BridgeLink SSL Settings Plugin: User Guide

For BridgeLink WebAdmin

Table of Contents

About This Guide

SSL Settings is a BridgeLink WebAdmin plugin that adds SSL/TLS encryption to your integration channels. It secures the connections used by **HTTP Listener**, **HTTP Sender**, **TCP Listener**, **TCP Sender**, **Web Service Listener**, and **Web Service Sender** connectors, and it gives administrators a central place to manage certificates, key stores, and trusted stores. This guide is for BridgeLink administrators who configure integration channels; no prior SSL/TLS expertise is required. By the end of this guide, you will be able to install and license the SSL Settings Plugin, manage certificates in the Certificate Manager, enable SSL on a connector, monitor certificate expiration, and call the plugin's SSL helper functions and REST API from your own channel code.

Requirements

- BridgeLink 26.6.0 or later
- A valid Innovar Healthcare license for SSL Settings, installed on the **License Manager** tab under **Settings** (see Installation and Activation)
- No special external prerequisites. You can create self-signed certificates directly in the plugin, or bring your own certificate and private key files (PEM, PKCS8, DER, JKS, JCEKS, or PKCS12) issued by your certificate authority.

Installation and Activation

SSL Settings ships pre-installed with most BridgeLink editions. If it does not already appear in your list of installed plugins, install it manually:

1. Sign in to **WebAdmin** as an administrator.
2. Click **Extensions** in the navigation.
3. Scroll down to the **Install Extension from File System** section.
4. Click **Browse...** and select the SSL Settings plugin `.zip` file on your local machine.
5. Click **Install**.
6. Restart the BridgeLink server. WebAdmin shows a **Restart Required** banner until you do.
7. After the restart, click **Refresh**.
8. Find the plugin under **Installed Plugins**, where it is listed as **SSL Settings**, and confirm its status shows **Enabled**.
9. If its status shows **Disabled** instead, select it, click **Enable**, and restart the server again.

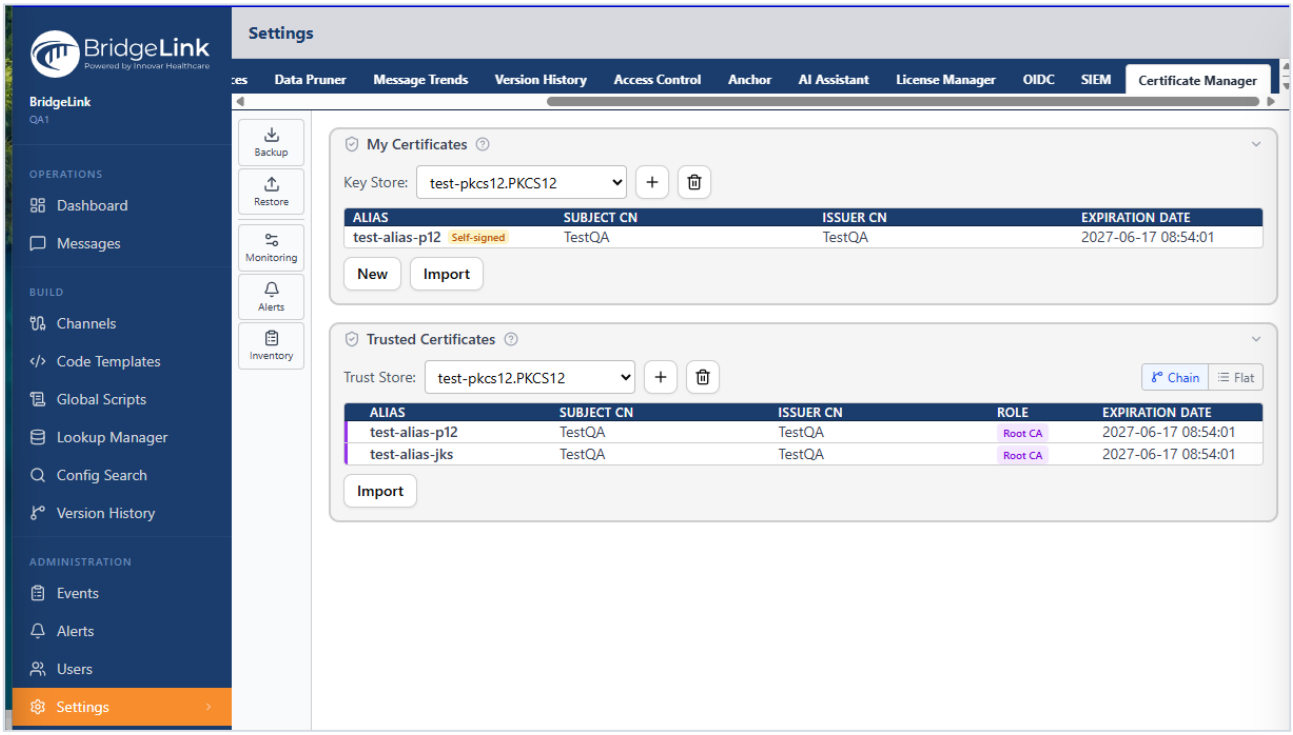
To install or confirm your license:

10. Click **Settings** in the navigation, under the **Administration** sidebar group.
11. Click the **License Manager** tab, then the **Installed Licenses** sub-tab.
12. Click **Install License** and choose the SSL Settings license file, a `.json` file supplied by Innovar Healthcare.
13. Review the **Install License** dialog, which previews the license's **Customer**, **Products**, **Issued**, **Expires**, and **Server match** (**Server match** reads "Yes, this server is licensed" when the license covers this server), then click **Install**. A confirmation toast reads "License installed."
14. Restart BridgeLink when prompted. A banner reads "License changes will take effect after restarting BridgeLink."
15. Click the **License Status** sub-tab and confirm the **SSL Settings** row shows **Active** under **Status**, with the correct **Expires** date (or **Lifetime**).

Configuration

Certificate Manager

The Certificate Manager is the central place where you manage all SSL/TLS certificates used by BridgeLink. Click **Settings** in the navigation, then select the **Certificate Manager** tab. The screen has two sections, **My Certificates** and **Trusted Certificates**, plus a toolbar with **Backup**, **Restore**, **Monitoring**, **Alerts**, and **Inventory** buttons.



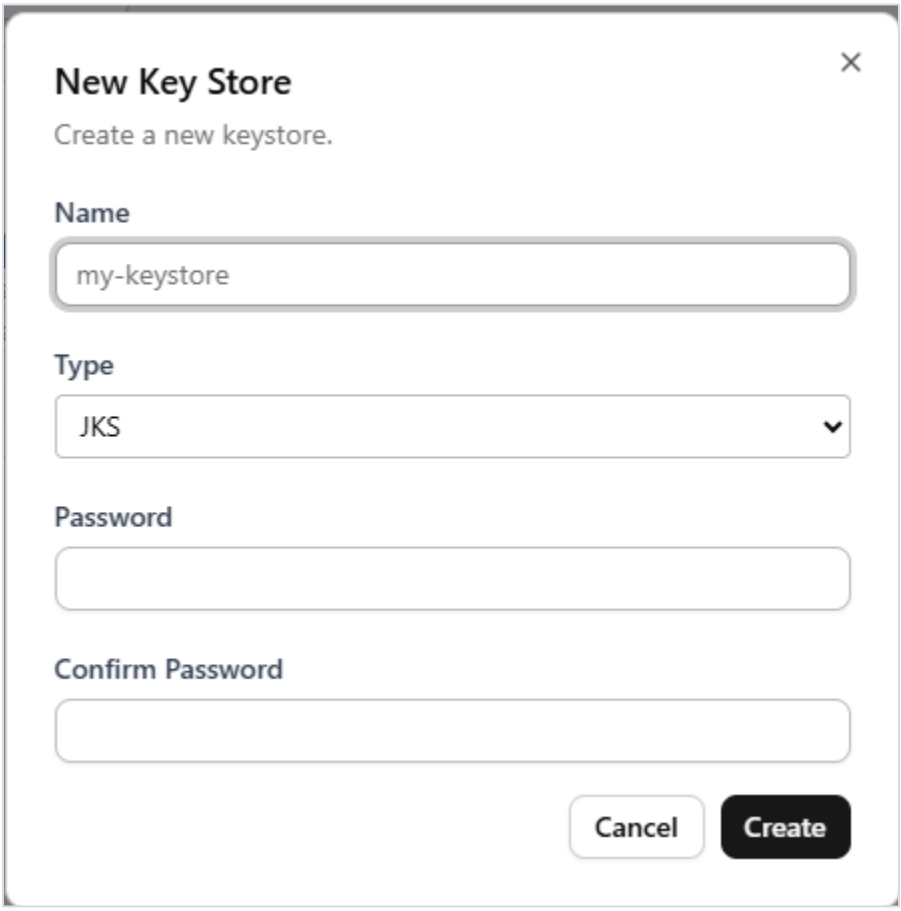
Certificate Manager screen showing My Certificates and Trusted Certificates

My Certificates

This section manages the certificates your BridgeLink server presents to identify itself when establishing a secure connection. Each certificate lives inside a **Key Store**, selected from the **Key Store** dropdown.

To create a key store:

1. Click the **+** button next to the **Key Store** dropdown.
2. In the **New Key Store** window, enter a **Name** for the key store.
3. Select the **Type**: `JKS`, `JCEKS`, or `PKCS12`.
4. Enter a **Password**.
5. Re-enter the same value in **Confirm Password**.
6. Click **Create**.

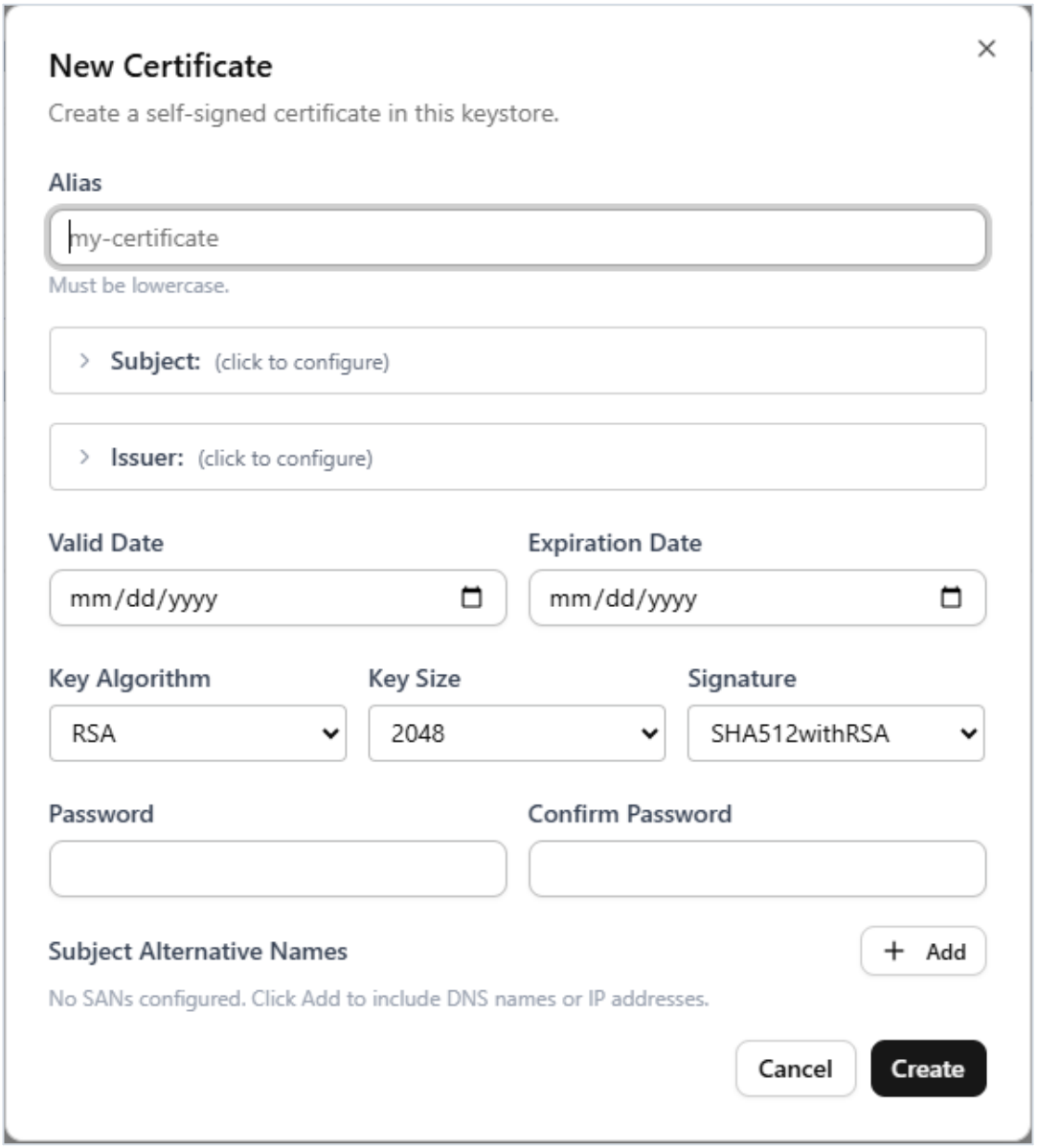


New Key Store dialog

To create a self-signed certificate:

1. In the **Key Store** dropdown, select the key store you want to use.
2. Click **New**. The **New Certificate** window opens.
3. Enter an **Alias** for the certificate. Aliases must be lowercase.
4. Click **Subject** to expand it, then fill in **Common Name (CN)** and any other distinguished-name fields you need.
5. Click **Issuer** to expand it and fill in the same fields. For a self-signed certificate, the Issuer is normally the same as the Subject.
6. Set the **Valid Date**.
7. Set the **Expiration Date**.
8. Review **Key Algorithm**, **Key Size**, and **Signature**. The defaults, `RSA`, `2048`, and `SHA512withRSA`, are sufficient for most deployments.
9. Enter a **Password**.
10. Re-enter the same value in **Confirm Password**.
11. Optionally, under **Subject Alternative Names**, click **Add** to include additional DNS names or IP addresses this certificate should also cover.

12. Click **Create**.



New Certificate dialog

To import a certificate into a key store:

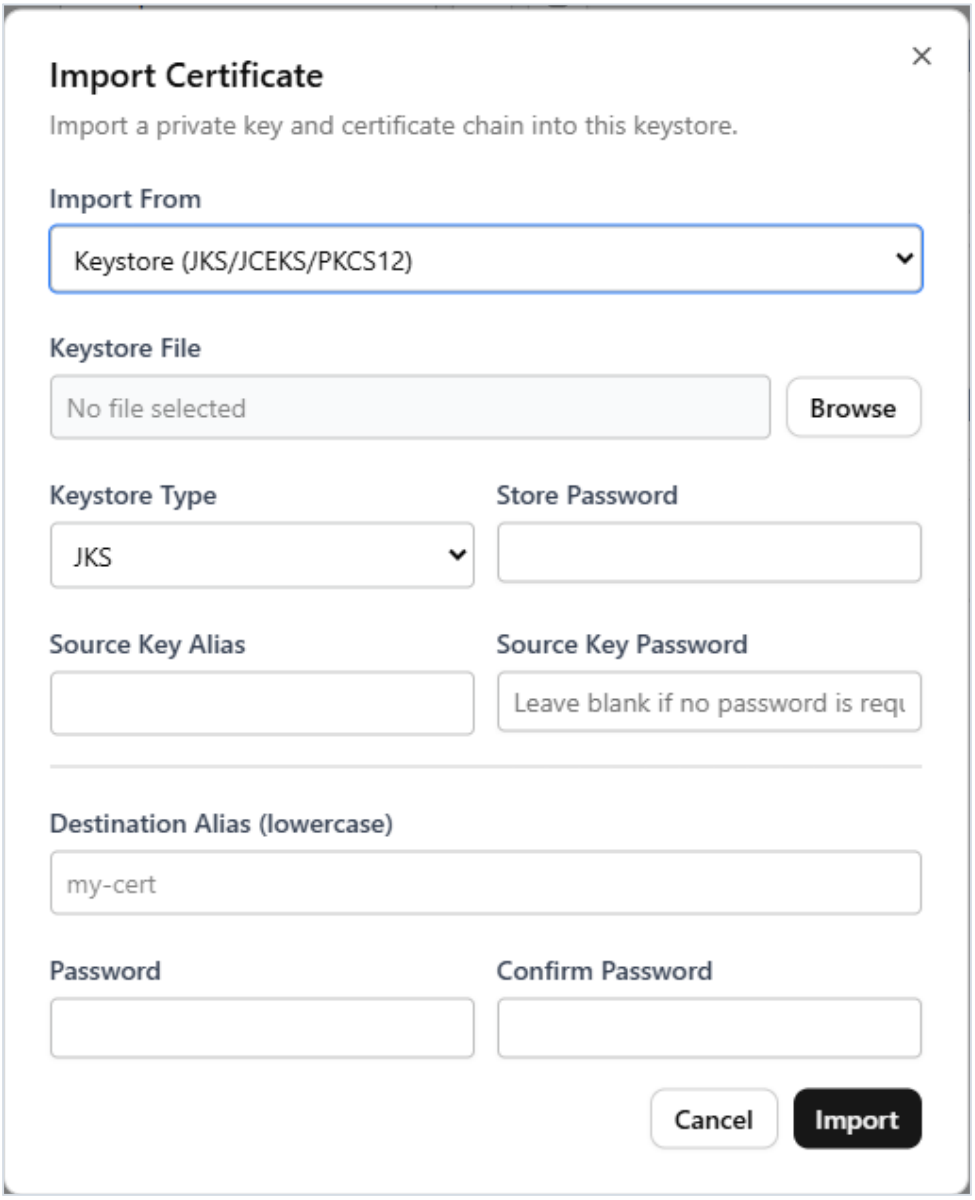
1. In the **Key Store** dropdown, select the target key store.
2. Click **Import**. The **Import Certificate** window opens.
3. In the **Import From** dropdown, choose the source:

Option	When to use
Keystore (JKS/JCEKS/PKCS12)	You have a keystore file, for example one exported from another server or tool.

Option	When to use
Private Key + Certificate (PEM/PKCS8/DER)	You have separate private key and certificate files, for example issued by a certificate authority.

If importing from a keystore file:

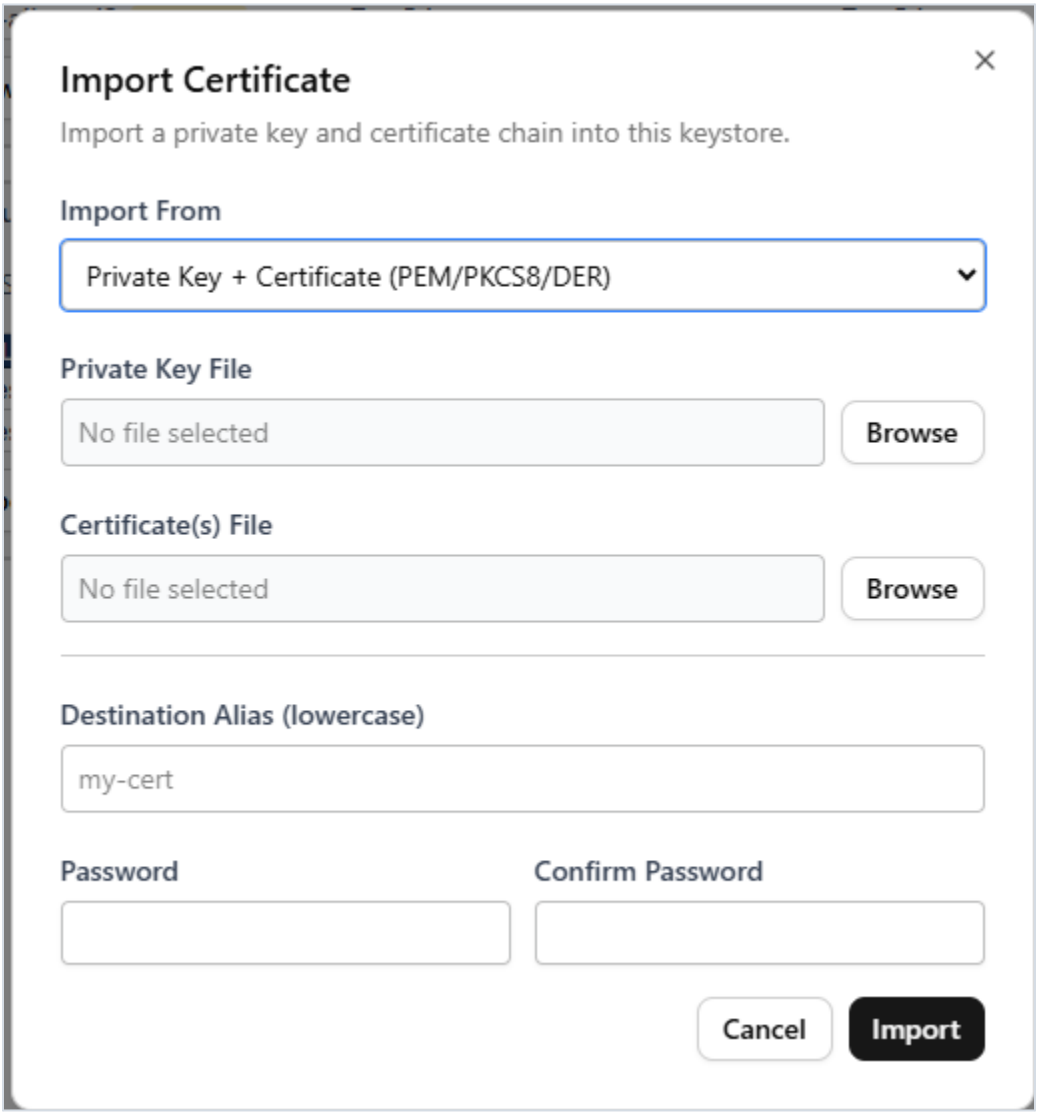
1. Click **Browse** next to **Keystore File** and select the file.
2. Select the **Keystore Type**: `JKS`, `JCEKS`, or `PKCS12`.
3. Enter the **Store Password**.
4. Enter the **Source Key Alias**.
5. Enter the **Source Key Password** if the key is separately protected. Leave it blank if none is required.
6. Enter a **Destination Alias** (lowercase).
7. Enter a **Password** for the imported certificate.
8. Re-enter the same value in **Confirm Password**.
9. Click **Import**.



Import Certificate: Keystore file

If importing from private key and certificate files:

1. Click **Browse** next to **Private Key File** and select the key file.
2. Click **Browse** next to **Certificate(s) File** and select the certificate file.
3. Enter a **Destination Alias** (lowercase).
4. Enter a **Password**.
5. Re-enter the same value in **Confirm Password**.
6. Click **Import**.



Import Certificate: Private Key + Certificate

The imported certificate appears in the **My Certificates** table with its **Alias**, **Subject CN**, **Issuer CN**, and **Expiration Date**. Self-signed certificates show a **Self-signed** badge.

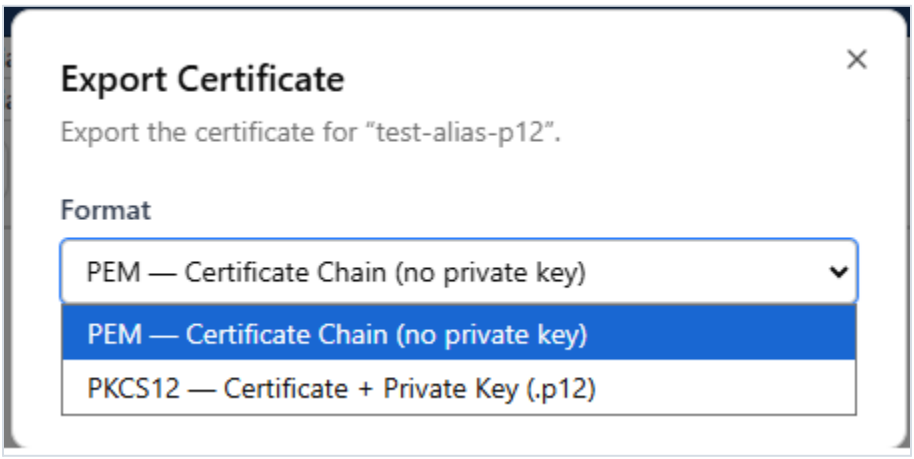
> If a private key file does not contain a recognizable private key block, the import is rejected with a message telling you to select a `.pem` or `.key` file with a `BEGIN PRIVATE KEY` block, or to import the certificate alone under Trusted Certificates instead.

To export a certificate, right-click it in the **My Certificates** table, choose **Export Certificate Chain**, and select a **Format**:

Format	What it exports
PEM	The certificate chain only. No private key, no password required.

Format	What it exports
PKCS12	The certificate chain bundled with the private key in a `.p12` file. Requires the **Keystore Password** (to read the private key) plus an **Output Password** to protect the exported file.

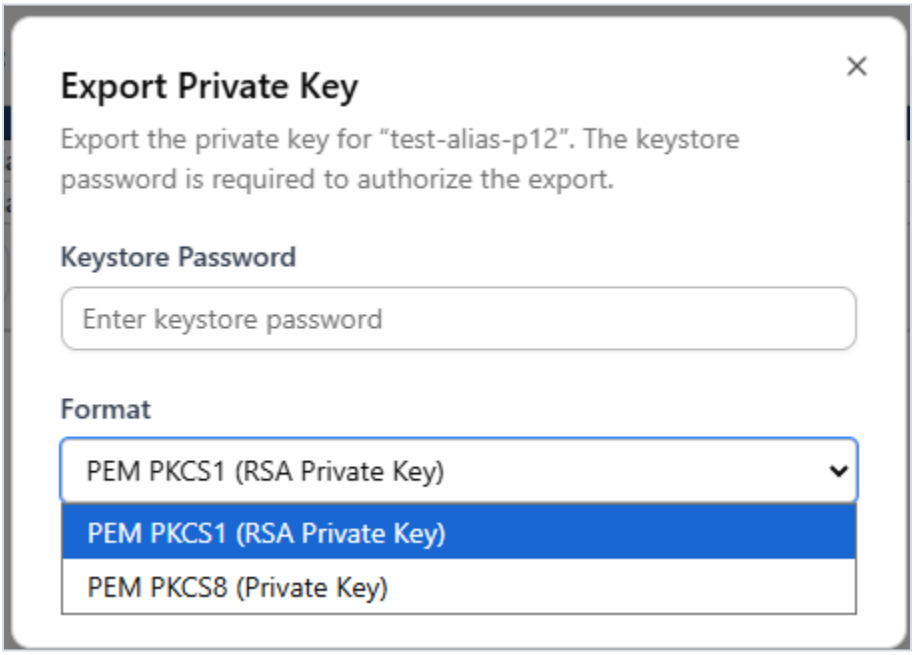
Click **Browse** to choose a save location, then click **Export**.



Export Certificate dialog

To export a private key, right-click the certificate, choose **Export Private Key**, enter the **Keystore Password**, select a **Format**, and click **Export**:

Format	Description
PEM PKCS1 (RSA Private Key)	Standard PEM format, widely supported.
PEM PKCS8 (Private Key)	Newer PEM format, commonly used with modern systems.



Export Private Key dialog

> Keep exported private keys secure. Never share one with unauthorized parties; it is used to decrypt data and authenticate your server. After three failed keystore-password attempts on an export, WebAdmin locks that export for 60 seconds before you can try again.

To export a public key, right-click the certificate, choose **Export Public Key**, click **Browse** to choose a save location, and click **Export**.

To delete a certificate or key store, select it, click **Delete**, and confirm. Deleting a key store removes every certificate it contains; this cannot be undone.

Trusted Certificates

This section manages certificates from external systems that your BridgeLink channels connect to or receive connections from. Adding a certificate here tells BridgeLink to trust that external system. Certificates live inside a **Trust Store**, selected from the **Trust Store** dropdown. Use the **Chain** and **Flat** buttons above the table to switch between a certificate-chain hierarchy view and a flat list.

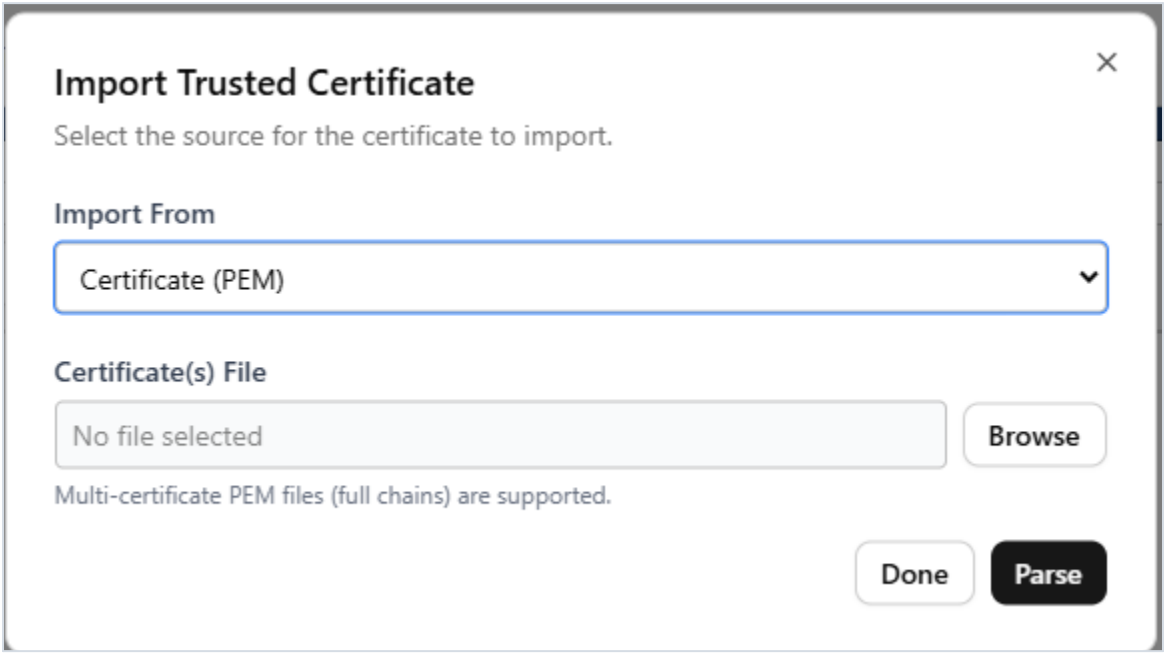
To import a trusted certificate:

1. In the **Trust Store** dropdown, select the trust store you want to use, or click the **+** button to create one first.
2. Click **Import**. The **Import Trusted Certificate** window opens.
3. In the **Import From** dropdown, choose the source:

Option	When to use
Certificate (PEM)	You have a standalone PEM certificate file, including multi-certificate PEM files containing a full chain.
Keystore (JKS/JCEKS/PKCS12)	You have a keystore file containing the external system's certificate.

If importing a PEM certificate:

1. Click **Browse** and select the `.pem` certificate file.
2. Click **Parse**. WebAdmin sorts the certificates it finds into a chain, root first.
3. Select a certificate from the list.
4. Enter a **Destination Alias** (lowercase, pre-filled from the certificate's Subject).
5. Click **Import**. Repeat steps 3 through 5 for any additional certificates in the chain.
6. Click **Done** when finished.

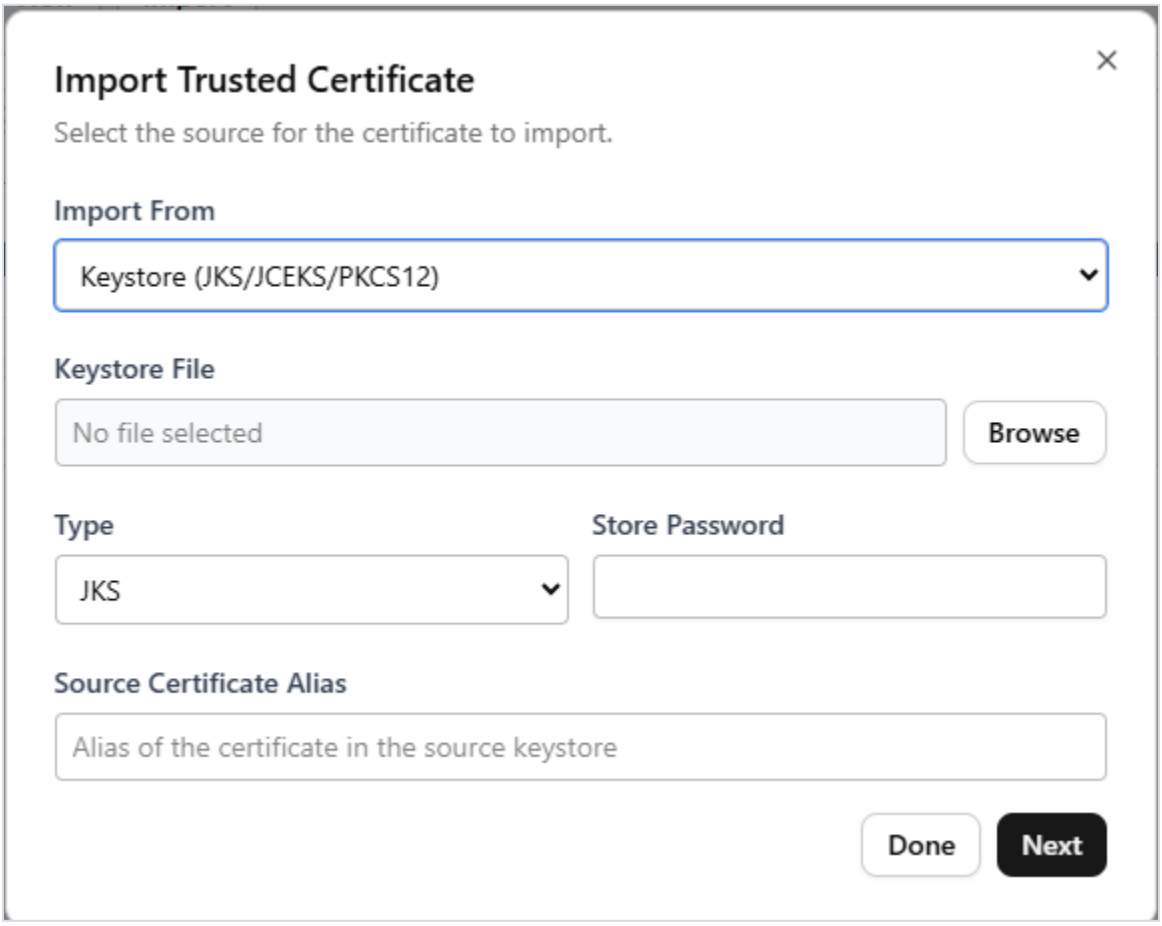


Import Trusted Certificate: PEM

If importing from a keystore file:

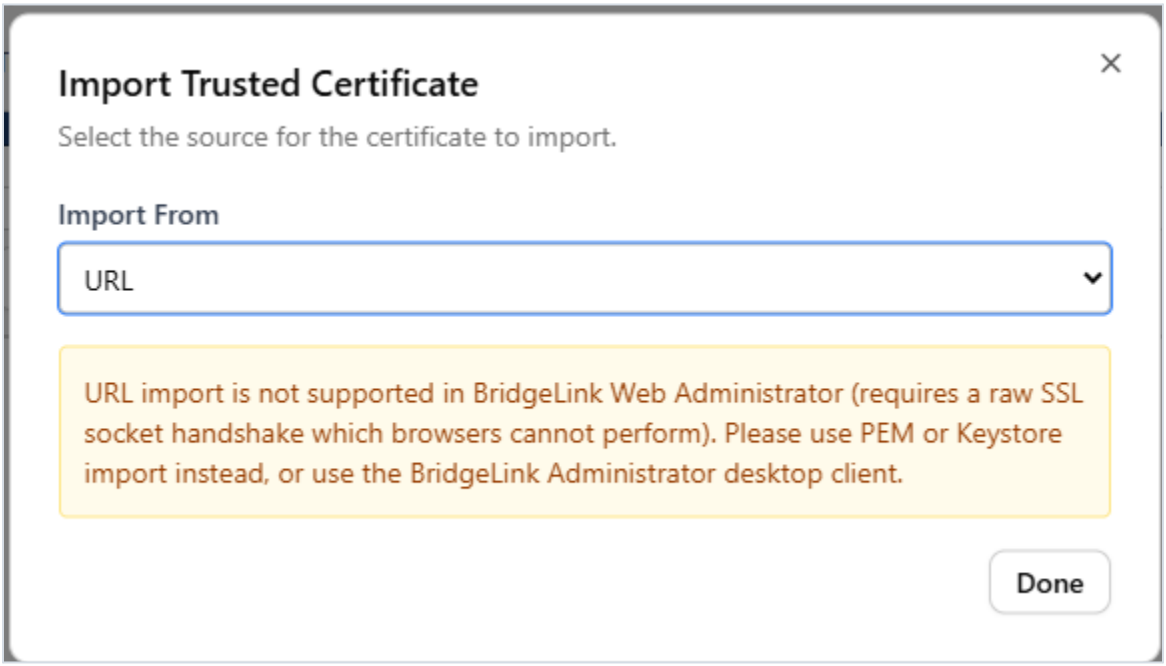
1. Click **Browse** and select the keystore file.
2. Select the **Type**: `JKS`, `JCEKS`, or `PKCS12`.
3. Enter the **Store Password**.
4. Enter the **Source Certificate Alias**.
5. Click **Next**.
6. Enter a **Destination Alias** (lowercase).

7. Click **Import**.



Import Trusted Certificate: Keystore

URL import is not supported. Selecting **URL** from the **Import From** dropdown shows a message explaining that fetching a certificate directly from a remote server requires a raw SSL socket handshake, which a browser cannot perform. Use **PEM** or **Keystore** import instead, or use the BridgeLink Administrator desktop client if you need to import directly from a URL.



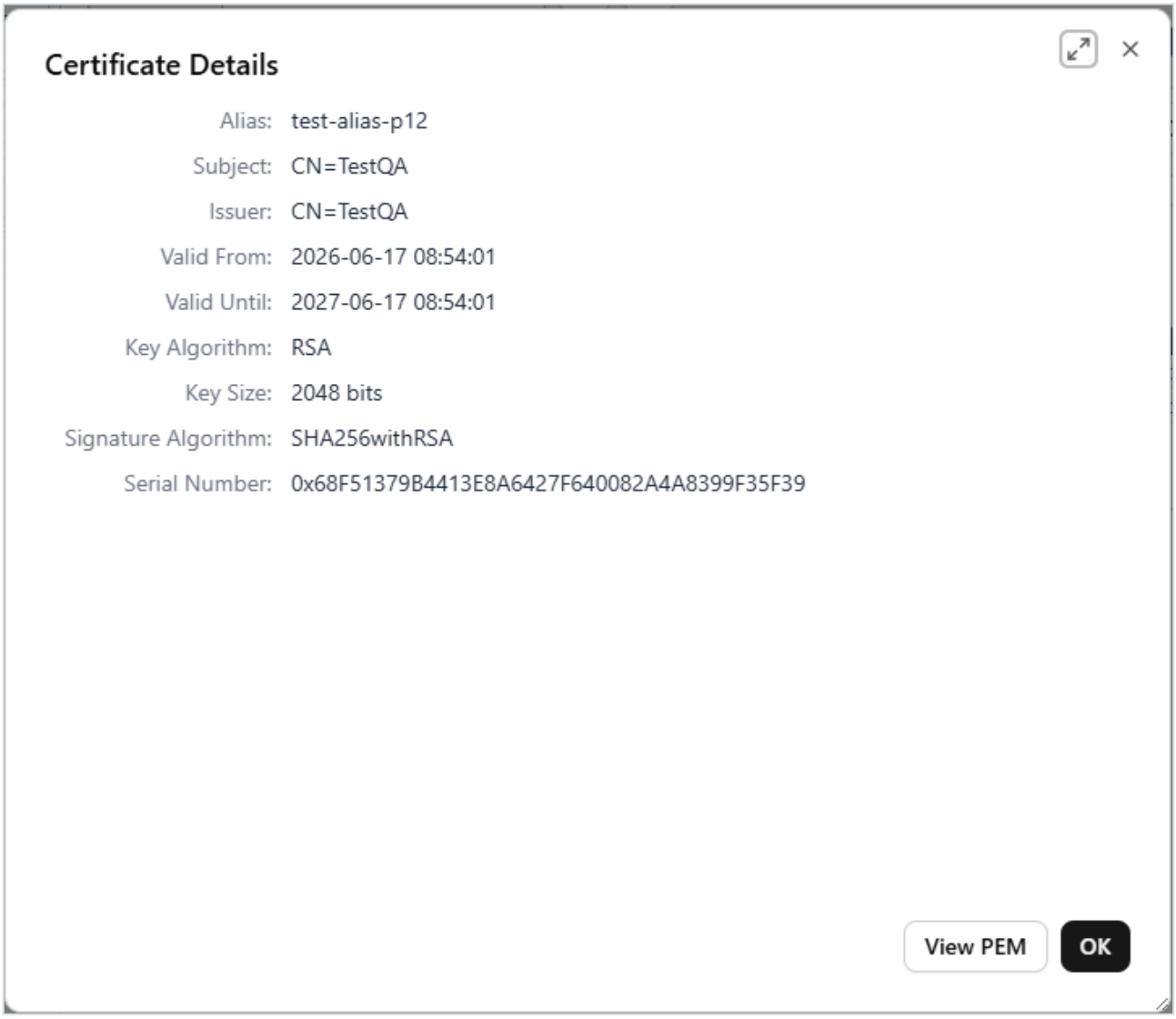
Import Trusted Certificate: URL is not supported in WebAdmin

The imported certificate appears in the **Trusted Certificates** table with its **Alias**, **Subject CN**, **Issuer CN**, and **Expiration Date**; the **Chain** view also shows a **Role** column (Root CA, Intermediate, or End-entity). Trusted certificates cannot be exported; they belong to external systems and do not contain a private key.

To delete a trusted certificate or trust store, select it, click **Delete**, and confirm.

Viewing Certificate Details

To view the full details of a certificate, right-click it in either table and select **View Details**.



Certificate Details window

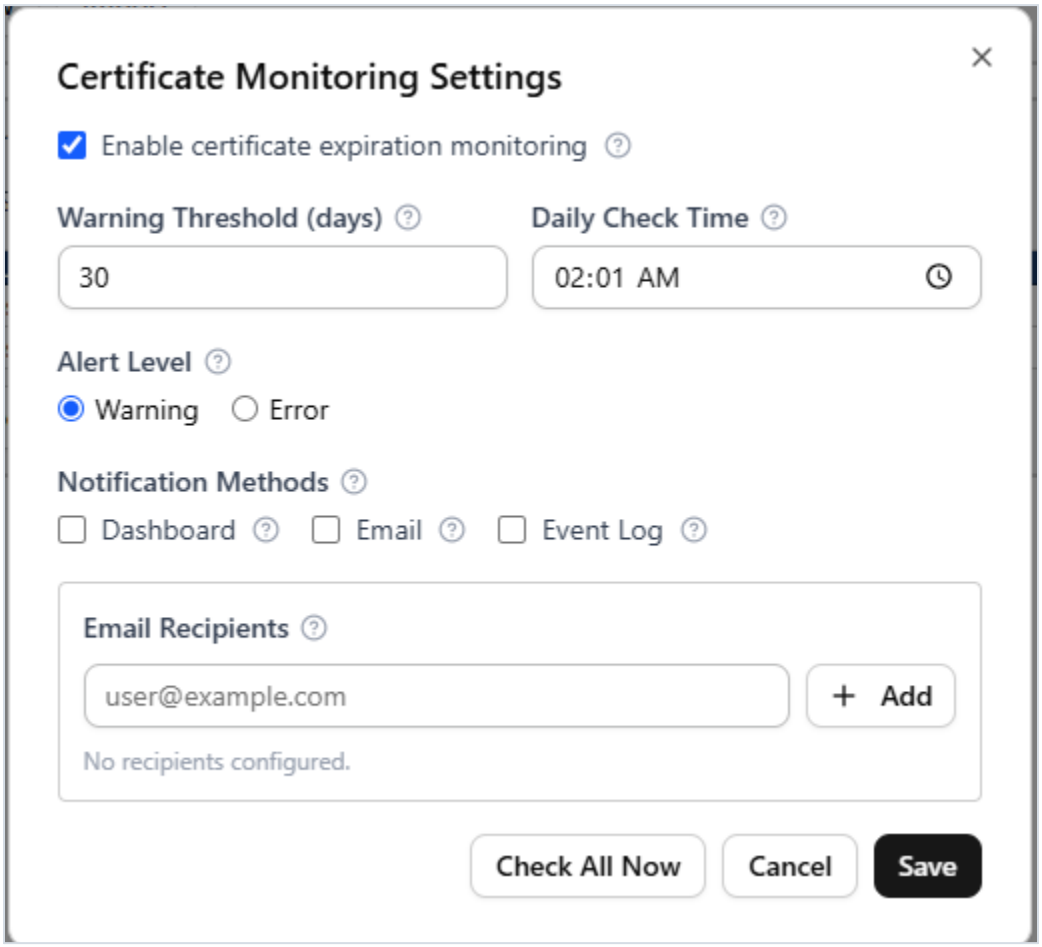
Field	Description
Alias	The friendly name identifying the certificate.
Subject	The entity the certificate belongs to, for example `CN=*.example.com`.
Issuer	The authority that issued the certificate.
Valid From	The date the certificate became valid.
Valid Until	The date the certificate expires. Shown in red if already expired.
Key Algorithm	The public key algorithm, for example `RSA`.
Key Size	The key size in bits.

Field	Description
Signature Algorithm	The algorithm used to sign the certificate, for example `SHA256withRSA`.
Serial Number	The certificate's unique serial number.
Subject Alternative Names	Additional DNS names or IP addresses the certificate is valid for, when present.

Click **View PEM** to display the certificate in PEM format, or **View Chain** (when the certificate has more than one certificate in its chain) to inspect each certificate in the hierarchy. Click **OK** to close the window.

Certificate Monitoring

To avoid unexpected SSL failures from expired certificates, BridgeLink can automatically monitor your certificates and alert you before they expire. Click **Monitoring** in the Certificate Manager toolbar to open **Certificate Monitoring Settings**.

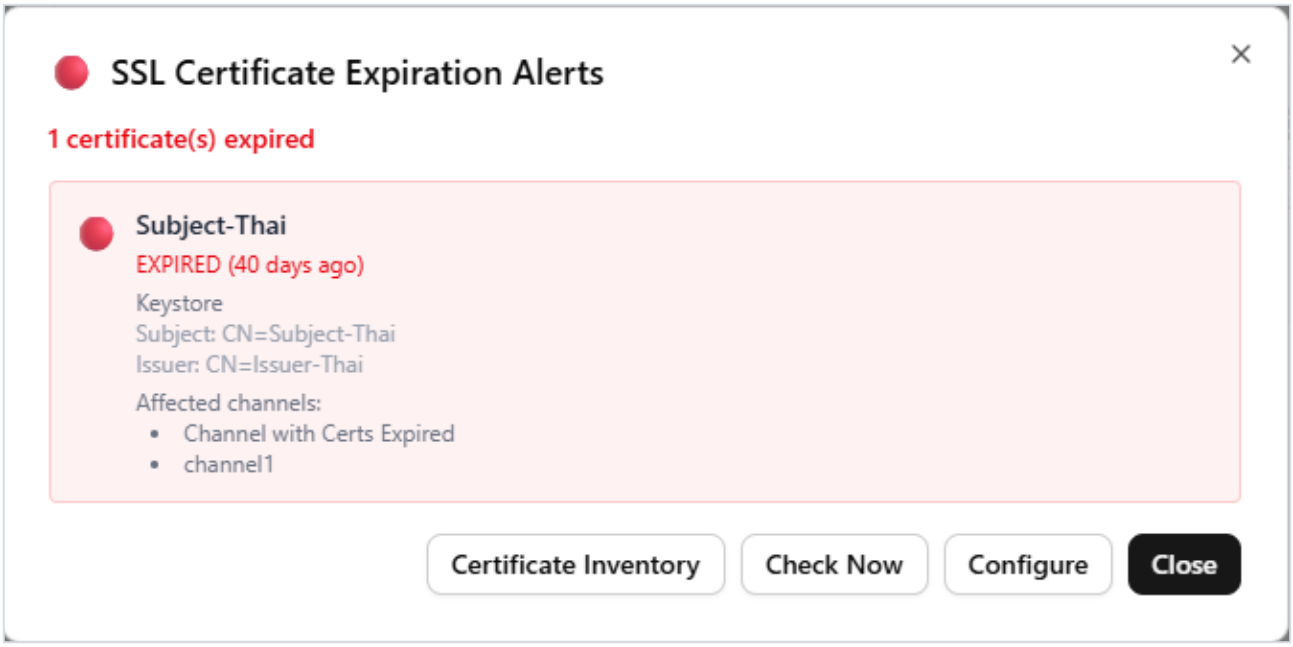


Certificate Monitoring Settings dialog

Setting	What it does	Default
Enable certificate expiration monitoring	Turns automatic monitoring on or off.	Enabled
Warning Threshold (days)	Certificates expiring within this many days are flagged as expiring soon.	`30`
Daily Check Time	The time of day (server time) BridgeLink checks certificates.	`02:00`
Alert Level	Logs alerts as **Warning** or **Error**.	Warning
Notification Methods	**Dashboard**, **Email**, and **Event Log** checkboxes controlling how you are notified.	Dashboard
Email Recipients	Addresses that receive expiration emails when **Email** is checked. Enter an address and click **Add**.	None configured

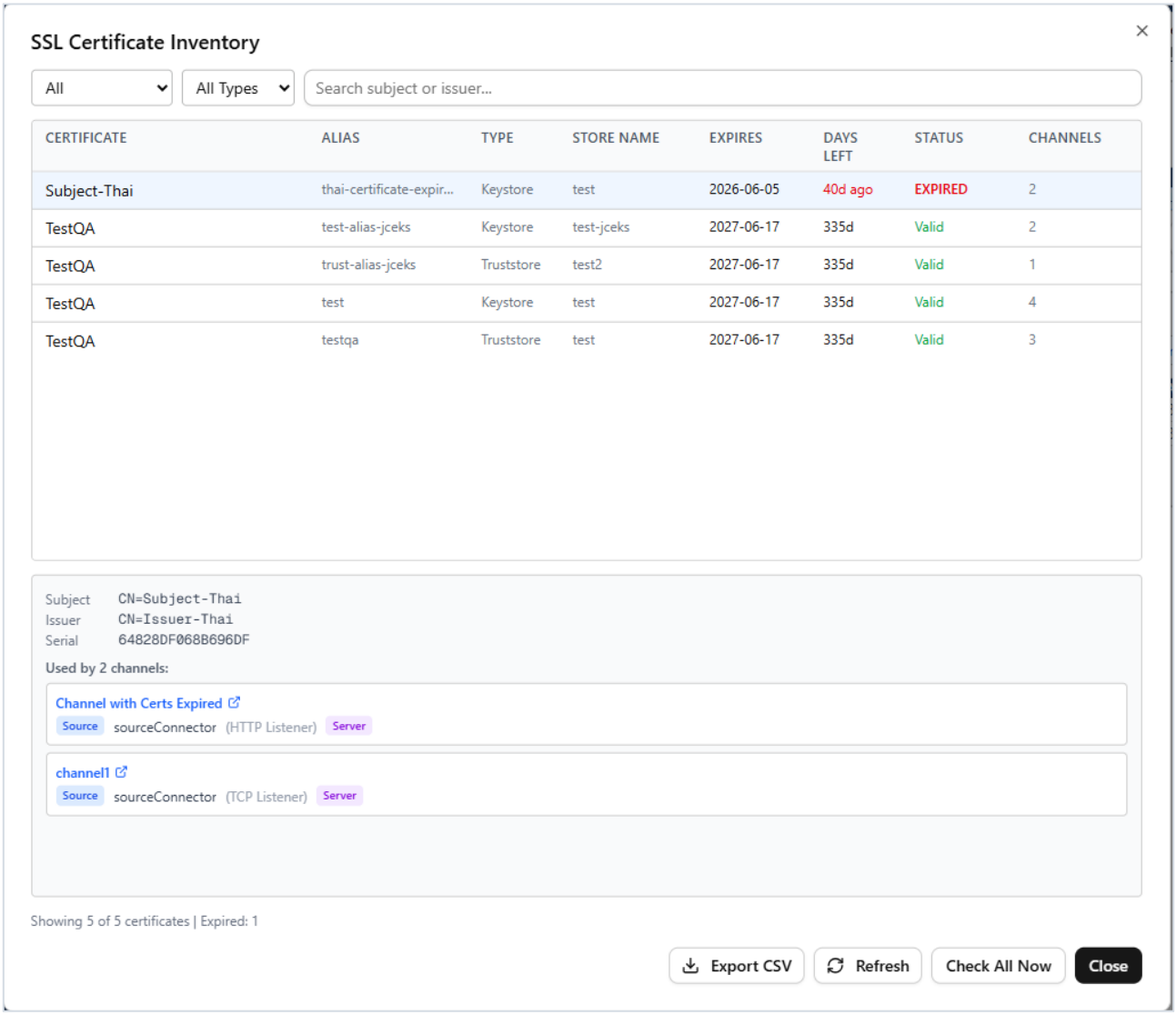
Once at least one recipient is added, a **Send Test Email** button appears so you can confirm delivery. Click **Check All Now** to run an immediate check on all certificates, or **Save** to apply your changes.

Click **Alerts** in the Certificate Manager toolbar to see certificates that are expiring soon or have already expired, along with the channels affected.



SSL Certificate Expiration Alerts window

Click **Certificate Inventory** in the Certificate Manager toolbar (or **Certificate Inventory** from the Alerts window) to see every certificate across every store, with its current status.



SSL Certificate Inventory window

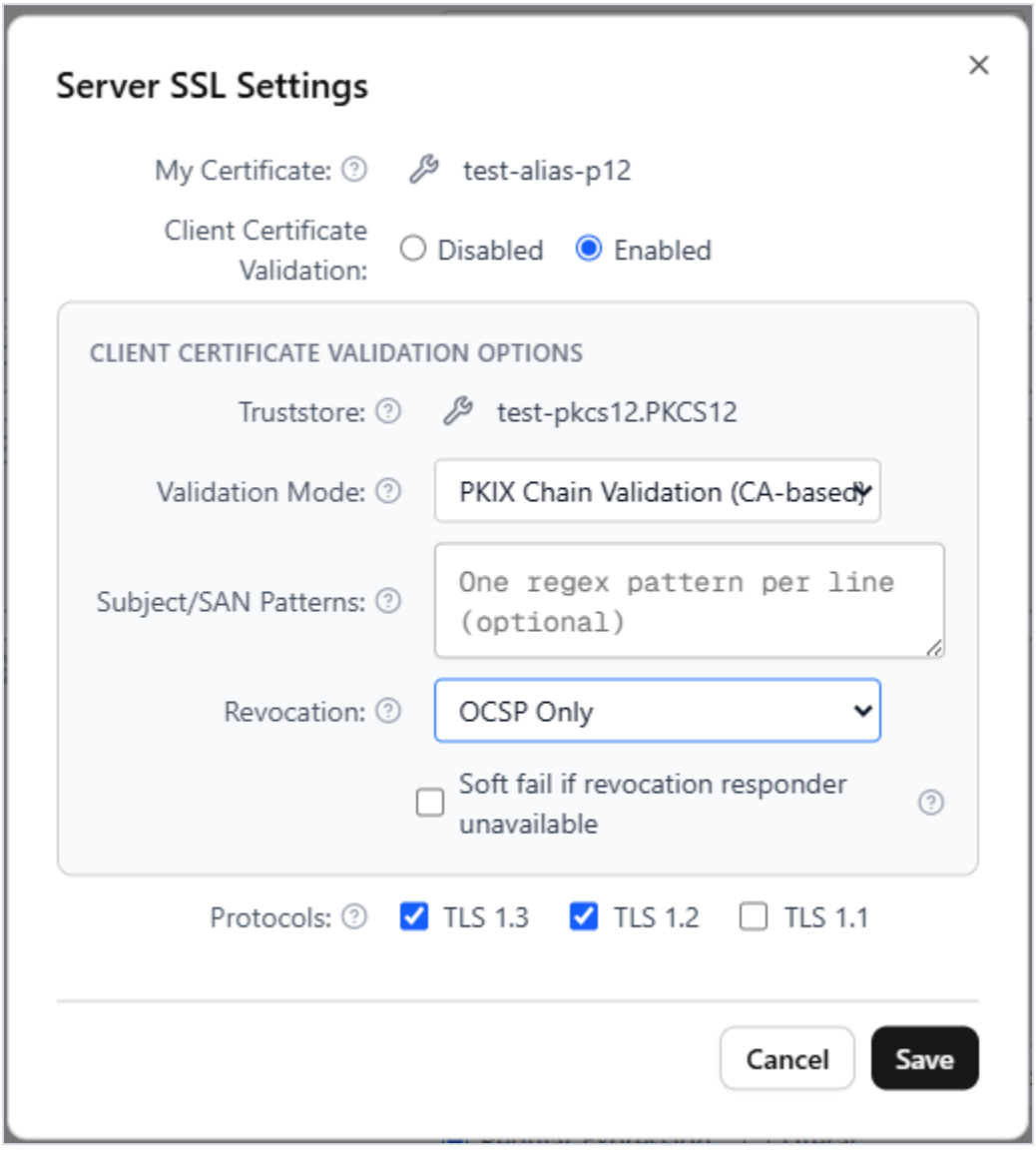
Column	Description
Certificate	The Common Name (CN) of the certificate.
Alias	The friendly name assigned to the certificate.
Type	Whether it is stored in a Keystore or Truststore.
Store Name	The name of the key store or trust store.
Expires	The expiration date.
Days Left	Days remaining; shown as `Nd ago` once expired.
Status	`Valid`, `Expiring Soon`, or `Expired`.
Channels	Number of channels currently using this certificate.

Use the status and type filters, or the search box, to narrow the list. Select a row to see its Subject, Issuer, Serial Number, and the channels using it, with a link to open each channel. Click **Export CSV** to download the full inventory, **Refresh** to reload it, or **Check All Now** to re-check every certificate.

Connector SSL Settings

HTTP Listener, TCP Listener, and Web Service Listener source connectors, and HTTP Sender, TCP Sender, and Web Service Sender destination connectors, each have a collapsible **SSL Settings** section in their connector settings. Set **Enable SSL** to **Yes** to turn on SSL/TLS for that connector; the configuration window opens automatically, and you can reopen it later with the settings (gear) icon.

Server SSL Settings (Listener connectors)



Server SSL Settings for a Listener connector

Setting	What it does	Default
My Certificate	The certificate this connector presents to connecting clients. Click the wrench icon to browse your key stores.	Not configured
Client Certificate Validation	**Disabled** accepts all clients without checking a certificate. **Enabled** requires clients to present a valid certificate.	Enabled

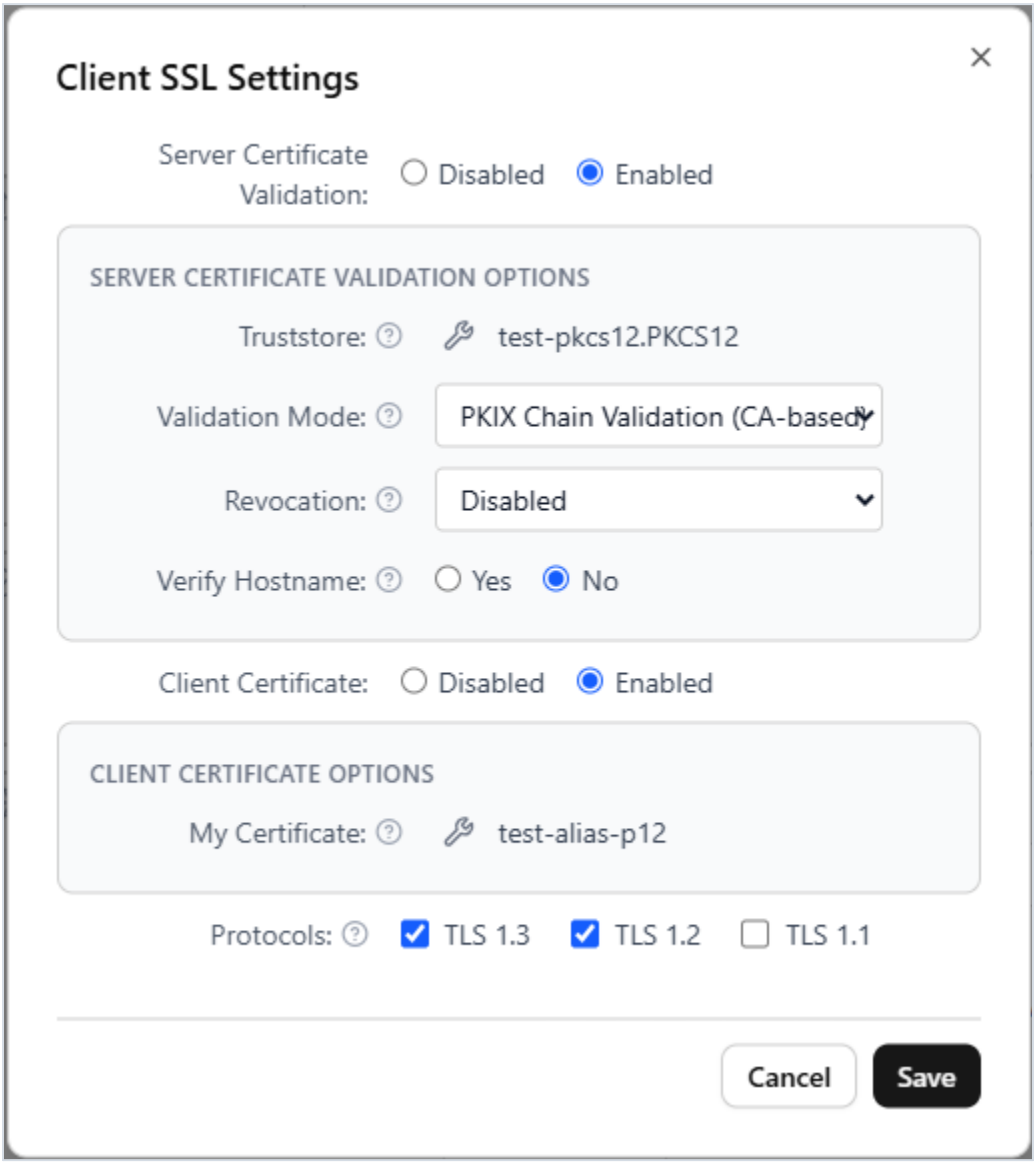
When **Client Certificate Validation** is **Enabled**, these additional options appear:

Setting	What it does	Default
Truststore	The trust store containing certificates of clients you want to allow.	Not configured
Validation Mode	**PKIX Chain Validation (CA-based)** validates against a CA chain. **Pinned Leaf Certificate** accepts only specific pinned certificates. **Trust All (No Validation)** skips validation and is not recommended for production.	`PKIX Chain Validation (CA-based)`
Pinned Leaf Pins	One SPKI SHA-256 fingerprint per line. Shown only when **Validation Mode** is **Pinned Leaf Certificate**.	Empty
Subject/SAN Patterns	Optional regex patterns, one per line, matched against the client certificate's Subject or SAN.	Empty (no filter)
Revocation	**Disabled**, **OCSP Only**, **CRL Only**, or **OCSP with CRL Fallback**.	`Disabled`
Soft fail if revocation responder unavailable	If checked, allows the connection when the OCSP/CRL responder cannot be reached. Shown when Revocation is not Disabled.	Unchecked

Under **Protocols**, choose which TLS versions this connector accepts: **TLS 1.3** and **TLS 1.2** are recommended and enabled by default. **TLS 1.1** is deprecated; enable it only if a legacy client requires it.

The configuration is identical across HTTP Listener, TCP Listener, and Web Service Listener connectors.

Client SSL Settings (Sender connectors)



Client SSL Settings for a Sender connector

Setting	What it does	Default
Server Certificate Validation	**Disabled** skips validation of the destination server's certificate. **Enabled** verifies it before connecting.	Enabled

When **Server Certificate Validation** is **Enabled**, the same **Truststore**, **Validation Mode**, **Revocation**, and **Soft fail** options described above apply, plus:

Setting	What it does	Default
Verify Hostname	**Yes** confirms the server certificate's hostname matches the destination. Not shown for TCP Sender, since TCP connections have no hostname to verify.	No

Below that, a separate **Client Certificate** control lets this connector present its own certificate for destinations that require mutual TLS:

Setting	What it does	Default
Client Certificate	**Enabled** presents a client certificate to the destination server during the handshake.	Disabled
My Certificate	The certificate presented to the destination server. Shown only when **Client Certificate** is **Enabled**.	Not configured

The same **Protocols** options (TLS 1.3, TLS 1.2, TLS 1.1) apply. The configuration is identical across HTTP Sender, TCP Sender, and Web Service Sender connectors, except that TCP Sender has no **Verify Hostname** option.

Using the SSL Settings Plugin

Using SSL Helper Functions in a JavaScript Transformer

In addition to connector-level SSL settings, the SSL Settings Plugin provides helper functions you can call from a channel's JavaScript transformer to make HTTPS calls with full SSL control. Open a JavaScript transformer, and in the **Reference** panel set **Category** to **SSL Helper Functions**. Two code templates are available: **HTTPS GET with SSLHelper** and **HTTPS POST with SSLHelper**.

Both templates use an `sslParams` map to configure SSL behavior:

Parameter	Required	Description
`truststoreUid`	Yes	UID of the trust store used to validate the server's certificate.
`validationMode`	No	`pkix` (default), `pinned_leaf`, or `trust_all`.
`pinnedLeafPins`	Conditional	SPKI SHA-256 hex fingerprints, one per certificate. Required when `validationMode` is `pinned_leaf`.
`revocationMode`	No	`disabled` (default), `ocsp_only`, `crl_only`, or `ocsp_with_crl_fallback`.
`softFail`	No	`true` allows the connection if the OCSP/CRL responder is unavailable. Default `false`.
`hostnameVerify`	No	`true` enforces strict hostname verification, recommended for production. Default `false`.
`clientCertEnabled`	No	`true` presents a client certificate for mutual TLS. Default `false`.
`keystoreUid`	Conditional	UID of the key store holding the client certificate. Required when `clientCertEnabled` is `true`.
`certAlias`	Conditional	Alias of the client certificate in the key store. Required when `clientCertEnabled` is `true`.
`tls13`	No	Enables TLS 1.3. Default `true`.
`tls12`	No	Enables TLS 1.2. Default `true`.
`tls11`	No	Enables TLS 1.1 (deprecated). Default `false`.

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

Parameter	Description
`connectTimeoutMs`	`Connection timeout in milliseconds. Default 15000.`

Parameter	Description
`readTimeoutMs`	`Read timeout in milliseconds. Default 30000.`
`throwOnHttpError`	`true throws an exception on an HTTP status of 300 or higher. Default false.`
`captureHeaders`	`true includes response headers in the result. Default false.`
`followRedirects`	`true follows HTTP redirects automatically. Default true.`

`SSLHelper.httpsGet()` and `SSLHelper.httpsPost()` return a map with these keys:

Key	Description
`status`	HTTP response status code, for example `200` or `404`.
`reason`	HTTP status reason phrase, for example `OK` or `Not Found`.
`body`	Response body as a string.
`headers`	Response headers. Populated only when `captureHeaders` is `true`.

```
var sslParams = new java.util.HashMap();
sslParams.put("truststoreUid", "my-truststore.PKCS12");
sslParams.put("validationMode", "pkix");
sslParams.put("revocationMode", "disabled");
sslParams.put("hostnameVerify", true);
sslParams.put("tls13", true);
sslParams.put("tls12", true);
sslParams.put("tls11", false);

var httpParams = new java.util.HashMap();
httpParams.put("connectTimeoutMs", 15000);
httpParams.put("readTimeoutMs", 30000);

var res = SSLHelper.httpsGet(
    "https://example.com:8443/api/resource",
    new java.util.HashMap(),
    sslParams,
    httpParams
);

if (res.get("status") >= 300) {
    throw "HTTP " + res.get("status") + " " + res.get("reason");
}
logger.info(String(res.get("body")));
```

> Set `hostnameVerify` to `true` in production environments. Set it to `false` only during development and testing.

Automating Certificate Management with the REST API

The SSL Settings Plugin exposes a REST API for managing key stores, trust stores, and certificates programmatically, useful for automation and scripting. All endpoints are under `/plugins/ssl` on your BridgeLink server. Sign in to WebAdmin, open your server's `/api` documentation page, and browse to the Plugin Services group to try the endpoints interactively.

Keystores

Method	Endpoint	Description
GET	`/keystores`	List all keystores.
POST	`/keystores`	Create a new keystore.
GET	`/keystores/{uid}`	Get a specific keystore by UID.
DELETE	`/keystores/{uid}`	Delete a specific keystore.
GET	`/keystores/{uid}/certificates`	List all certificates in a keystore.
POST	`/keystores/{uid}/certificates`	Create a new self-signed certificate in a keystore.
GET	`/keystores/{uid}/certificates/{alias}`	Get a specific certificate by alias.
DELETE	`/keystores/{uid}/certificates/{alias}`	Delete a certificate from a keystore.
POST	`/keystores/{uid}/certificates/import`	Import a private key and certificate chain into a keystore.
POST	`/keystores/{uid}/certificates/import-from-keystore`	Import a private key and certificate chain from another keystore file.
GET	`/keystores/{uid}/certificates/{alias}/export/chain`	Export the certificate chain in PEM format.
POST	`/keystores/{uid}/certificates/{alias}/export/certificate-private-key`	Export the certificate chain and private key together as JKS, JCEKS, or PKCS12.
POST	`/keystores/{uid}/certificates/{alias}/export/private-key`	Export the private key. Requires the keystore password.

Method	Endpoint	Description
GET	`/keystores/{uid}/certificates/{alias}/export/public-key`	Export the public key in PEM format.

Truststores

Method	Endpoint	Description
GET	`/truststores`	List all truststores.
POST	`/truststores`	Create a new truststore.
GET	`/truststores/{uid}`	Get a specific truststore by UID.
DELETE	`/truststores/{uid}`	Delete a specific truststore.
GET	`/truststores/{uid}/certificates`	List all trusted certificates in a truststore.
GET	`/truststores/{uid}/certificates/{alias}`	Get a specific trusted certificate by alias.
DELETE	`/truststores/{uid}/certificates/{alias}`	Delete a trusted certificate from a truststore.
GET	`/truststores/{uid}/certificates/{alias}/pem`	Get a trusted certificate in PEM format.
POST	`/truststores/{uid}/certificates/import-from-truststore`	Import a trusted certificate from another truststore.

Certificate Monitoring

Method	Endpoint	Description
GET	`/certificate-monitoring/config`	Get the certificate monitoring configuration.
POST	`/certificate-monitoring/config`	Set the certificate monitoring configuration.
POST	`/certificate-monitoring/check`	Trigger an immediate expiration check across all SSL-enabled channels.
GET	`/certificate-monitoring/alerts`	Get current certificate expiration alerts.

Method	Endpoint	Description
GET	`/certificate-monitoring/snapshots`	Get all certificate inventory snapshots.
GET	`/certificate-monitoring/last-scan-time`	Get the time of the last certificate scan.
POST	`/certificate-monitoring/test-email`	Send a test email notification.

Replace `{uid}` with the keystore or truststore UID, and `{alias}` with the certificate alias. You can find both values on the Certificate Manager screen or by calling the corresponding `GET` endpoints.

Troubleshooting

Symptom	Likely cause	What to do
An imported file is rejected with a message about a missing private key	You selected a certificate-only file instead of a private key file for a Private Key + Certificate import.	Select a `.pem` or `.key` file containing a `BEGIN PRIVATE KEY` block, or import the certificate alone under Trusted Certificates instead.
An export or import keeps failing and the button becomes disabled with a countdown	Three failed keystore-password attempts trigger a 60-second lockout on that export.	Wait for the countdown shown on the dialog, then re-enter the correct password.
An alias is rejected when creating or importing a certificate	Aliases must be lowercase and unique within the store.	Re-enter the alias in lowercase, or choose a different name if one already exists in that store.
A channel using SSL fails to connect after a certificate renewal	The connector still references the old certificate, or the new certificate has not been assigned.	Reselect the certificate on the connector's SSL Settings and confirm the alias and expiration date match the renewed certificate.
Client connections are rejected even though the client presents a valid certificate	Validation Mode, Subject/SAN Patterns, or Certificate Revocation are stricter than expected.	Review Validation Mode, Subject/SAN Patterns, and Certificate Revocation on the connector's SSL Settings. Temporarily setting Revocation to Disabled can help isolate the cause.
A destination connector cannot reach an older TLS-only endpoint	TLS 1.1 is disabled by default.	Enable TLS 1.1 under Protocols only if the destination cannot be upgraded; this is not recommended for regular use.
Selecting URL as an import source does nothing useful	URL import is not supported in WebAdmin.	Export the certificate from the source system as a PEM or keystore file and import that file instead, or use the BridgeLink Administrator desktop client.

Support

If you need help beyond this guide, contact Innovar Healthcare support at `support@innovarhealthcare.com`. Include your BridgeLink version, the SSL Settings plugin version (shown on the Extensions screen), and any relevant entries from the Server Log or SIEM related to the issue. Innovar Healthcare professional services can also help design and validate SSL/TLS configurations for complex or regulated environments.