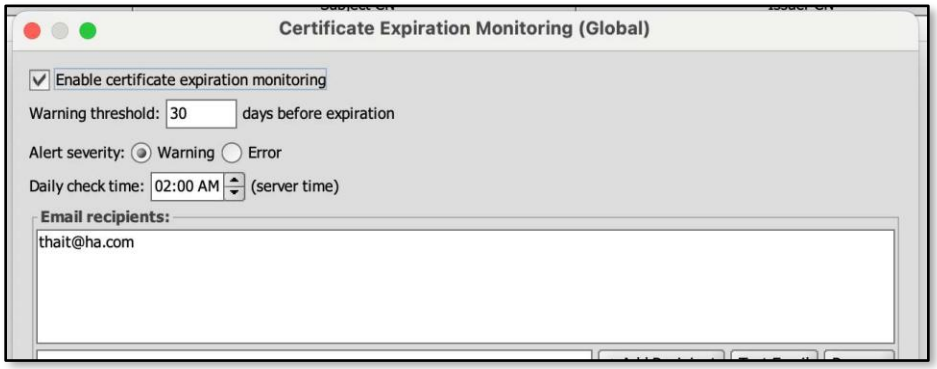

Figure 12: *Certificate Expiration Monitoring (Global) settings*

Field	Description
Enable certificate expiration monitoring	Check this box to turn on automatic monitoring
Warning threshold	Number of days before expiration to trigger an alert. Default is 30 days.
Alert severity	Choose Warning or Error; determines how the alert is logged in BridgeLink
Daily check time	The time BridgeLink checks certificates each day (server time). Default is 02:00 AM.
Email recipients	List of email addresses to notify. Enter an address and click + Add Recipient. Click Test Email to verify. Click Remove to delete.

Notification Methods:

Option	Description
Dashboard alerts	Shows alerts on the BridgeLink dashboard
Email notifications	Sends email alerts to the configured recipients
Include alert page link in email	Includes a direct link to the alert in the email body. Requires a Server URL (e.g., https://localhost:8443).
System event log	Logs alerts to the BridgeLink system event log

Click **Save** to apply. Click **Check All Now** to run an immediate check on all certificates.

View Alerts

Click **View Alerts** at the top of the Certificate Manager to see a list of certificates that are expiring soon or have already expired.

Alerts can be viewed in two tabs:

- **By Certificate:** groups alerts by certificate, showing expiry date, Subject, Issuer, and which channels are using it
- **By Channel:** groups alerts by channel

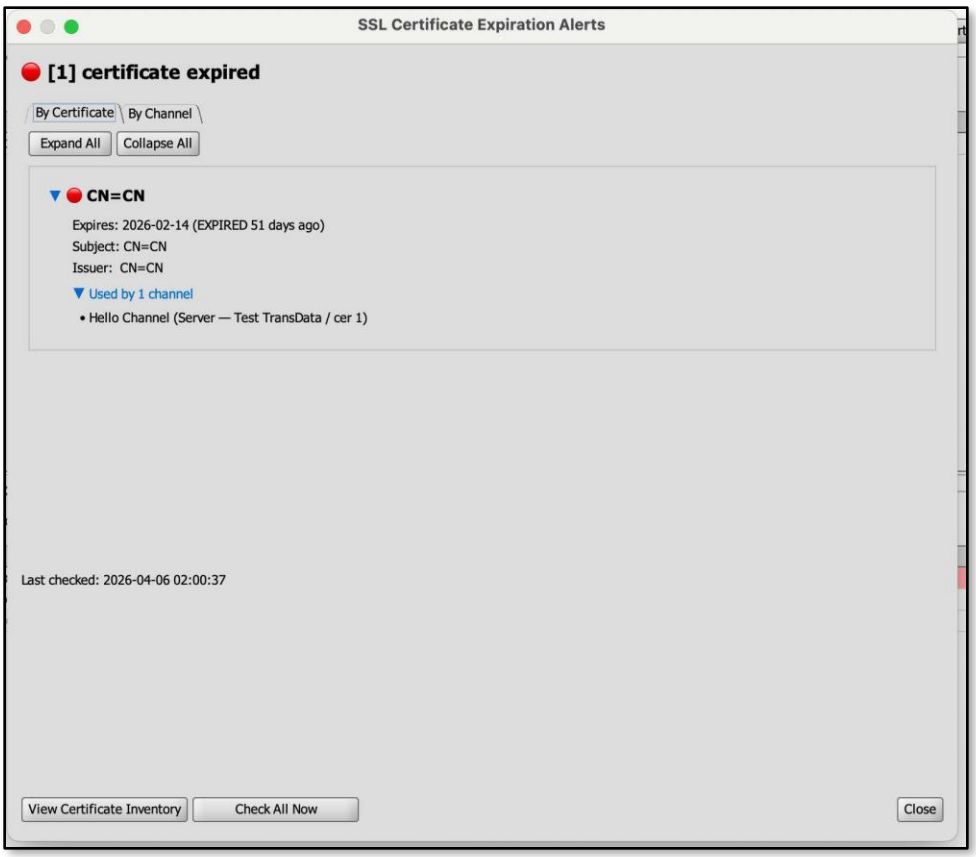

Figure 13: *SSL Certificate Expiration Alerts window*

Each alert shows a red indicator if the certificate has expired. Click **Expand All** or **Collapse All** to manage the view. Click **Check All Now** to refresh, or **View Certificate Inventory** to see the full inventory.

Certificate Inventory

Click **Certificate Inventory** at the top of the Certificate Manager to see a full list of all certificates across all stores, with their current status.

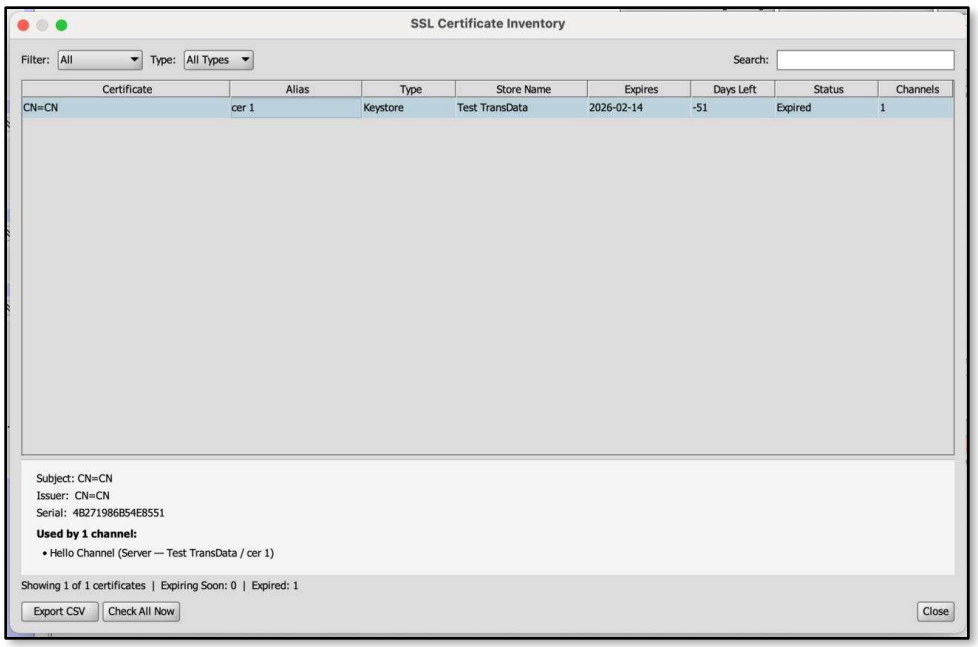

Figure 14: *SSL Certificate Inventory window*

Column	Description
Certificate	The Common Name (CN) of the certificate
Alias	The friendly name assigned to the certificate
Type	Whether it is stored in a Keystore or Trusted Store
Store Name	The name of the key store or trusted store
Expires	The expiration date
Days Left	Number of days remaining (negative = already expired)
Status	Active, Expiring Soon, or Expired
Channels	Number of channels currently using this certificate

Select a certificate to see its full details (Subject, Issuer, Serial, and which channels are using it) in the panel below. Click **Export CSV** to download the full inventory, or **Check All Now** to refresh the status.

> **Important:** Any certificate showing Expired or a negative Days Left value should be renewed immediately to avoid SSL connection failures in your channels.

4.2 Connector Configuration

HTTP Listener

To enable SSL on an HTTP Listener channel, open the channel in BridgeLink, navigate to the **Source** connector settings, and click the **SSL Settings** button. The **Server SSL Settings** window will appear.

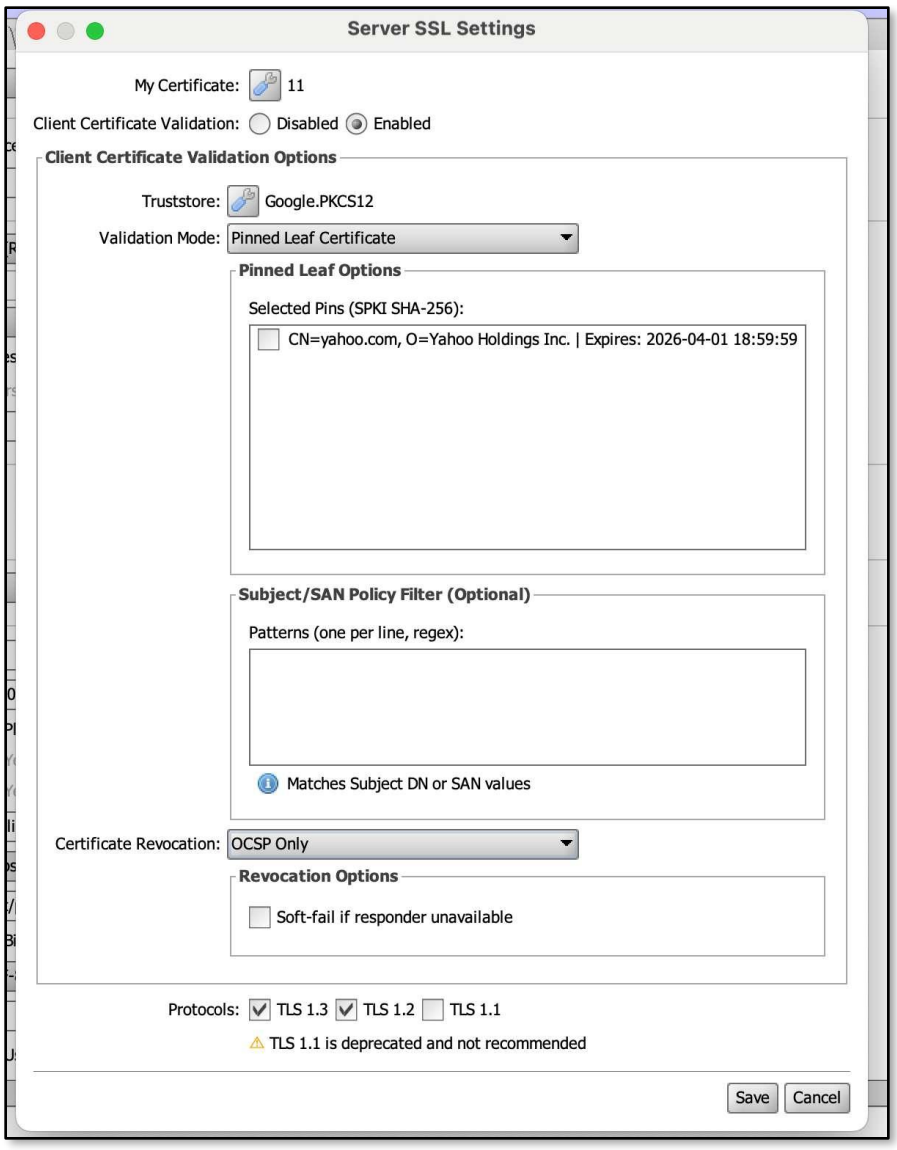

Figure 15: *Server SSL Settings for HTTP Listener*

Server SSL Settings:

Field	Description
My Certificate	Select the certificate your BridgeLink server will use to identify itself to connecting clients. Click the wrench icon to browse your Key Store.
Client Certificate Validation	Choose Disabled to accept all clients without certificate validation, or Enabled to require clients to present a valid certificate.

Client Certificate Validation Options:

These options appear when Client Certificate Validation is set to Enabled.

Field	Description
Truststore	Select the Trusted Store containing the certificates of clients you want to allow. Click the wrench icon to browse your Trusted Stores.
Validation Mode	Controls how the client's certificate is validated. See options below.

Validation Mode options:

Mode	Description
PKIX Chain Validation (CA-based)	Validates the client's certificate against a Certificate Authority (CA) chain. Use this when clients present certificates signed by a known CA.
Pinned Leaf Certificate	Only accepts specific certificates that have been explicitly pinned. Use this for tighter control over which clients can connect.
Trust All (No Validation)	Accepts any client certificate without validation. Not recommended for production environments.

If Validation Mode is set to **Pinned Leaf Certificate**, the following options appear:

Pinned Leaf Options: Selected Pins (SPKI SHA-256): This list shows the certificates currently pinned. Check the checkbox next to a certificate to activate it as a trusted pin. Each entry shows the certificate's CN and expiration date.

Subject/SAN Policy Filter (Optional): Enter one regex pattern per line to filter clients by their certificate's Subject DN or SAN values. Only clients whose certificates match one of the patterns will be allowed to connect. Leave empty to skip this filter.

Certificate Revocation:

Option	Description
Disabled	No revocation check is performed
OCSP Only	Checks revocation status using the Online Certificate Status Protocol (OCSP). Default and recommended option.
CRL Only	Checks revocation using a Certificate Revocation List (CRL) downloaded from the CA
OCSP with CRL Fallback	Tries OCSP first; if unavailable, falls back to CRL

Revocation Options:

Option	Description
Soft-fail if responder unavailable	If checked, BridgeLink will allow the connection even if the OCSP responder or CRL cannot be reached. If unchecked, the connection will be rejected when revocation status cannot be verified.

Protocols:

Select which TLS protocol versions are allowed for incoming connections:

Protocol	Status
TLS 1.3	Recommended; most secure, enable by default
TLS 1.2	Recommended; widely supported, enable by default
TLS 1.1	Deprecated and not recommended; disable unless required for legacy system compatibility

HTTP Sender

To enable SSL on an HTTP Sender channel, open the channel in BridgeLink, navigate to the **Destination** connector settings, and click the **SSL Settings** button. The **Client SSL Settings** window will appear.

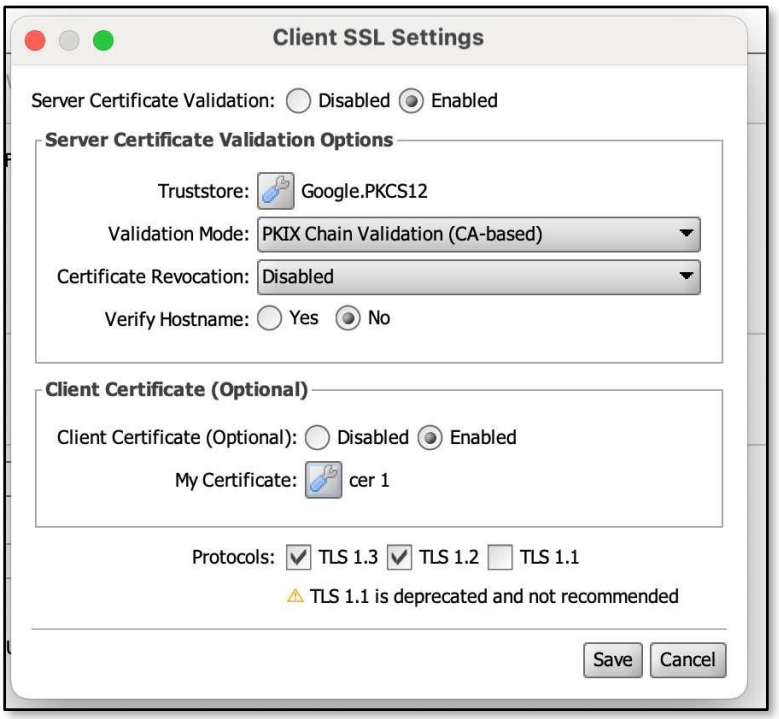

Figure 16: *Client SSL Settings for HTTP Sender*

Server Certificate Validation:

Field	Description
Server Certificate Validation	Choose Disabled to skip validation of the server's certificate, or Enabled to verify the server's certificate before connecting.
Truststore	Select the Trusted Store containing the server's certificate. Click the wrench icon to browse.

Validation Mode	Controls how the server's certificate is validated. Options: PKIX Chain Validation (CA-based), Pinned Leaf Certificate, or Trust All (No Validation).
Certificate Revocation	Controls revocation checking. Options: Disabled, OCSP Only, CRL Only, OCSP with CRL Fallback.
Soft-fail if responder unavailable	If checked, allows the connection even if the OCSP/CRL responder is unreachable.
Verify Hostname	Choose Yes to verify that the server's certificate hostname matches the destination URL (recommended for production). Choose No to skip hostname verification.

Client Certificate (Optional):

This section allows BridgeLink to present its own certificate to the destination server, required when the server enforces mutual TLS (mTLS).

Field	Description
Client Certificate	Choose Disabled to connect without a client certificate, or Enabled to present one.
My Certificate	Select the certificate BridgeLink will use to identify itself to the server. Click the wrench icon to browse your Key Store. Only visible when Client Certificate is Enabled.

Protocols:

Protocol	Status
TLS 1.3	Recommended
TLS 1.2	Recommended
TLS 1.1	Deprecated and not recommended

Click **Save** to apply the SSL settings to the connector.

TCP Listener

To enable SSL on a TCP Listener, open the channel, navigate to the **Source** connector settings, and click the **SSL Settings** button. The **Server SSL Settings** window will appear.

The configuration is identical to the HTTP Listener; refer to that section for full field descriptions.

TCP Sender

To enable SSL on a TCP Sender, open the channel, navigate to the **Destination** connector settings, and click the **SSL Settings** button. The **Client SSL Settings** window will appear.

The configuration is identical to the HTTP Sender; refer to that section for full field descriptions.

Web Service Listener

To enable SSL on a Web Service Listener, open the channel, navigate to the **Source** connector settings, and click the **SSL Settings** button. The **Server SSL Settings** window will appear.

The configuration is identical to the HTTP Listener; refer to that section for full field descriptions.

Web Service Sender

To enable SSL on a Web Service Sender, open the channel, navigate to the **Destination** connector settings, and click the **SSL Settings** button. The **Client SSL Settings** window will appear.

The configuration is identical to the HTTP Sender; refer to that section for full field descriptions.

Summary: All Listener connectors (HTTP, TCP, Web Service) use the Server SSL Settings screen. All Sender connectors use the Client SSL Settings screen. The fields and options are the same across connector types.

4.3 SSL Helper Functions (JavaScript Transformer)

In addition to connector-level SSL settings, the SSL Settings Plugin provides SSL Helper Functions that can be used directly in a channel's JavaScript transformer. This allows you to make HTTPS calls with full SSL control from within your channel logic.

To access the built-in code templates, open a JavaScript transformer in your channel, then in the **Reference** panel on the right, set **Category** to **SSL Helper Functions**. You will see two templates:

- **HTTPS GET with SSLHelper**
- **HTTPS POST with SSLHelper**

SSL Parameters:

Both templates use an sslParams map to configure SSL behavior. The available parameters are:

Parameter	Required	Description
trustStoreUid	Yes	The UID of the Trusted Store to use for server certificate validation
validationMode	No	Certificate validation mode: pkix (default), pinned_leaf, or trust_all
pinnedLeafPins	Conditional	Required if validationMode is pinned_leaf. A list of SPKI SHA-256 hex strings (64 chars each)
revocationMode	No	Revocation check mode: disabled (default), ocsp_only, crl_only, or ocsp_with_crl_fallback
softFail	No	true = allow connection if OCSP/CRL responder is unavailable. Default: false

hostnameVerify	No	true = strict hostname verification (recommended for production). false = skip (development only)
clientCertEnabled	No	true = present a client certificate (mTLS). Default: false
keystoreUid	Conditional	Required if clientCertEnabled is true. The UID of the Key Store containing the client certificate
certAlias	Conditional	Required if clientCertEnabled is true. The alias of the client certificate in the Key Store
tls13	No	true = enable TLS 1.3 (recommended). Default: true
tls12	No	true = enable TLS 1.2 (recommended). Default: true
tls11	No	false = disable TLS 1.1 (deprecated, not recommended). Default: false

HTTP Parameters:

Both templates also accept an httpParams map for controlling HTTP behavior:

Parameter	Description
connectTimeoutMs	Connection timeout in milliseconds. Default: 15000 (15 seconds)
readTimeoutMs	Read timeout in milliseconds. Default: 30000 (30 seconds)
throwOnHttpError	If true, throws an exception on HTTP error responses (status ≥ 300). Default: false
captureHeaders	If true, includes response headers in the result map. Default: false
followRedirects	If true, automatically follows HTTP redirects. Default: true

Response:

Both SSLHelper.httpsGet() and SSLHelper.httpsPost() return a map with the following keys:

Key	Description
status	HTTP response status code (e.g., 200, 404)
reason	HTTP status reason phrase (e.g., OK, Not Found)
body	Response body as a string
headers	Response headers (only populated if captureHeaders is true)

Tip: Use res.get("status") to check the response code and handle errors accordingly, as shown in the code templates.

Important: hostnameVerify should always be set to true in production environments. Setting it to false is only acceptable during development and testing.

5 API Reference

The SSL Settings Plugin exposes a REST API that allows you to manage keystores, truststores, and certificates programmatically. This is useful for automation, scripting, or integration with external systems.

You can access the full interactive API documentation by navigating to: https://<yourserver>:8443/api/#/Plugin%20Services

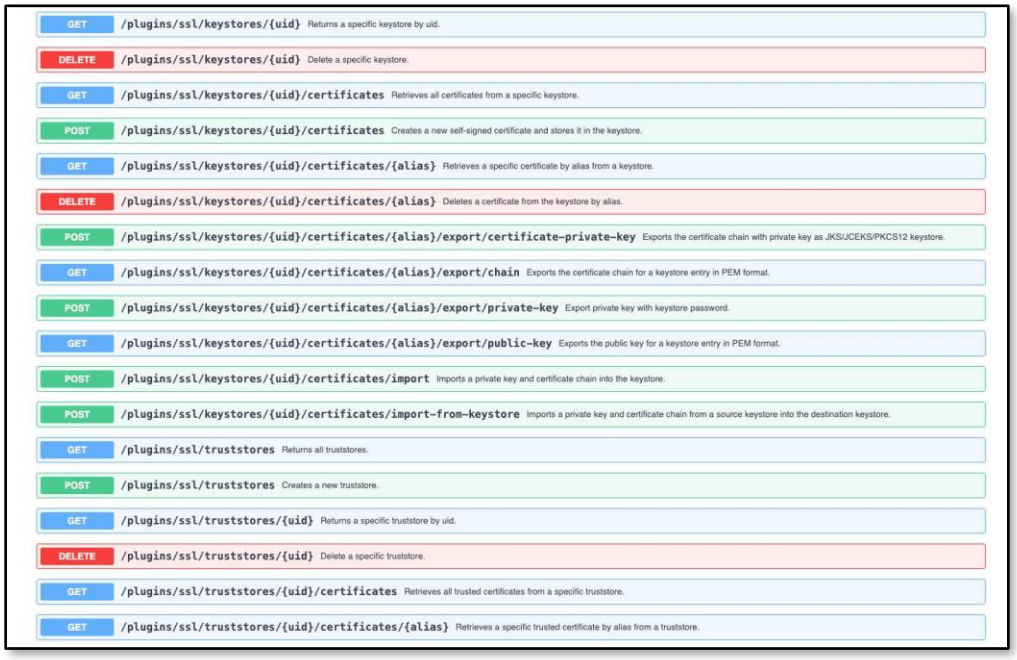

Figure 17: *SSL Plugin API endpoints*

5.1 Certificate Monitoring

Method	Endpoint	Description
GET	/plugins/ssl/certificate-monitoring/alert-page	Get the certificate monitoring alert page
GET	/plugins/ssl/certificate-monitoring/alerts	Get current certificate expiration alerts
POST	/plugins/ssl/certificate-monitoring/check	Trigger an immediate expiration check across all SSL-enabled channels
GET	/plugins/ssl/certificate-monitoring/last-scan-time	Get the time of the last certificate scan
GET	/plugins/ssl/certificate-monitoring/snapshots	Get certificate monitoring snapshots
POST	/plugins/ssl/certificate-monitoring/test-email	Send a test email notification

5.2 Keystores

Method	Endpoint	Description
GET	/plugins/ssl/keystores	Return all keystores
POST	/plugins/ssl/keystores	Create a new keystore
GET	/plugins/ssl/keystores/{uid}	Return a specific keystore by UID
DELETE	/plugins/ssl/keystores/{uid}	Delete a specific keystore

Keystore Certificates:

Method	Endpoint	Description
GET	/plugins/ssl/keystores/{uid}/certificates	Retrieve all certificates from a keystore
POST	/plugins/ssl/keystores/{uid}/certificates	Create a new self-signed certificate in a keystore
GET	/plugins/ssl/keystores/{uid}/certificates/{alias}	Retrieve a specific certificate by alias
DELETE	/plugins/ssl/keystores/{uid}/certificates/{alias}	Delete a certificate from a keystore
POST	/plugins/ssl/keystores/{uid}/certificates/import	Import a private key and certificate chain into a keystore
POST	/plugins/ssl/keystores/{uid}/certificates/import-from-keystore	Import a private key and certificate chain from another keystore

Keystore Certificate Exports:

Method	Endpoint	Description
POST	/plugins/ssl/keystores/{uid}/certificates/{alias}/export/certificate-private-key	Export certificate chain with private key as JKS/JCEKS/PKCS12
GET	/plugins/ssl/keystores/{uid}/certificates/{alias}/export/chain	Export the certificate chain in PEM format
POST	/plugins/ssl/keystores/{uid}/certificates/{alias}/export/private-key	Export the private key
GET	/plugins/ssl/keystores/{uid}/certificates/{alias}/export/public-key	Export the public key in PEM format

5.3 Truststores

Method	Endpoint	Description
GET	/plugins/ssl/truststores	Return all truststores
POST	/plugins/ssl/truststores	Create a new truststore
GET	/plugins/ssl/truststores/{uid}	Return a specific truststore by UID
DELETE	/plugins/ssl/truststores/{uid}	Delete a specific truststore

Truststore Certificates:

Method	Endpoint	Description
GET	/plugins/ssl/truststores/{uid}/certificates	Retrieve all trusted certificates from a truststore
GET	/plugins/ssl/truststores/{uid}/certificates/{alias}	Retrieve a specific trusted certificate by alias
DELETE	/plugins/ssl/truststores/{uid}/certificates/{alias}	Delete a trusted certificate from a truststore
GET	/plugins/ssl/truststores/{uid}/certificates/{alias}/pem	Retrieve a trusted certificate in PEM format
POST	/plugins/ssl/truststores/{uid}/certificates/import-from-truststore	Import a trusted certificate from another truststore

> **Tip:** Replace {uid} with the keystore or truststore UID, and {alias} with the certificate alias. You can find these values in the Certificate Manager screen or by calling the respective GET endpoints.

6 Document Control

Revision History:

Version	Date	Author	Description
3.1.0	April 21, 2026	Calvin Redding	Initial release for SSL Settings Plugin v3.1.0

Document Information:

Property	Value
Document Title	SSL Settings Plugin for BridgeLink: User Manual
Product Version	3.1.0
Classification	External
Owner	Innovar Healthcare Systems Group

Versioning Policy:

This document follows the plugin version number. Each plugin release (e.g., 3.1.0, 3.2.0) receives a corresponding manual version. Doc-only corrections between plugin releases may append a revision suffix (e.g., 3.1.0-r2).