



Connecting to Thredd Guide

Version: 1.1

20 May 2026

Publication number: CTG-1.1-5/20/2026

For the latest technical documentation, see the [Documentation Portal](#).

Thredd, 33 Kingsway, London, WC2B 6UF, UK

Support Email: occ@thredd.com

Support Phone: +44 (0) 203 740 9682

© Thredd 2026





Copyright

© Thredd 2026

The material contained in this guide is copyrighted and owned by Thredd Ltd together with any other intellectual property in such material.

Except for personal and non-commercial use, no part of this guide may be copied, republished, performed in public, broadcast, uploaded, transmitted, distributed, modified or dealt with in any manner at all, without the prior written permission of Thredd Ltd., and, then, only in such a way that the source and intellectual property rights are acknowledged.

To the maximum extent permitted by law, Thredd Ltd shall not be liable to any person or organisation, in any manner whatsoever from the use, construction or interpretation of, or the reliance upon, all or any of the information or materials contained in this guide.

The information in these materials is subject to change without notice and Thredd Ltd. assumes no responsibility for any errors.



About this guide

This guide is intended as a user guide, to provide information on connecting to Thredd services using Thredd's Secure Framework for the first time. It also provides information for existing clients that connect to Thredd via AWS and VPN.

Target audience

This guide is aimed at developers and system integrators who need to set up secure connections to Thredd services as a new client. It provides information for security consultants and Chief Information Security Officers (CISOs) who need to assess Thredd's security infrastructure.

Note: If you are an existing Thredd client and Thredd has contacted you about migrating your existing connectivity with Thredd services, see the dedicated Migration Guide.

What's changed?

If you want to find out what's changed since the previous release, see the [Document History](#) section.

Terminology used in this guide

Note: In this document, any terms in *italics* follow the standard (normative) terminology referenced in the OAuth 2.0 authorisation framework. For more information, see the [IETF OAuth 2.0 Authorization Framework](#).

How to use this guide

If you are new to Thredd and its security infrastructure, you should refer to the [Secure Framework](#) and [Setup Steps](#) to understand how to connect to Thredd. To learn about the set up steps that you need to complete, refer to the relevant sections.

Other documentation

Refer to the table below for a list of other relevant documents that you should use together with this guide.

Document	Description
Key Concepts Guide	Provides an introduction to card payments and how describes how Thredd supports your card program.
REST API via API Hub (V2.0)	Explains how to use the endpoints available in Thredd's latest REST APIs via API Hub (V2.0).
Cards API (V1.0)	Explains how to use the endpoints available in Thredd's legacy Cards API (V1.0), which uses REST.
Web Services Guide (SOAP)	Provides information on the available Thredd web services and fields in each web service.
EHI Guide	Provides details of the Thredd External Host Interface (EHI).
Thredd Portal Guide	Explains how you can use the Thredd Portal guide for managing cards.
Card Transaction System Guide	Describes how to submit card test transactions in the UAT environment.

Tip: For the latest technical documentation, see the [Documentation Portal](#).



Section 1: Getting Started

You should read this section if you are new to the Thredd Secure Framework and want to understand the basic principles, or want information on how to connect to Thredd's VPN and AWS options.

Topics covered in this section:

- [Thredd Secure Framework](#)
- [Setup steps for TLS and mTLS connections to Thredd](#)
- [VPN and AWS connections to Thredd](#)



Thredd Secure Framework

Thredd's Secure Framework is the combination of several components which enables clients to access Thredd's resources securely over TLS (Transport Level Security) or mTLS (Mutual Transport Level Security). *Thredd Portal* sits at the centre of this, enabling Admins to manage their organisation's applications, users, and credentials to access Thredd services through a single user interface and dashboard.

Thredd supports industry standards for authentication, identity and access management, and secure network and API-level connectivity.

Key features

Identity and Access Management through Thredd Portal

- Standards-based authentication – support for OpenID Connect (OIDC), SAML, and JWT.
- Single-Sign On (SSO), self-service onboarding, and token lifecycle management.
- Secure Access Control – role-based access (RBAC) and user management.

Connectivity Services

- Secure network and API-level connectivity via VPN, TLS and mTLS, and PKI and JWKS token validation.
- Certificate lifecycle management – request, revoke and renew certificates issued by [Thredd's Certificate Authority](#) through Thredd Portal.
- mTLS Termination for mTLS connections in an EHI mTLS setup.

Developer Tooling

- Access Thredd REST APIs via API Hub over TLS or mTLS.
- Test connectivity to Thredd REST APIs and API Hub using the Postman Collection in a UAT environment.

Identity and Access Management through Thredd Portal

Thredd provides a Software as a Service (SaaS) capability that acts as the *Identity Provider (IDP)* to authenticate user access to Thredd's interfaces through logging in to Thredd Portal. It also acts as an *OAuth OpenID Provider (OP)* for the registration and management of customer applications, generation and validation of access tokens, and for the enforcement of access control policies.

Thredd supports the following Client Authentication methods for connecting to the REST APIs:

- Client Secret – requires an access token from Thredd.
- Private Key JWT – requires an access token from Thredd, and a Signing Certificate to generate a JSON Web Token (JWT).

Thredd Certificate Authority through Thredd Portal

Thredd adopts a self-service approach that enables Admin users to independently request and manage certificates for their applications from Thredd's Certificate Authority (CA) through Thredd Portal.

Client applications require the following certificates to connect to services, depending on the configuration you choose:

- Signing Certificates – required for Private Key JWT client authentication, for creating a signed JWT for an access token.
- Transport Certificates – required for mTLS connections for establishing connections between resources.

Thredd Application	Certificates Required
REST API over TLS	Private Key JWT authentication requires a Signing Certificate. Client Secret authentication does not require a Signing Certificate.
REST API over mTLS	Transport Certificate for the mTLS connection, which you obtain from Thredd Portal. Private Key JWT authentication requires a Signing Certificate. Client Secret authentication does not require a Signing Certificate.
SOAP API	Transport Certificate, which you obtain from Thredd.
External Host Interface (EHI)	Server Certificate (which you obtain separately), Thredd EHI Trust Chain (that you download from Thredd Portal), and a Transport Certificate (which Thredd presents).
Thredd Portal	Certificates that are pre-installed by Thredd.



Set up Preparation and Steps

This guide helps you prepare and complete setting up your connection and identity and access management for Thredd services. This summary focuses on a typical setup with a TLS or mTLS connection to API Hub (REST APIs) via Thredd Secure Framework and Thredd Portal, and EHI.

Note: To access SOAP APIs, see [Creating Client Transport Certificates for SOAP APIs](#). For VPN setups, see [Connecting with AWS and VPN](#).

Thredd Portal

Thredd Portal is the main entry point to onboard to Thredd and access Thredd services. It provides support for multiple roles, including a Super Admin and at least one Organisation Admin. It enables a Super Admin to set up and manage the identity and access management for their organisation, applications, and users. Organisation Admins have similar privileges, but do not manage users.

Thredd Portal's access management features include:

- Organisation, application and identity management – users with a Super Admin or Organisation Admin role can create and manage applications, OAuth clients, SSO details, certificates, generate and validate access tokens, and more.
- User management – Super Admin users can invite users to onboard to Thredd and manage user roles, permissions, and access.
- Supports Multi-Factor Authentication (MFA), Single Sign-On, password-based sign-on, multiple identity providers, SAML and OIDC.

A Super Admin must first successfully set up their access and log in to Thredd Portal before configuring and testing access to Thredd's API Hub and REST APIs. The set up process comprises several stages, depending on the different services that you use.

Before you begin

You must provide the details of your organisation and the user that will operate as the Super Admin to Thredd. Your Thredd Implementation representative coordinates with you about this and registers these details with the Thredd platform.

As the Super Admin user, you can begin the process of connecting to Thredd's Secure Framework once you receive confirmation from Thredd.

Configuring your set up in a test environment first

For best practice, Thredd recommends that you first set up your organisation's access in the UAT (User Acceptance Test) environment. This enables you to test and verify that your organisation can use Thredd applications and API endpoints successfully.

Once you have verified that your setup and testing in the UAT environment is satisfactory, you can coordinate with Thredd and prepare to set up your Production environment. You will need to repeat all of these set up steps for your Production environment. This includes requesting new certificates and ensuring that environment-specification configurations are correct.

Summary of set up steps

Setting up access to Thredd starts with the Super Admin logging in to Thredd Portal and setting up their access to the Thredd platform. The Super Admin can then set up access for other users, but can choose to do this before or after setting up access to Thredd APIs and API Hub. Here is an overall summary of the different stages of the set up process – use the links to visit the step-by-step guidance for each stage.

Step 1: Set up the Super Admin in Thredd Portal

The Super Admin user will receive a welcome email inviting them to log in to Thredd Portal. They must set up their access to Thredd Portal before they can manage the access for their organisation, application and users.

As the Super Admin, you will need to add your organisation's identity provider details in Thredd Portal, enabling authentication using your own Identity Provider (IdP). If your organisation does not use an IdP, Thredd's own provision can act as the IdP.

You can choose one of the following methods for users to log in to Thredd services:

- Single Sign-On (SSO) – this involves adding an SSO configuration in Thredd Portal and inviting users via your identity provider. This is not required, but Thredd recommends it to help automate user onboarding and management.
- Email address and password – a Super Admin manually adds each user in Thredd Portal and using its invite feature to ask them to log in.

For an overview of Thredd Portal and the set up process, see [Thredd Portal overview](#).



Tip: This guidance outlines setting up the Super Admin for the first time and additional users. However, a Super Admin does not have to add users to Thredd Portal straight away. Instead, the Admin can choose to add the organisation's SSO details (if using SSO), create Client certificates, and test access to Thredd REST APIs first. A Super Admin can later return to this step to add the organisation's users.

Step 2: Create a new Application and OAuth Client in Thredd Portal

Before you can access Thredd's REST API, you must create an application and an OAuth Client in Thredd Portal to register it and receive a client ID. You can then request client credentials for your application and configure your application and REST interface (such as Postman) to provide them when making API calls. This allows Thredd to verify your identity and grant access.

See [Creating an Application and OAuth Client](#).

Step 3: Create credentials to access the REST API

You need to create the necessary credentials to access Thredd's REST APIs, and configure your client application to provide these credentials when making API calls.

Thredd offers two Client Authentication methods via its Authorisation Server. The credentials you require depend on the authentication method that you have chosen to use.

- Client Secret – requires an access token that you can request from Thredd's OAuth Token endpoint.
- Private Key JWT – requires an access token from Thredd, and Signing Certificate, which an Admin user can obtain from Thredd Portal, to create and register a private key. You must generate a Client Assertion, a JSON Web Token (JWT) that is signed by your private key.

If you are using an mTLS connection, your application must also present a Transport Certificate from Thredd. Thredd adopts a self-service approach, which allows an Admin user to independently request and manage certificates via Thredd Portal.

See [Connecting to the REST APIs](#).

Step 4: Configure the latest Postman collection to test Thredd's REST APIs via API Hub

You can access Thredd's API Hub and REST APIs over TLS or mTLS. Thredd provides separate Postman Collections for TLS and mTLS, which enable you to use Postman to test using the Thredd REST APIs in a UAT environment.

To use the collection, you need to configure Postman with the appropriate environment variables and credentials. Refer to the guides that matches how you are connecting to Thredd APIs:

- For TLS connections: [Using the Postman Collection to call REST APIs over TLS](#)
- For mTLS connections: [Using the Postman Collection to call REST APIs over mTLS](#)

Step 5: Set up access to Thredd Portal for your organisation's users

A Super Admin must invite your organisation's users to log in to Thredd applications, using either:

- Single Sign-On (SSO) – this involves inviting users to log in to Thredd Portal via your identity provider, providing you have set up SSO in Thredd Portal. Make sure that each user receives an email inviting them to activate their account. Once a user has logged in to Thredd Portal, it registers them in the system, and a Super Admin can assign the appropriate roles to the user, completing the user's profile.
- Email address and password – this involves manually adding each user, assigning an appropriate role, and sending an invite to them via Thredd Portal using its built-in tool. The user receives an email inviting them to activate their account, set a password and log in to Thredd Portal. Once complete, the user can log in to other Thredd applications.

For more information, see [Connecting to Thredd Portal](#).

Step 6: Set up EHI and other Thredd applications

Secure connections are required to access the following Thredd applications:

- External Host Interface (EHI)
- SOAP API (if relevant to you)



External Host Interface (EHI)

The External Host Interface (EHI) offers a way to exchange transactional data between the Thredd processing system and the Program Manager's externally-hosted systems. All transaction data processed by Thredd is transferred to the external host system via EHI in real time.

EHI provides two main functions:

- a real-time transaction notification data feed
- payment authorisation control

You can use EHI with either a TLS or mTLS connection. Follow the guidance that matches the connection method you will use for the EHI.

Set up summary for EHI with a TLS connection

1. Obtain and install a Server certificate from the Certificate Authority that you want to use.
2. Log in to Thredd Portal and obtain the details of Thredd's JWKS endpoint.
3. Integrate with the JWKS endpoint.
4. Implement signature validation logic.
5. Test your EHI endpoint for TLS communication.
6. Inform Thredd that you are ready to use the EHI application.

For more information, see [Setting up EHI for TLS connections and signed payloads](#).

Set up summary for EHI with an mTLS connection

1. Obtain a Server certificate from the Certificate Authority that you want to use. Install the Server and Client Certificates.
2. Download and install the EHI Trust Chain from Thredd, which contains Root Certificates and Issuing Certificates from Thredd.
3. Store the Root and Issuing Certificates on your mTLS termination point.
4. Ensure that your mTLS termination point has access to Thredd's Online Certificate Status Protocol (OCSP) responder.
5. Test the Client and Server Certificates on your EHI endpoint for mTLS communication.
6. Provide the EHI endpoint details to Thredd and inform Thredd that you are ready to use the EHI application.

For more information, see [Setting up EHI for mTLS connections](#).

Tip: Once you have set up your TLS or mTLS connection for the EHI, see the [EHI Guide \(JSON version\)](#) or [EHI Guide \(XML version\)](#).

SOAP API

Thredd's SOAP APIs are secured using mTLS. If you are using the SOAP APIs, you will need to create Transport Certificates. For more information, see [Creating Client Transport Certificates for SOAP APIs](#). Only refer to this information if Thredd has confirmed that you will use the SOAP APIs; otherwise, you can ignore this information.

Other services

Services such as Fraud Transaction Monitoring and 3D-Secure do not require you to set up secure connections via Thredd Secure Framework.



Connecting with AWS and VPN

Here, you can learn how to access the Thredd AWS environments if you will connect to Thredd this way. Thredd services are hosted in AWS.

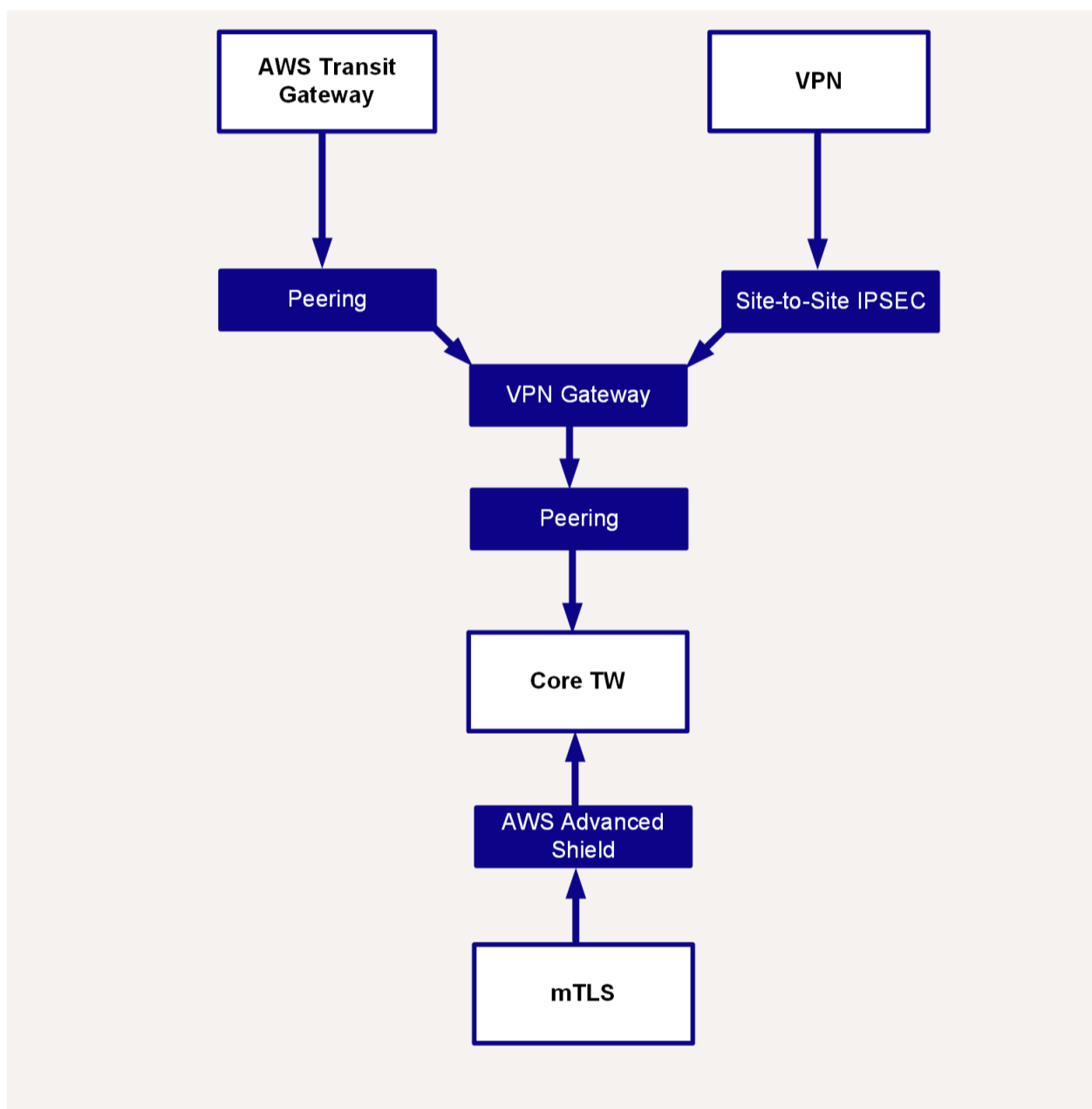
Note: Only refer to this information Thredd has confirmed that you will use it; otherwise, you can ignore this information.

There are three supported integration options for clients:

- **IPSec Site-to-Site VPN** for clients using AWS and other Cloud and on-premise infrastructure.
- **AWS Transit Gateway Peering** for clients that host their Infrastructure in AWS.
- **Mutual TLS (mTLS)** for certificate-based authentication.

Note: For a VPN setup, Thredd provides implementation guides for connecting to Thredd. For AWS, refer to the [AWS documentation](#).

See the following diagram of the flow for each option.





Connecting using VPN

IMPORTANT: It is the responsibility of the client to ensure that any local changes are communicated to Thredd to maintain connectivity.

Internet Protocol Security (IPsec) VPN connections take place over public networks, but the data exchanged over the VPN is still private because it is encrypted. VPNs make it possible to securely access and exchange confidential data over shared network infrastructure. In this instance, the public Internet.

IPsec is a framework of open standards to ensure private and secure communications over Internet Protocol (IP) networks. Encapsulating Security Payload (ESP) and Authentication Header (AH) are the two IPsec security protocols used to provide these security services.

Thredd VPN Setup Information

Thredd enables a private connection from AWS for each client that sets up two VPNs per site. You must complete the Thredd VPN Connectivity Setup Form and share it with Thredd. When Thredd has received this, it sends you a configuration template for your VPN product.

Thredd typically only supports static routing for VPNs. However, in certain circumstances, Border Gateway Protocol (BGP) routing can be implemented. For AWS configurations, Thredd provides Transit Gateway (TGW) peering to allow you to connect directly to our environment and minimise latency. You must include the TGW ID when completing the Thredd VPN Connectivity Setup Form.

When a subnet has been provided, Thredd confirms that it is available. Note the following:

- An EHI endpoint must be a subnet.
- The maximum subnet is /22. Thredd recommends using /24.
- Subnet /20 is not permitted by Thredd.
- All services run through the same Thredd tunnel.

Where two VPNs are required for production access, you must split them as follows:

- Web Services and EHI
- Thredd Portal



Connecting using AWS

IMPORTANT: It is the responsibility of the client to ensure that any local changes are communicated to Thredd to maintain connectivity.

AWS Transit Gateway Peering is for clients that host their Infrastructure in Amazon Web Services (AWS). Internet Protocol Security (IPsec) is a framework of open standards to ensure private and secure communications over Internet Protocol (IP) networks. Encapsulating Security Payload (ESP) and Authentication Header (AH) are the two IPsec security protocols used to provide these security services.

IPsec VPN connections take place over public networks, but the data exchanged over the VPN is still private because it is encrypted. VPNs make it possible to securely access and exchange confidential data over shared network infrastructure. In this instance, the public Internet.

Thredd AWS Setup Information

Thredd enables a private connection from AWS for each client and sets up a transit gateway attachment per site. You must complete the Thredd VPN Connectivity Setup Form and share it with Thredd. When Thredd has received this, Thredd sends you a configuration template for your VPN product.

Thredd only supports static routing for AWS Transit Gateway (TGW) attachments. When a subnet has been provided, Thredd confirms that it is available. Note the following:

- An EHI endpoint must be a subnet.
- The maximum subnet is /22. Thredd recommends using /24.
- Subnet /20 is not permitted by Thredd.
- All services run through the same Thredd tunnel.

Where two attachments are required for production access you must split them as follows:

- Web Services and EHI
- Thredd Portal



Mutual TLS explained

Mutual Transport Layer Security (Mutual TLS, mTLS) is an enhanced security protocol that requires a Client and Server to verify each other's identity before establishing a secure communication channel. This digital handshake requires both parties to provide their official identity.

What is Transport Layer Security (TLS) and how is mTLS different?

Transport Layer Security (TLS) is the standard internet protocol that ensures private, secure communication. For example, it enables HTTPS, where HTTP essentially runs over TLS, to secure the websites that people visit daily. TLS uses one-way authentication, where only the Server presents a certificate to the Client to prove its identity.

With *Mutual TLS* (mTLS), this process is reciprocal, as the 'Mutual' part of its name implies. mTLS requires two-way authentication, which means that both the Client and the Server must present a valid digital certificate that allows them to verify each other's identities.

Feature	TLS (one-way)	Mutual TLS (two-way)
Server Authentication	Required (the Server proves its identity to the Client)	Required
Client Authentication	Optional (usually none)	Required (the Client proves its identity to the Server)
Verification method	The Server uses a Server Certificate	The Client and the Server use Certificates

How mTLS works using a two-way verification process

The mTLS process involves four key steps:

1. Client request: A Client (your application) attempts to connect to the Server.
2. Server challenge and authentication: The Server sends its digital certificate to the Client and simultaneously requests the Client's Transport Certificate.
3. Client certificate presentation: The Client sends its unique Transport Certificate and a digital signature back to the Server.
4. Mutual verification:
 1. The Client verifies the Server's certificate.
 2. The Server verifies the Client's Transport Certificate, checking its validity, issuing authority, and integrity.
 3. If both checks pass, the Server and Client establish a TLS connection. If a check fails, the connection is immediately terminated.

Key benefits

By implementing mTLS, Thredd provides a significantly higher level of assurance and security for clients, and their data and services:

- Zero Trust Architecture (ZTA): mTLS is foundational to a Zero Trust security model. It ensures that no device or application is trusted by default, regardless of its location (inside or outside the network).
- Stronger authentication: It eliminates reliance on API keys alone. Your Transport Certificate is a unique, cryptographically-secured identity that is much harder to forge or steal.
- Protection against Man-In-The-Middle (MITM) attacks: The mutual verification process makes it virtually impossible for an unauthorised third party to impersonate either the client or the server to intercept data.

Transport Certificates to connect to REST APIs via mTLS

For Thredd's systems to securely authenticate your application, you must provide a valid Transport Certificate when using mTLS. This certificate is issued by Thredd's internal Certificate Authority and acts as the digital passport for your client application.

The Transport Certificate is mandatory for mTLS and is used solely for establishing the secure connection. Without a valid, signed Transport Certificate, your application is unable to connect to Thredd's secure endpoints. You will also require a Signing Certificate.

To learn more, see [Requesting certificates for applications using mTLS](#).



Using the EHI with mTLS

In an EHI (External Host Interface and mTLS) setup, your role of your organisation is reversed. Instead, your organisation acts as the *Server* and Thredd is the *Client*. To configure an EHI setup, you must download and install Thredd's Trust Chain on your EHI server. You do not need to request a Transport Certificate from Thredd, but you do need to obtain a Server certificate from a Certificate of Authority vendor, such as Verizon or Digicert or Amazon Web Services.

To learn more, see [Setting up EHI for mTLS connections](#).



Section 2: Thredd Portal

You should read this section to understand the setup steps Thredd Portal.

Topics covered in this section:

- [Thredd Portal overview](#)
- [Setting up Single Sign-On access and users to Thredd applications](#)
- [Setting up password logins to Thredd applications](#)
- [Understanding Roles](#)
- [Creating an Application and OAuth Client](#)
- [Understanding Scopes](#)



Thredd Portal overview

Thredd Portal is the main entry point to Thredd services and is the main component that is related to the API Hub; before you can use the API Hub, your organisation must be set up on Thredd Portal and Thredd Secure Framework. These combine several components that enable secure access to Thredd's resources, using a common identity store.

Thredd Portal provides a number of features, including:

- Organisation and application management – Super Admin and Organisation Admin users. Manage access for your organisation's applications, edit/add SSO details, create and manage certificates, generate and validate access tokens and certificates, and more.
- User management – Super Admin users. Invite users to onboard to Thredd and manage user roles, permissions, and access, including the option to deactivate and reactivate users.
- Cards and transaction management, and other related functionality – for users with the appropriate roles.
- Access to Webhook management in Thredd Portal – for users with the Developer role.

Tip: You can learn more about the cards and transaction management features of Thredd Portal in [Thredd Portal Guide: Getting Started](#).

Identity and Access Management

Thredd Portal provides identity and access management functionality at an organisation, application, and user level. Thredd Portal functions as a *Confidential Client*, where Thredd's own application infrastructure undertakes authentication and authorisation activity on behalf of the user.

Once a user is registered and has logged in to Thredd Portal, they can access other Thredd services. Thredd Portal also operates behind-the-scenes as an authentication server, enabling a single sign-on journey and access to Thredd's REST APIs.

Thredd Portal also provides access to Thredd's Certificate Authority for setting up and managing certificates to connect to Thredd services, including the REST APIs via API Hub.

To learn more, see [Thredd Secure Framework](#).

Connecting to Thredd Portal

Thredd Portal provides support for multiple roles, with Admin access available to a Super Admin and at least one Organisation Admin. A Super Admin user can set up and manage the settings for your organisation's applications, certificates, SSO configuration, and users through Thredd Portal. Organisation Admin users have a similar level of access, except they do not manage users.

This guidance focuses on the initial set up by a user with the Super Admin role in Thredd Portal.

You can choose one of the following methods for users to log in to Thredd services:

- Single Sign-On (SSO) – this involves adding an SSO configuration for SAML or OIDC in Thredd Portal and inviting users via your identity provider. This is not required, but Thredd recommends it to help automate user onboarding and management.
- Email address and password – this involves manually adding details for each user in Thredd Portal and sending an invite using its built-in invitation feature.

About Single Sign-On access (optional)

Thredd's Secure Framework allows you to optionally set up SSO, using *SAML* (Security Assertion Markup Language) or *OIDC* (OpenID Connect), to access Thredd services, for example Thredd Portal. This not mandatory but Thredd recommends that your users log in using SSO instead of using a password because it offers the following benefits:

- An enhanced user experience for users as it removes the hassle of remembering passwords.
- Companies to save time on maintenance.
- Reductions in overheads when managing accounts.

This enables a Single Sign-On journey by linking your IdP with Thredd's own provision. If you do not use an IdP, Thredd can act as the IdP.

This Single Sign-On journey is involved with:

- Accessing the organisation and user management features of Thredd Portal for Super Admin users, such as creating certificates with Thredd's CA, and the user management functionality to set up access for other users.
- Accessing the card and transaction management features of Thredd Portal.
- Managing access behind-the-scenes to the REST API.



Setting up access to Thredd Portal for the first time

A Super Admin user at your organisation must first register and log in to Thredd Portal before other users within your organisation can access Thredd Portal and other services.

You can begin this process when Thredd has registered your organisation and your Super Admin user, and has sent an email inviting them to log in to Thredd Portal.

The set up guidance outlines the full process of setting up the Super Admin for the first time and setting up additional users. However, a Super Admin does not have to invite/add users to register with Thredd Portal straight away. Instead, the Admin can choose to add the organisation's SSO details (if using SSO), create Client certificates, and test access to Thredd REST APIs and API Hub first. A Super Admin can later return to this step to set up the SSO or password-based access for the organisation's users.

Set up guides

To learn more about setting up access for your organisation, applications and users in Thredd Portal, see:

- [Setting up Single Sign-On access and users on the Thredd platform](#)
- [Setting up password-based access to Thredd Portal](#)
- [Understanding Roles](#)
- [Creating an Application and OAuth Client](#)
- [Understanding Scopes](#)

If your organisation will use Single Sign-On to access Thredd services, the following guides will help you to configure your SSO provider:

- [Configuring SSO with Okta \(SAML\)](#)
- [Configuring SSO with Google \(SAML\)](#)
- [Configuring SSO with Okta \(OIDC\)](#)



Setting up Single Sign-On access to Thredd applications

Using Single Sign-On (SSO) enables you to automate user onboarding; you can invite users to log in to Thredd Portal URL and automatically create their accounts. This eliminates the need to manually add and invite users in Thredd Portal.

A Super Admin can set up and manage the settings for your organisation's applications, certificates, SSO configuration, and users through Thredd Portal. An Organisation Admin also has these permissions, except user access.

This guidance assumes that you are the Super Admin and need to log in to Thredd Portal for the first time. You can set up Single Sign-On (SSO) when Thredd sends an email to the Super Admin user notifying them that it is ready for your organisation to do this.

Note: For best practice, Thredd recommends that you first set up your organisation's access in the test (UAT) environment, so that you can verify that your organisation can use Thredd applications and API endpoints successfully. You can then set up users later.

Prerequisites

- You must have the Super Admin role for your organisation in Thredd Portal.
- You must have received confirmation from Thredd that your organisation is ready to set up SSO.
- You must know your SSO protocol details to configure SAML (Security Assertion Markup Language) or OIDC (OpenID Connect).

Step 1: Log in to Thredd Portal for the first time

When Thredd has completed the initial step of registering your organisation and Super Admin user, it sends two emails to the Super Admin:

- An invitation containing a link to log in to Thredd Portal and set up your access.
- A separate email that contains your temporary password that you must use to log in to Thredd Portal.

You need to refer to both emails during the following process to log in to Thredd Portal for the first time.

1. Click on the link in the email for **Log in to Thredd Portal**. This directs you to the initial access screen.
2. Enter the temporary password from the other email; this is the only password that you can use to gain access initially. Copy the temporary password and paste it into the field under **Temporary password**. Then select **Next**.

3. On the next screen, you must create a new password. The password must be at least eight characters in length. Enter and confirm the new password in the separate fields, and then select the **Save and Continue** button. You can now start the SSO Setup.



Note: If the link does not work or you need support, use the link at the bottom of the emails to contact Thredd. Do not reply to the email. It is system-generated and sent from a notification-only address that cannot accept incoming emails.

Step 2: Confirm your SSO setup in Thredd Portal

Next, you can select and configure your preferred method for Single Sign-On access to the Thredd platform.

Tip: You can also check and manage SSO settings via **System Admin > SSO Configuration** in Thredd Portal.

- In the SSO setup screen, provide the details for your organisation and SSO protocol. This allows Thredd to automatically verify the identity of your organisation and its users when they access the Thredd platform. Select either **SAML** or **OIDC**.
 - SAML** – confirm the metadata delivery mode and any required information, either:
 - Fetch from URL:** Provide the metadata delivery URL – locate it in your organisation's identity provider settings.
 - File or RAW XML:** If the information is not already saved here, upload the file. You can upload only one file.

The figure displays two side-by-side screenshots of the SSO configuration interface. Both screenshots show the 'What is your provider?' section with 'SAML' selected. The left screenshot shows 'What is your chosen metadata delivery mode?' with 'Fetch from URL' selected, and a text input field for the URL with the placeholder text 'Enter the SAML identity provider's metadata URL'. The right screenshot shows 'What is your chosen metadata delivery mode?' with 'File or RAW XML' selected, and a file upload area with a folder icon, the text 'Click or drag file to this area to upload', 'Supports upload of a single file only.', and 'Maximum file size: 10 MB'. Both screenshots have a 'Continue' button at the bottom.

Figure: SAML using Fetch from URL, and SAML using File or RAW XML

- OIDC** – confirm the **Issuer ID**, **Client ID**, and **Authentication Method**. For the **Authentication Method**, specify one of the following:
 - Client Secret:** Provide the **Client Secret** – this method involves sharing a symmetric password between the Client and Server.
 - Private Key JWT:** Provide the **Client Private Key** (PEM format), **Algorithm**, **Expires in** (the period of time in hours, minutes and seconds that the client assertion is valid for), and the **Redirect URL**. This method is an asymmetric method where the client signs a token with a private key, which the Server validates using a registered public key.



What is your provider?

SAML OIDC

Issuer ID

Client ID

Authentication Method

Client Secret

[Continue](#)

What is your provider?

SAML OIDC

Issuer ID

Client ID

Authentication Method

Client Private Key

The PEM-formatted private key used to sign the client assertion.

Algorithm

The algorithm type used to sign the client assertion.

Expires in

Defines the period of time that the client assertion is valid for.

Redirect URL

Figure: OIDC using Client Secret, and OIDC using Private Key JWT

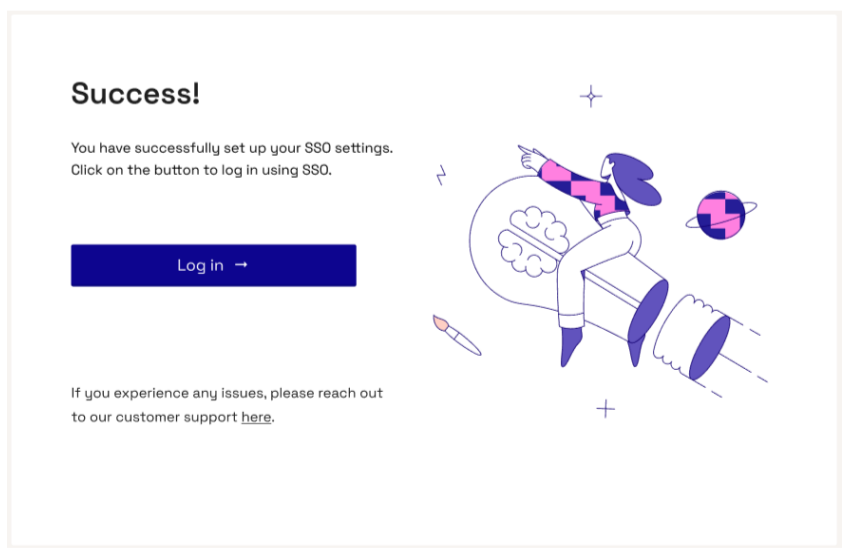
- Review your details before saving to make sure that everything is correct. If you need to edit details, select the pencil icon.

SSO Details +

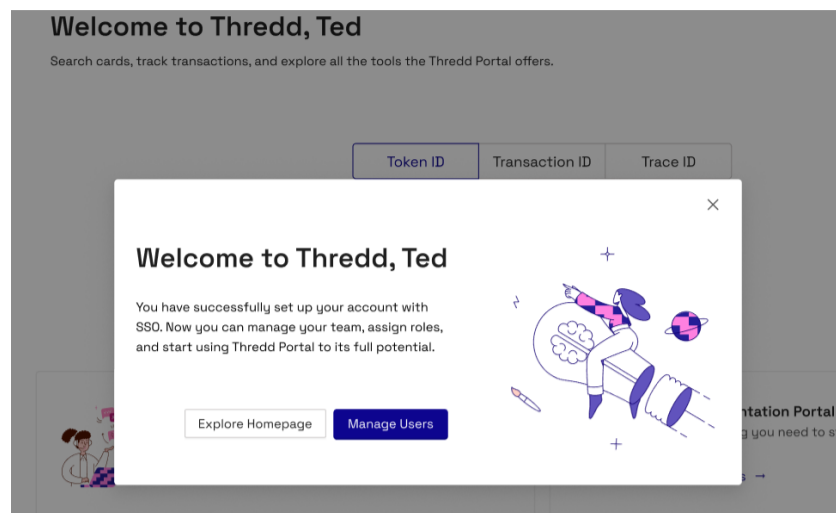
Previous step

Confirm and finish

- After you have confirmed your SSO settings, you can log in to Thredd Portal using SSO.



- When you log in to Thredd Portal, the welcome screen offers the option to manage users immediately. You can do this now (see Step 3) or return to this step later and instead set up access for your organisation to the Thredd API Hub (see [Connecting to the REST APIs](#)).



Step 3: Configure user access to the Thredd platform

To ensure that the users for your organisation can access the Thredd platform, you must invite users via your SSO provider. When a new user tries to log in to Thredd Portal, it registers them and assigns the **Read-Only** role by default.

A Super Admin can view the users that are registered for their organisation in Thredd Portal, and complete the following tasks:

- Manage user access, deactivate or reactivate a user's access to the platform.
- Assign the appropriate roles and permissions to a user.
- Edit user details, such as the email address that a user logs in with.
- Resend invitations to users.

Note: You must ensure that you configure access for any users who require it, but you can complete this task when you are ready to do so. For example, if you want to first test your organisation's connection to Thredd, such as the REST APIs, and then return to this step.

To view and manage users:

1. Navigate to **System Admin** and select **User Management**.
2. The **User Management** screen appears, and displays a directory of users by default.

Note: To add more users, you must invite them via your SSO provider.

3. The **Users** screen displays a summary of information about each user in the following columns:
 - **User ID**
 - **First Name**
 - **Last Name**
 - **Email Address**
 - **Role** – displays all roles assigned to a user, which can be more than one role.
 - **Status** – indicates if a user has logged in to Thredd Portal and is active after receiving an invitation and changing their password.
 - **Last logged in**
 - **IP Address**
 - **Actions** – the action menu for each user, which you can use to review or edit the details and role of a user.
4. To edit the details for a user, select the action menu (three dots) for an individual user and select **Edit User**.
5. The **Edit User** screen appears. Make any changes as you require, such as the role assigned to the user under **Roles**. The platform assigns all users a **Read-Only** role by default which allows view-only access to cards and transactions. You can assign a different role, and you can also assign more than one role to a user, as appropriate. Available roles include:
 - **Admin Roles:** Admin, Sensitive Information Manager
 - **Developer Roles:** Developer
 - **Cards & Transactions Roles:** Cards Operations Specialist, Card Config Manager, Card Balance Manager
 - **Read-Only Roles:** Read-Only



Thredd Portal Roles

Select Thredd roles (optional)

- Manager
- Read Only
- Card Operations Specialist
- Card Configuration Manager
- Card Balance Manager
- Chargeback Manager
- Developer
- Sensitive Information Manager

Tip: To learn more, see [Understanding Roles](#).

6. When you have made the changes, select **Next**.
7. The **Review User Details** screen appears. If the details are correct, select **Save**. To make a change, select **Back** to return to the **Edit User** screen and save the details when you are ready. A message appears to confirm that the user details have been updated.

Tip: Users who will use the cards and transaction management functionality can find information in the [Thredd Portal: Cards and Transaction Management Guide](#).

Checking and managing an existing SSO Configuration

You can review and manage the identity providers for your organisation within the administrator area of Thredd Portal. To review existing details:

1. Select **System Admin**, and then select the **SSO Configuration**. The screen displays any existing configurations in a list.
2. To review a specific configuration, select the **Actions** menu (an icon of three dots), and then select **Review Application**.

Alternatively, to check the complete details for a specific application:

1. Navigate to **System Admin > Applications**.
2. Locate the application you want to review, select the **Actions** menu (an icon of three dots) beside it, and then select **View details**. It displays the **OAuth Clients** tab by default, and the **Overview** tab.

Editing an existing SSO configuration

To edit an SSO configuration from the **OAuth Clients** tab of an Application:

1. Select the **Edit Configuration** button.
2. The **Edit Configuration** screen appears. Review the details in each section, making any changes only as necessary.
3. When you have finished reviewing or editing the details, select **OK**.

Deleting an SSO configuration

Before proceeding, make sure that you first review the details of the SSO services and have correctly identified which service to delete.

To delete an SSO configuration:

1. Navigate to **System Admin > SSO Configuration**.
2. Identify the configuration that you want to delete. Select the **Actions** menu (three dots) beside the configuration, and then select **Delete**.
3. A screen appears with a message that asks whether you are sure that you want to delete it.
 - If you do not want to proceed, select **Cancel**.
 - If you do want to proceed with deleting it, select the **Confirm and delete** button.

Warning: Deleting an SSO configuration will permanently revoke access for all users that are using it to access the Thredd platform. Therefore, make sure that any users who will continue to need access to the Thredd platform can gain access using an alternative SSO configuration. Alternatively, if you only want to make changes then you can edit the SSO configuration instead.



Setting up password login to Thredd applications

Clients who do not have their own Single Sign-On (SSO)-compliant solution (such as Entra, Google, or Okta) can manually add users and set up their password-based access to Thredd Portal.

A Super Admin can set up and manage the settings for your organisation's applications, certificates, and users through Thredd Portal. An Organisation Admin also has these permissions, except those for managing users.

This guidance assumes that you are the Super Admin and need to log in to Thredd Portal for the first time. You can manually add users and set up password-based access when Thredd sends an email to the Super Admin user notifying them that it is ready for your organisation to do this.

Note: For best practice, Thredd recommends that you first set up your organisation's access in the test (UAT) environment, so that you can verify that your organisation can use Thredd applications and API endpoints successfully. You can then set up users later.

About password (non-SSO) user access

Adding new users who will use a password to access the Thredd platform is a manual process. It requires an Admin to manually create and invite each individual user via Thredd Portal, which sends email to users inviting them to log in. Offboarding users is also a manual process, requiring Admin users to revoke access for an individual user via Thredd Portal.

Tip: As an optional alternative, Thredd recommends that you configure Single Sign-On (SSO) access to Thredd. This enables you to automate user onboarding, where your users can visit the Thredd Portal URL and automatically create their account upon their first successful login. This eliminates the need to manually create and invite users.

Prerequisites

- You must have the Super Admin role for your organisation.
- You must have received confirmation from Thredd that you can start setting up your organisation's users.

Step 1. Log in to Thredd Portal for the first time

When Thredd has completed the initial step of registering your organisation and Super Admin user, it sends two emails to the Super Admin:

- An invitation containing a link to log in to Thredd Portal and set up your access.
- A separate email that contains your temporary password that you must use to log in to Thredd Portal.

You need to refer to both emails during the following process to log in to Thredd Portal for the first time.

1. Click on the link in the email for **Log in to Thredd Portal**. This directs you to the initial access screen.
2. Enter the temporary password from the other email; this is the only password that you can use to gain access initially. Copy the temporary password and paste it into the field under **Temporary password**. Then select **Next**.

Figure: Provide temporary password screen for Thredd Portal

3. On the next screen, you must create a new password. The password must be at least eight characters in length. Enter and confirm the new password in the separate fields, and then select the **Save and Continue** button.

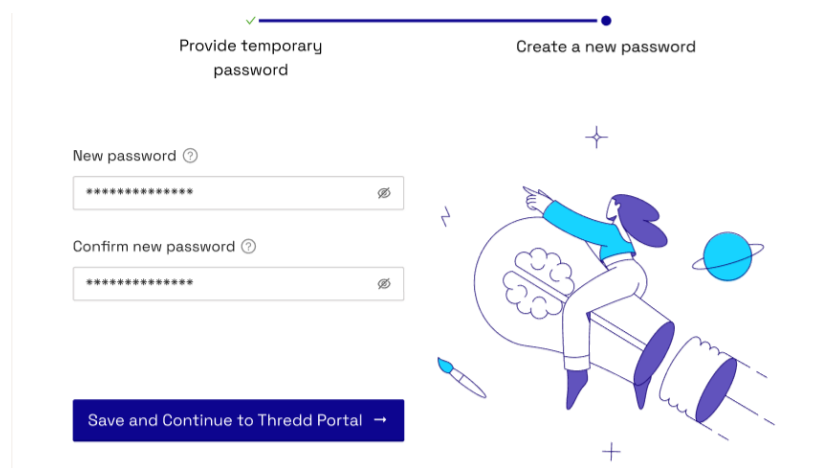


Figure: Create a new password screen for Thredd Portal

When you log in to Thredd Portal, you can set up access for your organisation's users whenever you are ready.

Note: If the link does not work or you need support, use the link at the bottom of the emails to contact Thredd. Do not reply to the email. It is system-generated and sent from a notification-only address that cannot accept incoming emails.

Step 2. Add users and configure their roles in the Thredd platform

A Super Admin user must add the details of other users at your organisation to ensure that they have access to the Thredd platform. A Super Admin can add, edit and remove users, including additional Admin users if your organisation requires this. You can either add additional users in the same flow or an Admin user can add other users later via **System Admin > User Management**.

Tip: While you must manually add and invite users who require access, you can complete this task when you are ready to do so. For example, if you want to first test your organisation's connection to the Thredd platform, such as to REST APIs, and then return to this step.

To add a new user immediately:

1. Select **Add New Users**.
2. The **Add new users** screen appears. Enter the user's first and last names, and email address.
3. Next, assign a role to the user. The platform assigns all users a **Read-Only** role by default which allows view-only access to cards and transactions. You can assign a different role to the user, and assign more than one role, as appropriate:
 - **Admin Roles:** Admin, Sensitive Information Manager
 - **Developer Roles:** Developer
 - **Cards & Transactions Roles:** Cards Operations Specialist, Card Config Manager, Card Balance Manager
 - **Read-Only Roles:** Read-Only

Tip: To learn more, see [Understanding Roles](#).

4. When you have entered the user details, you can save them or add additional users in the same flow. If you want to add additional users, select **Add Another User** and repeat the process. If you are finished adding users, select **Continue**. You can also add other users later.
5. The **Review and Finish** screen appears. If all user details are correct, select **Confirm and finish**. To make a change, select **Previous step** to return to the **Add new users** screen.
6. When you return to the **User Management** screen, you can check if the new user details have been added successfully.

Note: After an Admin user has set up access for their organisation's users, users will receive an email inviting them to log in to Thredd Portal. A user must follow the instructions in the invitation email in order to access the Thredd services appropriate for their role.

Adding a new user via User Management

Admin users can add, view, deactivate and reactivate the users for their organisation in the **User Management** area of Thredd Portal. You can also edit user details, such as the email address that a user logs in with, and assign the appropriate Thredd Portal Roles to them.

To add or view users:

1. Navigate to **System Admin** and select **User Management**.
2. The **User Management** screen appears, and displays the **User Directory** and a summary about each user. If you are adding users for the first time, you might only see the Super Admin user in the list.



- **User details** – the registered first name, last name, and email address for a user in Thredd Portal.
- **Status** – indicates whether users have logged in, changed their password and are active in Thredd Portal after receiving their invitation.
- **Organization** – the organisation that the user belongs to.
- **Actions** – the action menu for each user, which you can use to review or edit the details and role of a user.

3. To add a user, select **Invite User**.



Figure: User Directory screen in Thredd Portal

4. The **Create New User** screen appears. Enter the user's email address, and first and last names, and then select **Next**.

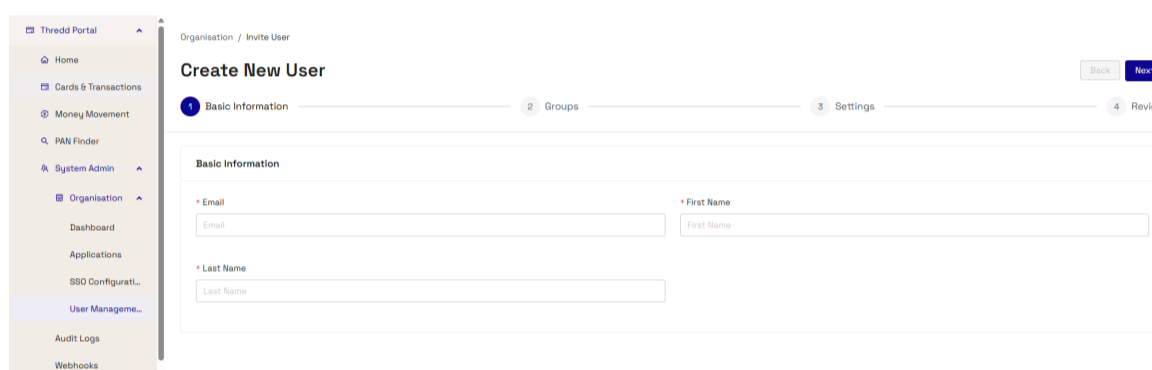


Figure: Create New User screen in Thredd Portal

5. Assign a Thredd Portal Role to a user. The platform assigns all users a **Read-Only** role by default which allows view-only access to cards and transactions. You can assign a different role to the user, and assign more than one role, as appropriate:

- **Admin Roles:** Organisation Admin, Sensitive Information Manager
- **Developer Roles:** Developer
- **Cards & Transactions Roles:** including Cards Operations Specialist, Card Configuration Manager, Card Balance Manager
- **Manager Roles:** Manager
- **Read-Only Roles:** Read Only

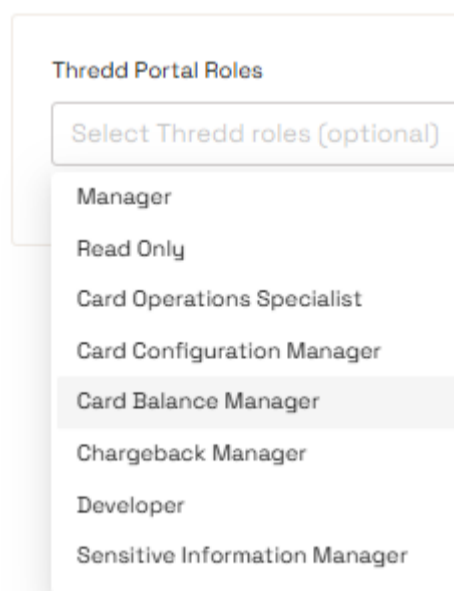


Figure: Thredd Portal Roles selection menu for a user profile

6. Continue to follow the prompts in the New User flow, and review the user details. If correct, save the new user profile.

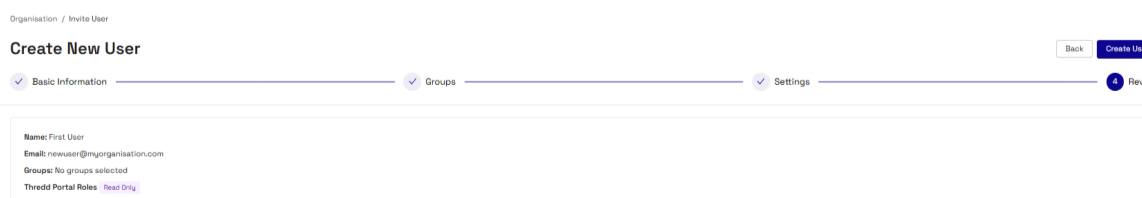




Figure: Create New User progress screen in Thredd Portal

Editing an existing user profile

A Super Admin can add, edit and remove users. To edit the details of a user:

1. Select the action menu (three dots) for an individual user.
2. Select **Edit User**.
3. The **Edit User** screen appears and displays the user's name, email address, and the role assigned to the user. You can view a list of the roles available to a user under **Roles**.

The platform assigns all users a **Read-Only** role by default which allows view-only access to cards and transactions. You can assign a different role to this if required, and you can also assign more than one role to a user.
4. To change the assigned role, select the role that is most appropriate for the user based on its description.
5. Select **Next**.
6. The **Review User Details** screen appears. If correct, select **Save**. To make a change, select **Back** to return to the **Edit User** screen.
7. When you are ready to save the details, select **Save**. A message appears to confirm that the user details have been updated successfully.

Warning: Deactivating a user will revoke their access to the Thredd platform. If you need to restore access, you can reactivate a user.



Understanding Roles

When a Super Admin user configures an individual user's profile in Thredd Portal, they can assign one or multiple roles to the user.

Roles are associated with *Users* – they provide the permissions to each individual user within your organisation based on the given *Role*. Thredd Portal Roles are preconfigured; for example, Developer, Manager, and Admin. Super Admin users can assign, view and update the Thredd Portal Roles for an individual user in Thredd Portal via **System Admin > User Management**.

Available Thredd Portal Roles

The following table describes what each Thredd Portal Role allows a user to do in Thredd Portal.

Note: Users with the Organisation Admin role in Thredd Portal have similar permissions to a Super Admin, except they cannot manage users.

Thredd Portal Role	Description
Manager	<ul style="list-style-type: none"> • Transaction Search & View • Remove auth • Card Search & View • Balance adjustment • Card Load/Unload • Change card status • PIN & CVC2 services • Edit cardholder details • Edit card configurations • Extend Thredd Expiry Date • Activate a Card • Balance Transfer
Read Only	View-only access to cards and transactions information. <ul style="list-style-type: none"> • Card Search & View • Transaction Search & View
Card Operations Specialist	Card lifecycle management and transaction visibility. <ul style="list-style-type: none"> • Card Search & View • Transaction Search & View • Change card status • Activate a Card • Extend Thredd Expiry Date • PIN & CVC2 services • Remove auth
Card Configuration Manager	Manages card configurations. <ul style="list-style-type: none"> • Card Search & View • Transaction Search & View • Edit card configurations
Card Balance Manager	Manages balance adjustments and balance transfers. <ul style="list-style-type: none"> • Card Search & View • Transaction Search & View



Thredd Portal Role	Description
	<ul style="list-style-type: none">• Balance Transfer• Balance adjustment
Developer	Configures and manages webhooks for system integrations.
Sensitive Information Manager	Restricted access to sensitive card data, including PAN Finder operations.
Organisation Admin	Full platform control, including configurations and all operational settings, except user access.

Assigning Thredd Portal Roles to a User in Thredd Portal

A Super Admin user can add a Thredd Portal Roles to a User via their user profile in Thredd Portal.

To view and edit the Roles for a User:

1. Navigate to **System Admin**, then **User Management**.
2. In the list under **User**, locate the user that you want to check and select the **Actions** menu (three dots) next to the specific user, and select **Edit User**.
3. The **Edit User** screen appears and displays the user's Basic Information. Select **Next** to view the Groups and Thredd Portal Roles.
4. Select the field under **Thredd Portal Roles** to display a dropdown menu of the different roles that are available for the user. Assign the appropriate portal roles that will provide access to the specific features and card operations that a user requires. You can also select a group if a group is available. Select **Next**.
5. Review the user's details and select **Update User** to save the settings. If you need to make any changes, select **Back** and repeat the previous steps before updating the user details.

Tip: If the user is logged in to Thredd Portal at the time that the Admin makes any changes, the user might need to log out and then log in again to ensure that the changes take effect.



Creating an Application and OAuth Client

Before you can access Thredd's REST API, you must create an application and an OAuth Client in Thredd Portal to register it and receive a Client ID. You can then request client credentials for your application, and configure it and a REST interface (such as Postman) to provide credentials when making API calls. Think of the Client ID as the passport for your OAuth Client, and therefore your application. Combined with the credentials you need for your chosen Client Authentication method, it allows Thredd to verify your identity and grant access.

Summary of the steps

The steps for creating a new application and OAuth Client are as follows:

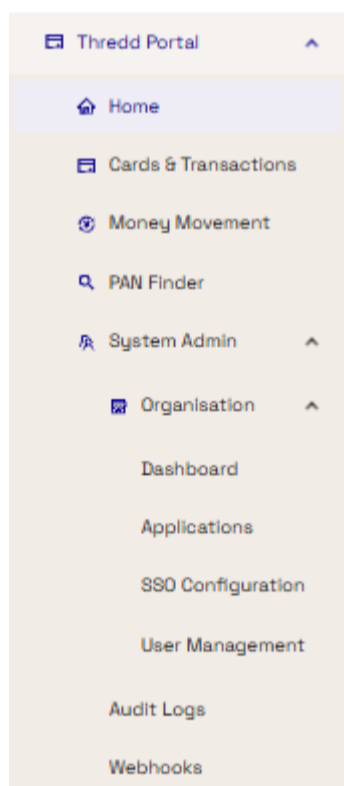
1. Log in to Thredd Portal.
2. Create a new client application, adding a name and description.
3. Create a new OAuth Client for the application.
4. Ask Thredd to add a PMID to your OAuth Client.

Prerequisites

1. You must have access to Thredd Portal.
2. You must be an Admin for your organisation.
3. You must have installed OpenSSL on the machine (the Server) that will make requests to Thredd platform (the Client).

Step 1: Log in to Thredd Portal

Log in to Thredd Portal and select **System Admin**. In **System Admin** menu, select **Organisation**, then **Applications**.



Step 2: Create a new client application

The **Applications** screen lists any applications that are registered in Thredd Portal.

Note: If an existing application is already present (and you do not need to create a new one), then check whether an OAuth Client already exists for it. Select the **Actions** menu and select **Review Application**. Check the **OAuth Clients** screen; if an OAuth Client is not present, see step 3.

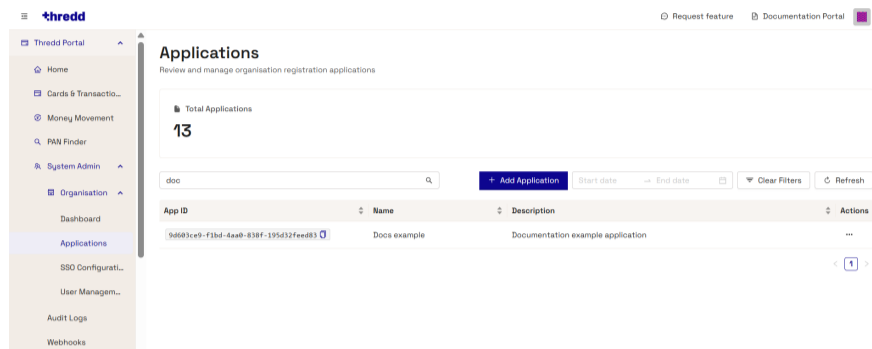
To create a new application:

1. Select the **Add Application** button.
2. The **Create New Application** screen appears. Enter the basic information, such as a name and description, for the application.



Tip: Thredd recommends that you use an application name that you can easily identify for any application management activities. For example, to distinguish it from other applications when you need to obtain credentials, such as a Client ID, access token or certificate.

3. Select **Next** and review the new application summary; select **Back** to make changes or select **Create Application** to save it.
4. Your new application now appears in the **Applications** screen. Select the menu (three dots) under **Actions** and select **Review Application**. The application summary opens on the **OAuth Clients** tab by default. See *Step 3. Create a new OAuth Client*.



Step 3: Create a new OAuth Client

The application details has two tabs; **OAuth Clients** and **Certificates**. The steps to create an OAuth Client are as follows:

1. In the **OAuth Clients** tab, select **Create Client**.
2. The **Create OAuth Client** screen appears, starting with the **Basic information** stage. Enter the basic information, such as the **Client Name** and **Description**, and choose the **Application Type**. Select **Next**. The Application Type choices are:
 - **Native App (Mobile/Desktop):** iOS, Android, or desktop applications that cannot securely store credentials
 - **Single Page Applications (SPA):** React, Vue, Angular web applications
 - **Web Application:** Traditional server-side web applications
 - **Machine to Machine:** APIs, microservices, backend services

The screenshot shows the 'Create OAuth Client' form in the 'Basic Information' stage. It includes a progress bar with four steps: 1. Basic Information, 2. Authentication, 3. Token Configuration, and 4. Review & Submit. The form has two main input fields: 'Client Name' (with placeholder 'My Application') and 'Description' (with placeholder 'A mobile application for accessing user data...'). Below these is the 'Application Type' section with four radio button options: 'Native App (Mobile/Desktop)', 'Single Page Application (SPA)', 'Web Application', and 'Machine to Machine'.

3. The **Authentication** stage summarises the recommended **Security Configuration** based on the Application Type that you selected. To continue, select **Next**. To change the information, select **Back**. The security configurations for each Application Type are as follows:
 - **Native App (Mobile/Desktop):** Grant Types are authorization_code, refresh_token, and PKCE is Required (Recommended). Note: A Native App is a mobile or desktop application that cannot securely store credentials.
 - **Single Page Applications (SPA):** Grant Types are authorization_code, refresh_token, and PKCE is Required (Recommended).
 - **Web Application:** Grant Types are authorization_code, refresh_token, and PKCE is Optional.
 - **Machine to Machine:** Grant Types are client_credentials, and PKCE is Optional.
4. The **Token Configuration** stage allows you to configure how long tokens remain valid. Shorter lifetimes are more secure but may require more frequent re-authentication. Once you have reviewed the default settings or made changes, select **Next**.
5. The **Review** stage allows you to review your configuration, based on the information that you entered in Basic Information, Authentication, and Token Configuration. You can view the list of available **Scopes** for your application in the **Authentication** section. To proceed, select **Submit**; to make changes select **Back** and then submit the details to finish creating the OAuth Client.

You have now registered a new OAuth Client, with a unique Client ID for your application. You can view it in the **OAuth Clients** screen of your application's details in Thredd Portal.



Organisation / Applications

Create OAuth Client

Back Submit

Basic Information Authentication Token Configuration Review & Submit

Please review the settings below before creating your OAuth client.

Basic Information

Name: Example client Type: M2M
Description: Documentation example

Authentication

Scopes: [apata.read](#) [apata.write](#) [cta.read](#) [cts.write](#) [3ds.read](#) [bulkcard.read](#) [bulkcard.write](#) [cards.read](#) [cards.write](#) [cvr.read](#) [cvr.write](#) [pin.read](#) [pin.write](#) [limits.read](#) [limits.write](#) [digitalwallets](#) [digitalchannel](#) [eds.read](#) [eds.write](#) [issuer.read](#) [issuer.write](#) [scamdetect](#) [cards.sensitive](#) [cards.encrypted](#) [es](#)

Grant Types: [client_credentials](#)

Token Configuration

Access Token: 1 hour(s)	Refresh Token: 30 day(s)	ID Token: 1 hour(s)
----------------------------	-----------------------------	------------------------

Step 4: Ask Thredd to add a PMID to your OAuth Client

Now that you have created the application and OAuth Client, you need to associate a PMID with your OAuth Client. This PMID is passed in your OAuth token.

To do this, raise an IAM support ticket with Thredd for the IAM team and request that Thredd adds the specific PMID for accessing the API Hub to your OAuth Client.

Make sure that your request clearly specifies the exact PMID needed to avoid delays.

Note: Only Thredd's internal Identity and Access Management team users can assign or modify PMIDs.

Next steps

You must make sure that you request and provide the necessary credentials from Thredd to access Thredd's REST APIs.

These credentials depend on the Client Authentication method that your Super Admin selected when setting up your organisation in Thredd Portal, as either:

- Client Secret
- Private Key JWT

If you are using an mTLS connection, your application must also present a Transport Certificate from Thredd. Thredd adopts a self-service approach, which allows an Admin user to independently request and manage certificates via Thredd Portal.

See [Connecting to the REST APIs](#).



Understanding Scopes

Thredd assigns products (and scopes) to your organisation during the implementation stage, according to the Thredd services that you will use. *Applications* are associated with *Scopes* – these relate to the products and services that your organisation will use in its agreement with Thredd. When you register Application for your organisation in Thredd Portal, it is assigned the appropriate scopes. Super Admin and Organisation Admin users can view the available Scopes for their Application in Thredd Portal via **System Admin > Applications**.

Available Scopes

The following is an example of some of the scopes that are available from Thredd. These relate to the API endpoints that your organisation has access to. Thredd assigns scopes to your organisation according to the Thredd services that your organisation will use.

Scope Name	Description
cards.read	Enables you to use GET endpoints to return information on cards.
cards.write	Enables you to use the POST, UPDATE and PATCH endpoints to create and update card information.
pin.read	Enables you to use the Retrieve PIN endpoint.
pin.write	Enables you to use the Set PIN and Unblock PIN endpoint.
cvv.read	Enables you to use the Retrieve CVV endpoint.
cvv.write	Enables you to use the Unblock CVV endpoint.
bulkcard.read	Enables you to use the Get Bulk Card Progress endpoint.
bulkcard.write	Enables you to use the Create Bulk Card endpoint.
cards.encrypted	Enables you to use the Get Encrypted Data endpoint.
cards.sensitive	Enables you to use the Get Full PAN endpoint.
3ds.read	Enables you to use the Get 3DS Configuration endpoint.
ads.read	Enables you to use the GET endpoints for Visa Alias Directory.
ads.write	Enables you to use the POST, DELETE and PUT endpoints for Visa Alias Directory.



Section 3: REST Applications

You should read this section to understand the set up steps for REST applications.

Topics covered in this section:

- [Connecting to the REST APIs](#)
- [Requesting a Signing Certificate to access REST APIs using Private Key JWT client authentication](#)
- [Requesting certificates for applications using mTLS](#)
- [Using the Postman Collection to call REST APIs over TLS](#)
- [Using the Postman Collection to call REST APIs over mTLS](#)
- [Updating Postman](#)



Connecting to the REST APIs

There are two main areas in which you will need to set up your connection to access Thredd REST APIs:

1. Thredd Portal: Log in to Thredd Portal and create a client application, and obtain the client credentials for your application.
2. A REST tool: Configure your REST interface, such as Postman, to interact with Thredd's identity and access management platform, so that you can access the API endpoints.

For best practice, Thredd recommends that a Super Admin first sets up your organisation's access in the UAT (User Acceptance Test) environment. This enables you to test and verify that your organisation can use the API endpoints successfully. Once an Admin has verified that your testing in the UAT environment is satisfactory, you can prepare for migrating to a Production environment.

Note: For your Production environment, you will need to request new credentials and ensure that the environment-specific configuration is correct. Make sure that old credentials are not cached, such as a certificate or token, and that you send updated credentials to Thredd.

Overview of set up steps

You need to complete the following steps to set up your connection to Thredd for the first time.

1. Create a client application and OAuth Client in Thredd Portal and get a Client ID. If you have already done this, see step 2.
2. Get credentials for your client application via Thredd Portal.
3. Configure your REST tool to interact with Thredd to make calls to Thredd's API that include an access token. You must obtain an access token using your client credentials. You can also download Thredd's Postman Collection and configure your client credentials as variables in Postman to test making API calls in a UAT environment.
4. Get an access token from Thredd's OAuth token endpoint to include in the Authorization header of your API request
5. Make a call to Thredd's REST API to test your setup.

Step 1: Create a Client Application in Thredd Portal and get a Client ID

In order to call Thredd's REST APIs, you need to create an application and OAuth Client in Thredd Portal. This registers your client application with the Thredd's identity and access management platform. Make a note of the Client ID of your application.

See [Creating an Application and OAuth Client](#). If you have already completed this step, see step 2.

Step 2: Get credentials for your Application via Thredd Portal

To access any Thredd API endpoint you must exchange your client credentials for a short-lived Bearer token (OAuth token).

The credentials that you require to obtain an access (OAuth) token for Thredd's REST APIs depend on the Client Authentication method that you will use; *Client Secret* or *Private Key JWT*.

Once you have obtained your credentials, you must request an OAuth token and include it in the header of your requests to the REST API endpoints. To get an OAuth token, you can either use the endpoint within the Postman Collection (when testing in a UAT environment) or call the OAuth Token endpoint directly when using a Production environment (see step 3).

Complete the following steps to obtain the information you need via Thredd Portal, based on your Client Authentication method.

Client Secret authentication

You need:

- Client ID of your application
- Access token from Thredd's OAuth Token endpoint
- If you use mTLS, you also require a Transport Certificate from Thredd

Get the Client ID:

1. In Thredd Portal, navigate to **System Admin**, and select **Applications**. Select the **Actions** menu (icon of three dots) and select **Review Application**. A screen appears with tabs for the **OAuth Clients** and **Certificates** for your Application.
2. On the **OAuth Clients** tab, locate the OAuth Client that you will use to connect to the Thredd REST APIs. Select the **Actions** menu (icon of



three dots) and select **View Details**. The **Overview** screen for the OAuth Client appears.

3. On the **Overview** tab, locate the **Client ID** under Client Configuration. You can copy the Client ID value from here when you need to.

Private Key JWT authentication

You need:

- Client ID of your application
- Signing Certificate
- Access to your JWKS endpoint
- JSON Web Token, which you generate using your Signing Certificate and fetch from your JWKS endpoint
- Access token from Thredd's OAuth Token endpoint
- If you use mTLS, you also require a Transport Certificate from Thredd

Get the Client ID:

1. In Thredd Portal, navigate to **System Admin**, and select **Applications**. Select the **Actions** menu (icon of three dots) and select **Review Application**. A screen appears with tabs for the **OAuth Clients** and **Certificates** for your Application.
2. On the **OAuth Clients** tab, locate the OAuth Client that you will use to connect to the Thredd REST APIs. Select the **Actions** menu (icon of three dots) and select **View Details**. The **Overview** screen for the OAuth Client appears.
3. On the **Overview** tab, locate the **Client ID** under Client Configuration. You can copy the Client ID value from here when you need to.

Get a Signing Certificate:

An Admin user can obtain a Signing Certificate from Thredd Portal to create and register a private key. You must generate a Client Assertion, a JSON Web Token (JWT) that is signed by your private key, and configure your client for Private Key JWT authentication. See [Requesting a Signing Certificate for an application](#).

Get the JSON Web Key Set via your JWKS endpoint:

To ensure secure communication with Thredd, its Identity and Access Management (IAM) platform uses JSON Web Key Sets (JWKS). This industry-standard format allows your systems to automatically verify the authenticity of digital signatures and encrypted data.

Every organisation registered with Thredd's IAM platform is assigned a unique, public JWKS endpoint. The JWKS is a set of keys that includes your public key that you created when you requested your Signing Certificate in Thredd Portal.

Use the following URL to get your organisation's registered keys via your JWKS endpoint. The `<organisation_id>` parameter is your unique Thredd organisation identifier.

```
http://jwks.threddid.com/<organisation_id>/jwks.json
```

Get a Transport Certificate – only mTLS connections:

If you are using an mTLS connection, your application requires a Transport Certificate for establishing mutual connections between your Client and Thredd. An Admin user can obtain a Transport Certificate from Thredd Portal. See [Requesting a Transport Certificate for applications using mTLS](#).

Step 3: Configure your REST tool to make calls to Thredd API using your client credentials

You can access Thredd's API Hub and REST APIs over TLS or mTLS. You must make sure that you configure your REST tool with the appropriate settings, for example:

- Base URL – this depends on whether you are using the UAT or Production environment, and TLS or mTLS connectivity.
- Headers – you must set the correct Authorization, Content-Type, and X-Region headers.

Base URL

- TLS connectivity for the UAT environment: uat-api.thredd.com
- TLS connectivity for the Production environment: api.thredd.com
- mTLS for the UAT environment: uat-mtls.thredd.com
- mTLS connectivity for the Production environment: mtls.thredd.com



Configuring headers

Whether you are using your own tool or the Postman Collection, you must make sure that you configure the following headers with the correct information in each API request. These headers are mandatory for all requests that you make to the API Hub.

- Authorization header – include your OAuth token in the [authorization](#) header as appropriate for your application's authentication method.
- X-Region header – this is mandatory and determines the region/environment you are trying to connect to. Select one of the following values to include in the [X-Region](#) header:
 - Use [0](#) for the Default environment
 - Use [1](#) for the EMEA environment
 - Use [2](#) for the APAC environment

Note: If you are not sure which environment to use, contact your Thredd Account Manager or Implementation team. By default, the Postman Collection is set to use the Default environment (0), but you can change this in the variables section.

- Content-Type header – specify the media type of the requested resource in the [content-type](#) header. Example value: [application/x-www-form-urlencoded](#)

For some endpoints, you can pass pagination values. Where this is relevant, this is stated in the documentation for a given endpoint in API Reference on the API Hub.

Thredd provides separate Postman Collections for TLS and mTLS, which enable you to use Postman to test using the Thredd REST APIs in a UAT environment. To use the collection, you need to configure Postman with the appropriate environment variables and credentials.

Postman Collection

Download the Postman Collection from the API Hub and follow the guide that matches your configuration:

- For TLS connections: [Using the Postman Collection to call REST APIs over TLS](#)
- For mTLS connections: [Using the Postman Collection to call REST APIs over mTLS](#)

To download the Postman Collection, see [Accessing the API Hub](#).

Step 4: Get an access token using your client credentials

You must request an access token from Thredd and include it in the Authorization header of your requests to the REST API endpoints. You can get an access token using the following methods:

- Use the endpoint within the Postman Collection (when testing in a UAT environment).
- Call the OAuth Token endpoint directly when using a Production environment.

Access tokens are valid for one hour and include the scopes that are associated with your application.

Follow the advice that match your Client Authentication method.

Client Secret authentication

You must provide your:

- Client ID
- Client secret

Endpoint:

POST <https://threddid.com/oauth2/token>

Example request body (URL-encoded):

```
curl --location 'https://threddid.com/oauth2/token' \  
      --header 'Content-Type: application/x-www-form-urlencoded' \  
      --data-urlencode 'grant_type=client_credentials' \  
      --data-urlencode 'client_id=YOUR_CLIENT_ID' \  
      --data-urlencode 'client_secret=YOUR_CLIENT_SECRET'
```

If you use an mTLS connection, you must also include your Transport Certificate in your request to the OAuth token endpoint. The Authorisation Server will reject all requests without this.

Private Key JWT authentication

You must provide your:



- Client ID
- Signed JSON Web Token

Endpoint:

```
POST https://threddid.com/oauth2/token
```

Example request body (URL-encoded):

```
curl --location 'https://threddid.com/oauth2/token' \
  --header 'Content-Type: application/x-www-form-urlencoded' \
  --data-urlencode 'grant_type=client_credentials' \
  --data-urlencode 'client_assertion_type=urn:ietf:params:oauth:client-assertion-type:jwt-bearer' \
  --data-urlencode 'client_assertion=YOUR_SIGNED_JWT'
```

If you use an mTLS connection, you must also include your Transport Certificate in your request to the OAuth token endpoint. Thredd's Authorisation Server will reject all requests without this.

Example OAuth token response

```
{
  "access_token": "eyJhbGciOi...",
  "token_type": "Bearer",
  "expires_in": 3600
}
```

Access tokens are valid for one hour and include the scopes that are associated with your application.

You must configure your application to refresh the token automatically when it expires.

Step 5: Make a call to Thredd's REST APIs

Once you have the [access_token](#), you must include it in the Authorization header of every request to Thredd's REST APIs.

The next step is to test making an API call. Refer to the documentation for the base URLs and API Reference for endpoints at the API Hub: <https://cardsapidocs.thredd.com/v2.0/docs>

Remember to include the following in your requests:

- Base URL – this depends on whether you are using the UAT or Production environment, and TLS or mTLS connectivity.
- Mandatory information in your request headers.
- Correct HTTP method for the endpoint.

If you change the environment or region that you want to make API calls to, you must update the Base URL and headers that you include in your requests.

Example request header (TLS)

```
GET /core/api/v1/products HTTP/1.1
Host: api.thredd.com
Authorization: Bearer eyJhbGciOi...
Content-Type: application/json
```

Example request header (mTLS)

```
GET /core/api/v1/products HTTP/1.1
Host: mtls.thredd.com
Authorization: Bearer eyJhbGciOi...
Content-Type: application/json
```

Note: For mTLS, when making your request to uat-mtls.thredd.com (UAT environment) or mtls.thredd.com (production environment) you must also include your Transport Certificate in the request. Thredd's APIs will reject all requests that do not include this.



Requesting a Signing Certificate to access REST APIs using Private Key JWT

Client applications that use the Private Key JWT Client Authentication method require a Signing Certificate. Creating a Signing Certificate is an essential step for enabling access for OAuth 2.0 client applications that connect to Thredd's REST APIs.

This method requires you to generate a Client Assertion, which is a JSON Web Token (JWT) that is signed by your private key. When you request a Signing Certificate, you register a public key with Thredd and can generate a private key for signing assertions to send to the Authorisation Server.

Whether you want to generate a new certificate, or renew or revoke a certificate, you can manage certificates through Thredd Portal.

Note: If you are using Client Secret authentication, you do not need a Signing Certificate. However, if you will connect to Thredd APIs using an mTLS connection, you need a Transport Certificate regardless of the client authentication method you use. For mTLS connections, see [Requesting a Transport Certificate for applications using mTLS](#).

Purpose of Signing Certificates

A *Signing Certificate* (or Signing Key) is required to create signed messages, for authentication of clients, and non-repudiation and authentication of notifications. To authenticate with Thredd using the Private Key JWT method, you must generate a Client Assertion in the form of a JSON Web Token (JWT) and ensure that your client presents it when requesting an access token from Thredd. You must then include the access token in any calls that you make to the Thredd REST APIs.

When you request a Signing Certificate from Thredd, it registers a public key on Thredd's identity and access management platform, and generates a private key that you store on the Client (your application).

At the start of a TLS session, the Client (your application) presents the signing key during a TLS handshake to prove its identity to the Server (Thredd). This allows Thredd to verify its identity, requiring the Client to sign a message with its private key, which remains securely stored on the Client. If the signature matches, the Server trusts the Client and generates a temporary secure token for access to the APIs. This ensures message integrity, because only the holder of the private key can sign data.

Note: You must use certificates from Thredd to connect to Thredd APIs. Using self-signed or third-party certificates may result in Thredd refusing connections.

Certificate Lifecycle Management

Regularly rotating (refreshing) your certificate is essential to maintain secure communication between your applications and the Thredd platform. Thredd sends an email to you when a certificate is approaching its expiry date, notifying you 60 days in advance. If you want to renew a certificate, you should request a new Signing Certificate and rotate it before the date that it expires. This ensures that your application can continue to access Thredd applications.

You can also revoke a certificate at any time by logging in to Thredd Portal and choosing the option to revoke it in the **Certificates** tab of your application. If you submit a request to revoke a Signing Certificate, Thredd fulfils it instantly.

Summary of the steps

The steps for obtaining a certificate are as follows:

1. Log in to Thredd Portal and select or create the application that requires a certificate.
2. Create a private key and Certificate Signing Request for a new Signing Certificate.
3. Upload the Certificate Signing Request for your application.
4. Download the certificate for your application.
5. Store the private key on your client and ensure it can use the Signing Certificate.

Prerequisites

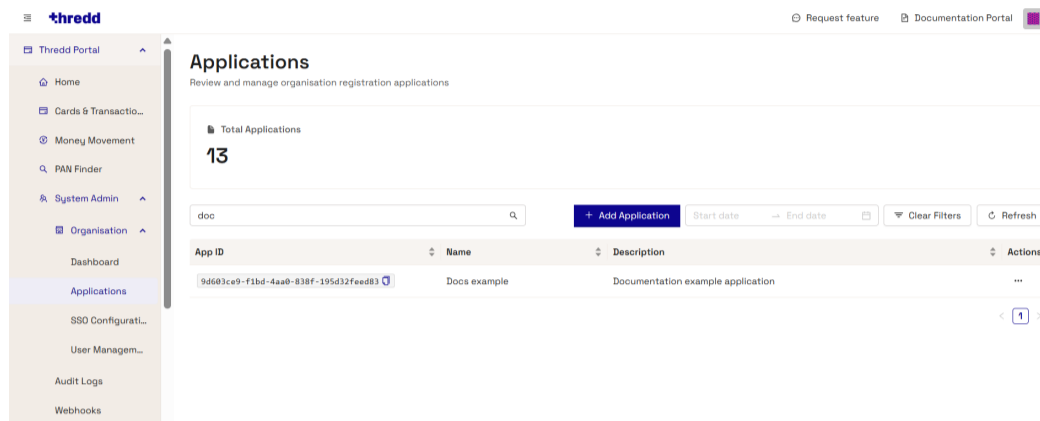
1. You must have access to Thredd Portal.
2. You must have an Admin role in Thredd Portal.



3. You must have installed OpenSSL on the machine (the Server) that will make requests to Thredd platform (the Client).
4. Your organisation has an application in Thredd Portal. If you need to create an application, see [Creating an Application and OAuth Client](#).

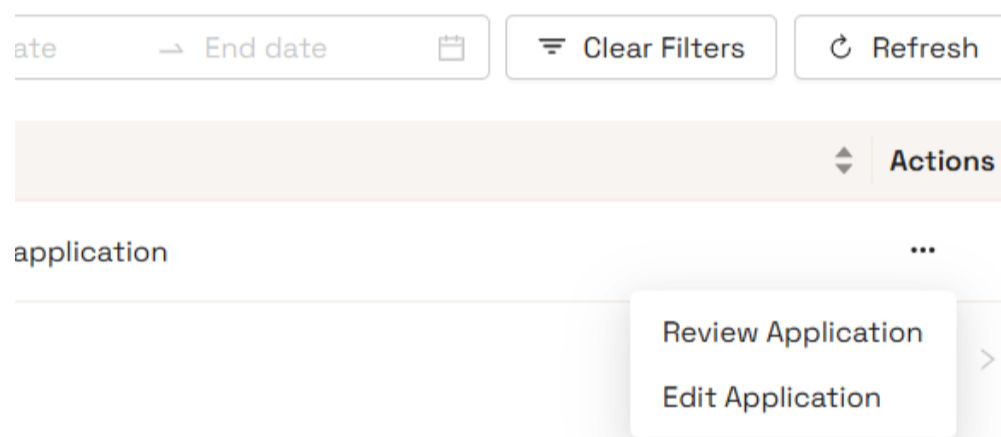
Step 1: Log in to Thredd Portal and select or create the application that requires a certificate

Log in to Thredd Portal and select **System Admin**. From the **System Admin** menu, select **Applications**. You will have access to at least one organisation for your parent company. The **Applications** screen lists any applications that are registered in Thredd Portal.



If an existing application is already present and you want to create a signing certificate for it, do the following:

1. Select the **Actions** menu (three dots) and select **Review Application**.



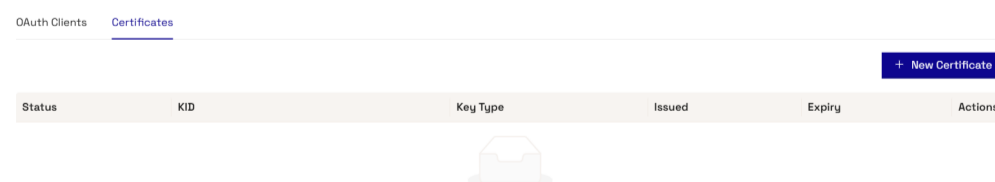
2. The application summary appears and displays the **OAuth Clients** screen. Select the **Certificates** tab. See step 2 to create a certificate.

Tip: If you need to create a new application, you can do so by selecting the **Add Application** button. Follow the on-screen prompts to create the application, and then create an OAuth Client for it. For a full guide, including an explanation of the different Application Types and their associated Security Configurations, see [Creating an Application and OAuth Client](#).

Step 2: Create a private key and Certificate Signing Request for a new Signing Certificate

Make sure you are in the application summary for your application, and do the following:

1. Select the **Certificates** tab.
2. Select the **New Certificate** button. The **New Certificate** screen appears.



3. Select **Signing** from the **Certificate Type** dropdown menu, and then select **Next**.



New Certificate [Close]

Progress: Certificate Type (Active), Generate CSR, Upload CSR/PEM

* Certificate Type

Signing (Selected)

Transport

Encryption

Next

4. Copy the OpenSSL command from the clipboard and run the command on the machine that you have installed OpenSSL on. This command creates a private key and a Certificate Signing Request (CSR). You must store the private key securely.

Example:

```
openssl req -new -newkey rsa:2048 -nodes -out MyUniqueFileName-rtssigning.csr -keyout MyUniqueFileName-rtssigning.key -subj "/C=MyCountry/O=MyOrganization/OU=MyOrganizationalUnit/CN=MyUniqueFileName" -sha256
```

Note: This example contains placeholder values, but the command that you copy contains your unique file name for .csr (-out) and .key (-keyout), and organisation information in the certificate subject (-subj).

New Certificate [Close]

Progress: Certificate Type, Generate CSR (Active), Upload CSR/PEM

Generate CSR

```
openssl req -new -newkey rsa:2048 -nodes -out MyUniqueFileName-rtssigning.csr -keyout MyUniqueFileName-rtssigning.key -subj "/C=MyCountry/O=MyOrganization/OU=MyOrganizationalUnit/CN=MyUniqueFileName" -sha256
```

Previous Next

5. Once you have successfully created the private key (.key) and CSR (.csr) files, select **Next**.

Step 3: Upload the Certificate Signing Request for your application

1. Select the **Upload CSR** button and select your Certificate Signing Request file. This uploads your Certificate Signing request onto the Thredd platform, so that it can use the file to request the Signing Certificate from the Thredd Certificate Authority (CA).



2. Select **Save**.

If this is successful, Thredd's system adds the newly-issued Signing Certificate to your application and displays the Key ID (KID) in your certificates list. You can verify this in the **App Certificates** screen of the **Certificates** tab for your application.

Step 4: Download the certificate for your application

Next, download the certificate from the **App Certificates** screen.

Select the **Actions** menu (three dots) next to the SIGNING certificate, and select **Download Certificate**.

Issued	Expiry	Actions
26 Feb 2026 15:34:45	26 Feb 2027 16:34:45	...
26 Feb 2026 15:34:19	26 Feb 2027 16:34:18	...

Download Certificate

Revoke Certificate

Step 5: Generate a Client Assertion using your registered private key

Your Signing Key is now registered with Thredd's identity and access management system.

- The *public key* is available on your organisation's corresponding JSON Web Key Set (JWKS) endpoint.
- You must store the *private key* securely on your Client and use it to authenticate when calling Thredd's OAuth token endpoint.

To authenticate with Thredd, you must generate a Client Assertion in the form of a JSON Web Token (JWT) and ensure that your client presents it in all calls that it makes to the Thredd REST APIs. This token is signed with the private key that you registered when you created the Signing Key when you requested the Signing Certificate. You can use the following example of the JWT structure to use.

Required JWT structure

- iss (Issuer): Your unique Client ID for your client application.
- sub (Subject): Your unique Client ID (same as the issuer).



- aud (Audience): The URL of the Thredd Authorisation Server token endpoint at www.threddid.com/oauth2/token
- jti (JWT ID): A unique identifier for the token to prevent replay attacks.
- exp (Expiration): A timestamp indicating when the token expires (typically set to 5 minutes from generation).

Signing the token

You must ensure that JWTs are signed using your *private key*. To understand how to do this for your setup, refer to your Client library documentation.

Tip: Most programming languages have well-supported libraries that handle the main aspects of creating and signing a JWT.

Next steps

Once you have configured your Client to use the Signing Certificate/Key to create a JWT, you can begin setting up your connection to the API Hub using the latest Postman Collection.

- For TLS connections, see [Using the Postman Collection to call REST APIs over TLS](#).
- For mTLS connections, you must also request a Transport Certificate and use the mTLS Postman Collection. See [Requesting a Transport Certificate for applications using mTLS](#) and [Using the Postman Collection to call REST APIs over mTLS](#).



Requesting certificates for applications using mTLS

Client applications require a Transport Certificate in order for you to access Thredd's REST APIs over an mTLS connection.

Additionally, if you are using the Private Key JWT Client Authentication method, then you also require a *Signing Certificate* to create and register a private key. You must generate a Client Assertion, a JSON Web Token (JWT) that is signed by your private key. If you are using Client Secret authentication, you do not need a Signing Certificate.

Whether you want to generate a new certificate, or renew or revoke a certificate, you can manage certificates through Thredd Portal.

Note: You must use certificates from Thredd to connect to Thredd APIs. Using self-signed or third-party certificates may result in Thredd refusing connections.

Purpose of Transport and Signing Certificates

A *Transport Certificate* is mandatory for mTLS and is used solely for establishing the secure mutual connection between the Server and the Client. Without a valid, signed Transport Certificate, your application is unable to connect to Thredd's secure API endpoints.

A *Signing Certificate* (or Signing Key) is required if you are using the Private Key JWT Client Authentication method. To authenticate with Thredd using the Private Key JWT method, you must generate a Client Assertion in the form of a JSON Web Token (JWT) and ensure that your client presents it when requesting an access token from Thredd. You must then include the access token in any calls that you make to the Thredd REST APIs. To learn more, see [Requesting a Signing Certificate](#).

Certificate Lifecycle Management

Regularly rotating (refreshing) your certificates is essential to maintain secure communication between your applications and the Thredd platform. New certificates are valid for 350 days; Thredd sends an email to you when a certificate is approaching its expiry date, notifying you 60 days in advance. If you want to renew a certificate, you should request a new Transport or Signing Certificate and rotate it before the date that it expires. This ensures that your application can continue to access Thredd applications.

You can also revoke a certificate at any time by logging in to Thredd Portal and choosing the option to revoke it in the **Certificates** tab of your Application. Fulfilling a revocation request can take up to one hour for Transport Certificates and is instant for Signing Certificates.

Summary of the steps

The steps for obtaining a certificate are as follows:

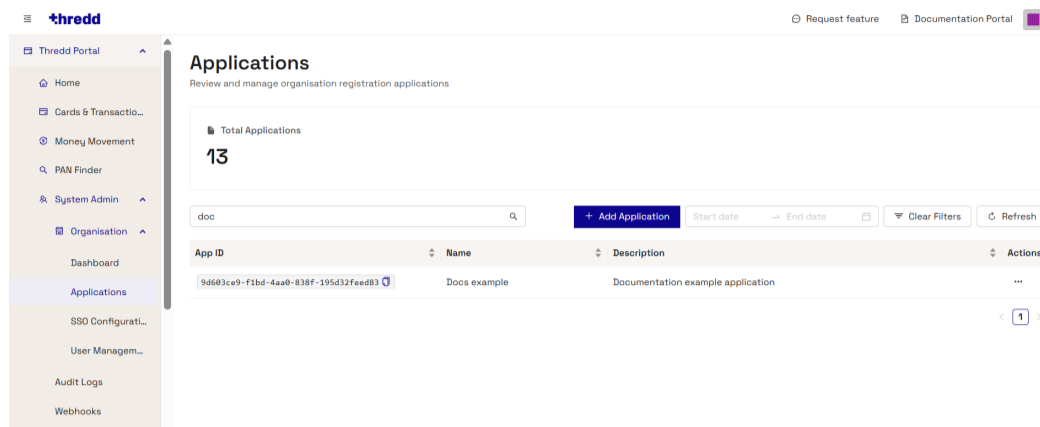
1. Log in to Thredd Portal and select or create the application that requires a certificate.
2. Create a private key and Certificate Signing Request for a new Transport Certificate.
3. Upload the Certificate Signing Request for your application.
4. Download the certificate for your application.
5. Store the private key on your client and ensure it can use the Transport Certificate.
6. Configure your Client to use the Transport Certificate in your requests.
7. Only for applications using Private Key JWT Client Authentication: Create a private key and Certificate Signing Request for a new Signing Certificate. You need to download the Signing Certificate, store the private key, and configure your Client to use JWT authentication.

Prerequisites

1. You must have access to Thredd Portal.
2. You must have an Admin role in Thredd Portal.
3. You must have installed OpenSSL on the machine (the Server) that will make requests to Thredd platform (the Client).

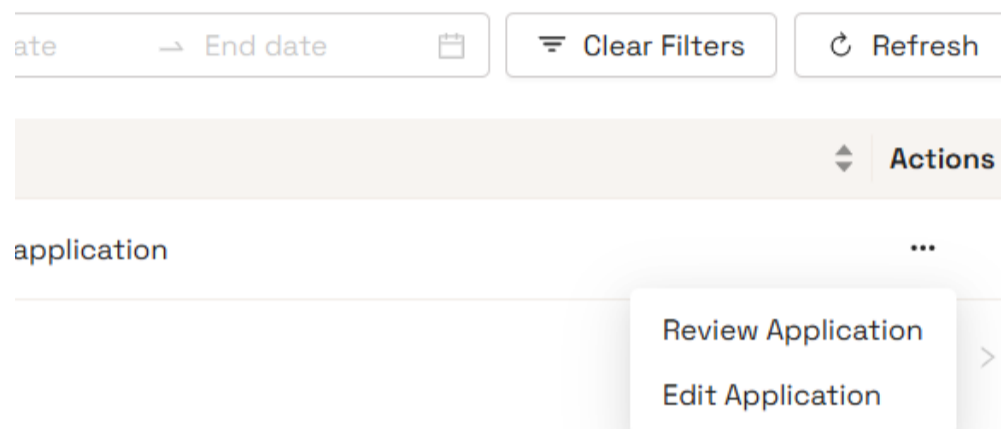
Step 1: Log in to Thredd Portal and select or create the application that requires a certificate

Log in to Thredd Portal and select **System Admin**. From the **System Admin** menu, select **Applications**. You will have access to at least one organisation for your parent company. The **Applications** screen lists any applications that are registered in Thredd Portal.



If an existing application is already present and you want to create a signing certificate for it, do the following:

1. Select the **Actions** menu (three dots) and select **Review Application**.



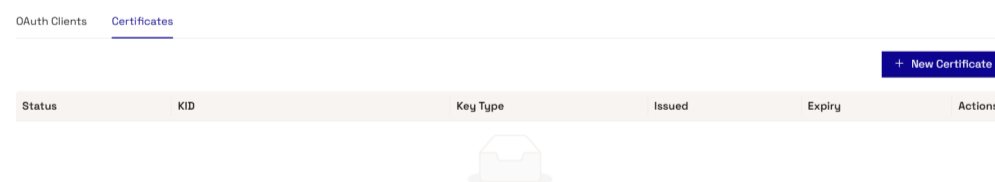
2. The application summary screen appears and displays the **OAuth Clients** screen. Select the **Certificates** tab. See step 2 to create a certificate.

Tip: If you need to create a new application, you can do so by selecting the **Add Application** button. Follow the on-screen prompts to create the Application, and then create an OAuth Client for it. For a full guide, including an explanation of the different Application Types and their associated Security Configurations, see [Creating an Application and OAuth Client](#).

Step 2: Review the application details and create a new Transport Certificate

The **Applications** screen lists an entry for your application once it is registered in Thredd Portal. To request a Transport Certificate:

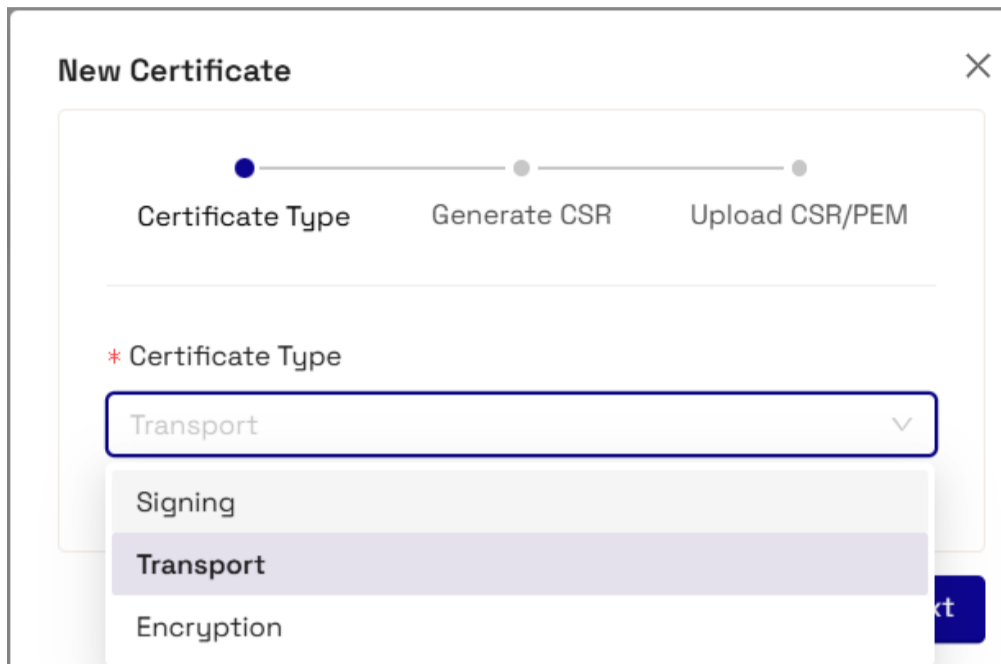
1. Select the Action menu (three dots) under **Actions** and select **Review Application**.
2. The **OAuth Clients** screen appears; it displays a summary of your application details and permissions. Check that these details are correct, and only update any details if you need to. Specific configuration details are under **Client Configuration**, such as the Client ID. The **Permissions** tab lists the available **Scopes** for your organisation.
3. Next, select the **Certificates** tab and select **New Certificate** button. The **New Certificate** screen appears.





Step 3: Create a private key and Certificate Signing Request for a Transport Certificate

1. Select **Transport** from the **Certificate Type** dropdown menu, and then select **Next**.

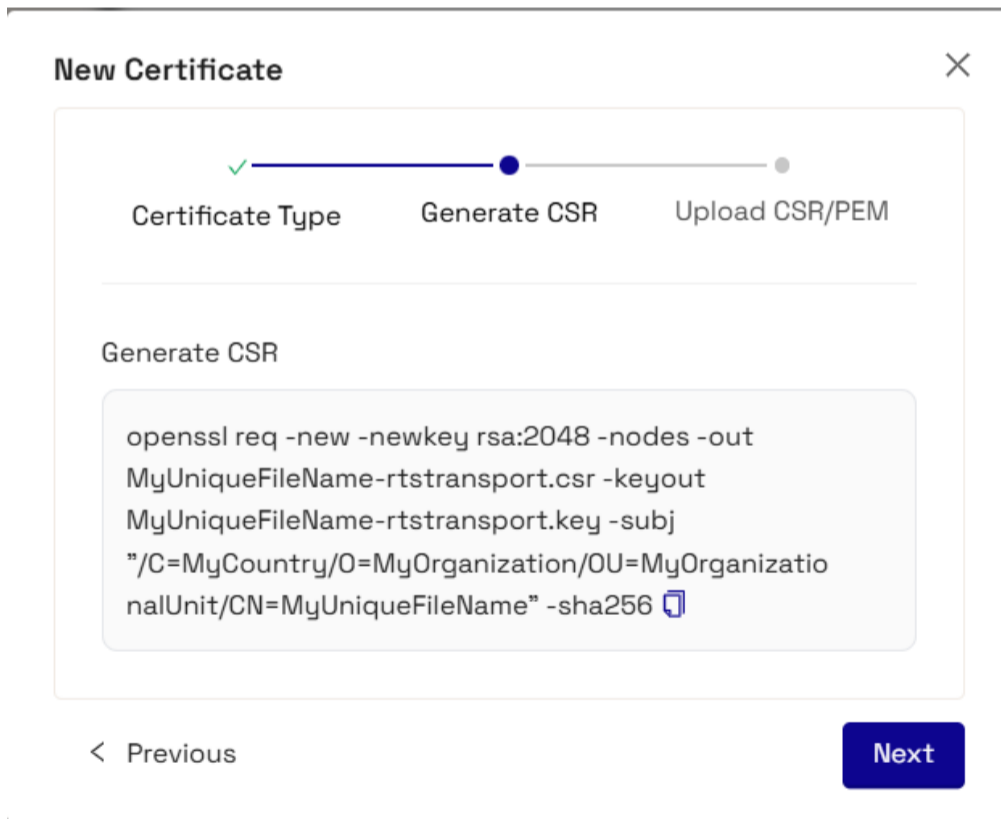


2. Copy the OpenSSL command from the clipboard and run the command on the machine that you have installed OpenSSL on. This command creates a private key and a Certificate Signing Request (CSR). This command is unique to generate a Transport Certificate with the unique `-rtstransport.csr` and `-rtstransport.key` values.
3. You must store the private key securely. Once you have successfully created the private key (.key) and CSR (.csr) files, select **Next**.

Example:

```
openssl req -new -newkey rsa:2048 -nodes -out MyUniqueFileName-rtstransport.csr -keyout MyUniqueFileName-rtstransport.key -subj "/C=MyCountry/O=MyOrganization/OU=MyOrganizationalUnit/CN=MyUniqueFileName" -sha256
```

Note: The following example contains placeholder values, but the command that you copy contains your unique file name for .csr (-out) and .key (-keyout), and organisation information in the certificate subject (-subj).



Step 4: Upload the Certificate Signing Request for your application

1. Select the **Upload CSR** button and select your Certificate Signing Request file. This uploads your Certificate Signing request onto the Thredd platform, so that it can use the file to request the Transport Certificate from the Thredd Certificate Authority (CA).



2. Select **Save**.

If this is successful, Thredd's system adds the newly-issued Transport Certificate to your application. You can verify this in the **App Certificates** screen of the **Certificates** tab for your application.

Step 5: Download the certificate for your application

Next, download the certificate from the **App Certificates** screen. Select the **Actions** menu (three dots) next to the TRANSPORT certificate, and select **Download Certificate**.

Issued	Expiry	Actions
26 Feb 2026 15:34:45	26 Feb 2027 16:34:45	...
26 Feb 2026 15:34:19	26 Feb 2027 16:34:18	...

Download Certificate

Revoke Certificate

Step 6: Configure your Client to use the Transport Certificate in your requests

You must configure your Client to ensure that it includes the Transport Certificate in all calls that it makes to the Thredd platform. To understand how to do this in your specific case, refer to your Client library documentation.

The domain for mTLS connections is: mtls.thredd.com

Step 7: Request a new Signing Certificate (Private Key JWT authentication only)

If you are using Private Key JWT Client Authentication, then your client application also requires a Signing Certificate for signing assertions sent to the Authorisation Server.

Note: If you are using Client Secret authentication, you do not need a Signing Certificate. See [Using the Postman Collection to call REST APIs over mTLS](#).



For Private Key JWT Client Authentication, creating a Signing Certificate is an essential step for enabling access for OAuth 2.0 client applications that connect to Thredd's REST APIs.

There are a few steps that you need to complete after you have requested the Signing Certificate. For example, you must generate a Client Assertion in the form of a JSON Web Token (JWT) and ensure that your client presents it in all calls that it makes to the Thredd REST APIs. For a guide to requesting a Signing Certificate and configuring your client to use Private Key JWT, see [Requesting a Signing Certificate](#).

Next steps

Once you have obtained the necessary credentials for your Client application, you can begin setting up your connection to the API Hub using the latest Postman Collection. See [Using the Postman Collection to call REST APIs over mTLS](#).



Using the Postman Collection to call REST APIs over TLS

Follow these steps to configure Postman with your client credentials so that you can obtain an access token. You must include an access token in all requests to Thredd's REST APIs.

The instructions to obtain an access token are slightly different depending on the Client Authentication method that you have chosen to use.

- Client Secret – requires an access token that you can request from Thredd's OAuth endpoint and within the Postman Collection. You do not require a Signing Certificate.
- Private Key JWT – requires an access token and a JSON Web Token (JWT). You can request the access token from Thredd's OAuth endpoint and within the Postman Collection. To obtain a JWT, you require a Signing Certificate, which an Admin user can obtain from Thredd Portal to create and register a private key. You must generate a Client Assertion, a JSON Web Token (JWT) that is signed by your private key.

Set up summary

You need to complete the following steps to set up Postman to use the Thredd Postman Collection for TLS connections for the first time.

1. Download the Postman Collection.
2. Get variables for the Postman Collection.
3. Set variables in Postman.
4. Test making an API call using the Postman Collection.

Prerequisites

- An Admin user at your organisation must have set up your application that requires access to Thredd APIs in Thredd Portal.
- An Admin user must have requested any certificates that you require from within Thredd Portal.
- You must be able to confirm that you have access by logging in to Thredd Portal.

Step 1: Download the Postman Collection

The Thredd TLS Postman Collection is available for you to download from the Thredd API Hub at: <https://cardsapidocs.thredd.com/v2.0/docs>
Visit the website and make sure that you download the Postman Collection for TLS connections.

Step 2: Get variables for the Postman Collection

Before you can start using the Postman Collection, you must obtain the following variables and enter them in the **Variables** tab in Postman:

- Application ID (UUID) – to enter into the **ssa_id** field
- Client ID – to enter into the **client_id** field
- Signing Certificate (KID) – to enter into the **kid** field ONLY if you using Private Key JWT authentication. If you are using Client Secret authentication leave this blank.

Thredd Portal contains the details for your applications. A Super Admin can get this information from the following areas in Thredd Portal.

Get the Application ID from Thredd Portal

1. Select **System Admin** from the main menu, and then **Applications**. The **Applications** screen displays the applications that you have registered with Thredd.

Note: If the **Applications** list is empty, you need to create an application. See [Creating an Application and OAuth Client](#).

2. Locate your application, and use the copy icon to copy the value under **App ID**.
3. Paste the App ID value into the **ssa_id** field in your Postman Collection.

Get the Client ID from the Overview tab of your application in Thredd Portal

1. Locate the application that you copied the **App ID** from in the **Applications** screen of Thredd Portal in the previous step.
2. Select the **Actions** menu and select **Review Application**. A screen appears with **OAuth Clients** and **Certificates** tabs for your application.



3. Remain on the **OAuth Clients** tab and locate the OAuth Client that you will use to connect to the Thredd REST APIs via the API Hub. Select the **Actions** menu (icon of three dots) and select **View Details**. The **Overview** screen for the OAuth Client appears.
4. On the **Overview** tab, locate the **Client ID** under Client Configuration, and copy the Client ID value. Next, paste it into the **client_id** variable field in your Postman Collection.

Get the KID from the Certificates tab of your application Thredd Portal (Private Key JWT)

If you are using Private Key JWT authentication, you will have requested a Signing Certificate, which generates a KID.

Note: If you are using Client Secret authentication, you do not need a Signing Certificate, which means you do not have or need a KID.

1. Ensure that you are in the details screen for the application that corresponds with the **App ID** and **Client ID** that you copied in the **Applications** screen of Thredd Portal in the previous steps.
2. Select the **Certificates** tab. The Certificates screen displays any certificates available for your application and the **Status**, **KID**, **Key Type** such as **SIGNING**, and **Issued** and **Expiry** dates.
3. Locate the certificate that is the **SIGNING** Key Type, locate the corresponding **KID**, and use the copy icon to copy the KID value. Next, paste it into the **kid** variable field in your Postman Collection.

Step 3: Set variables in Postman

Before you can use the Postman Collection, you must set the following variables in Postman. This ensures that when you make an API call, Postman references these variables and uses the stored values in your API calls. You must not hard-code static values in your API calls.

Variable	Description
<code>client_id</code>	This is either a UUID or a URL with a UUID on the end. If it is an URL you must include the full URL. Example: <code>https://{{domain}}/{{restOfUrl}}</code> If it is a UUID, you must include only the UUID. Example: <code>abcdef01-0124-4567-ffaa-fedcba098731</code>
<code>kid</code>	The Key ID of the Signing Certificate – only if you are using Private Key JWT authentication. If you are using Client Secret authentication, then leave this blank.
<code>private_key</code>	The Signing Certificate Private Key – only if you are using Private Key JWT authentication. You will have created this when you requested the Signing Certificate in Thredd Portal. You must include the first and last lines: <code>-----BEGIN PRIVATE KEY-----</code> and <code>-----END PRIVATE KEY-----</code> If you are using Client Secret authentication, then leave this blank.

Step 4: Check your settings and finish setting up

Make sure that you have configured the following settings for your API calls:

- You have set the correct Base URLs for the API Hub.
- You have obtained an access token.
- Including the access token in the authorisation header.
- Adding an X-Region header – this is mandatory and determines the region that you want to connect to; select one of the following options:
 - `0` for the default environment
 - `1` for the EMEA environment
 - `2` for the APAC environment

Once you have completed these steps, you are ready to make your first API call. For more information, see the [Thredd API Hub](#).

Preparing your Production environment

Once an Admin has verified that your testing in the UAT environment is satisfactory, you can prepare for migrating to a Production environment. You will need to repeat all of these steps for Production, such as requesting new certificates or access tokens, and ensuring that environment-specification configurations are correct. See [Updating Postman](#).



Using the Postman Collection to call REST APIs over mTLS

Follow these steps to configure Postman with your client credentials so that you can obtain an access token. You must include an access token in all requests to Thredd's REST APIs.

The instructions to obtain an access token are slightly different depending on the Client Authentication method that you have chosen to use.

- Client Secret – requires an access token that you can request from Thredd's OAuth Token endpoint and within the Postman Collection. You do not require a Signing Certificate.
- Private Key JWT – requires an access token and a JSON Web Token (JWT). You can request the access token from Thredd's OAuth Token endpoint and within the Postman Collection. To obtain a JWT, you require a Signing Certificate, which an Admin user can obtain from Thredd Portal to create and register a private key. You must generate a Client Assertion, a JSON Web Token (JWT) that is signed by your private key.

Additionally, because you are using an mTLS connection, your application must also present a Transport Certificate, which an Admin user can obtain from Thredd Portal.

Set up summary

You need to complete the following steps to set up Postman to use the Thredd Postman Collection for mTLS for the first time.

1. Download the Postman Collection.
2. Get variables for the Postman Collection.
3. Assign your application's Thredd-issued Transport Certificate to Postman.
4. Set variables in Postman.
5. Test making an API call using the Postman Collection.

Prerequisites

- An Admin user at your organisation must have set up your application that requires access to Thredd APIs in Thredd Portal.
- An Admin user must have requested any certificates that you require from within Thredd Portal. See [Requesting certificates for applications using mTLS](#).
- You must be able to confirm that you have access by logging in to Thredd Portal.

Step 1: Download the Postman Collection

The Thredd mTLS Postman Collection is available for you to download from the Thredd API Hub at: <https://cardsapidocs.thredd.com/v2.0/docs>. Visit the website and make sure that you download the Postman Collection for mTLS connections.

Step 2: Get variables for the Postman Collection

Before you can start using the Postman Collection, you must obtain the following variables and enter them in the **Variables** tab in Postman:

- Application ID (UUID) – to enter into the **ssa_id** field
- Client ID – to enter into the **client_id** field
- Signing Certificate (KID) – to enter into the **kid** field ONLY if you using Private Key JWT authentication. If you are using Client Secret authentication, then leave this blank.

Thredd Portal contains the details for your applications. A Super Admin can get this information from the following areas in Thredd Portal.

Get the Application ID from Thredd Portal

1. Select **System Admin** from the main menu, and then **Applications**. The **Applications** screen displays the applications that you have registered with Thredd.

Note: If the **Applications** list is empty, you need to create an application. See [Creating an Application and OAuth Client](#).



2. Locate your application, and use the copy icon to copy the value under **App ID**.
3. Paste the App ID value into the **ssa_id** field in your Postman Collection.

Get the Client ID from the Overview tab of your application in Thredd Portal

1. Locate the application that you copied the **App ID** from in the **Applications** screen of Thredd Portal in the previous step.
2. Select the **Actions** menu and select **Review Application**. A screen appears with **OAuth Clients** and **Certificates** tabs for your application.
3. Remain on the **OAuth Clients** tab and locate the OAuth Client that you will use to connect to the Thredd REST APIs via the API Hub. Select the **Actions** menu (icon of three dots) and select **View Details**. The **Overview** screen for the OAuth Client appears.
4. On the **Overview** tab, locate the **Client ID** under Client Configuration, and copy the Client ID value. Next, paste it into the **client_id** variable field in your Postman Collection.

Get the KID from the Certificates tab of your application Thredd Portal (Private Key JWT)

If you are using Private Key JWT authentication, you will have requested a Signing Certificate, which generates a KID.

Note: If you are using Client Secret authentication, you do not need a Signing Certificate, which means you do not have or need a KID.

1. Ensure that you are in the details screen for the application that corresponds with the **App ID** and **Client ID** that you copied in the **Applications** screen of Thredd Portal in the previous steps.
2. Select the **Certificates** tab. The Certificates screen displays any certificates available for your application and the **Status**, **KID**, **Key Type** such as **TRANSPORT** or **SIGNING**, and **Issued** and **Expiry** dates.
3. Locate the certificate that is the **SIGNING** Key Type, locate the corresponding **KID**, and use the copy icon to copy the KID value. Next, paste it into the **kid** variable field in your Postman Collection.

Step 3: Download and assign the Transport Certificate to Postman

To use the Thredd mTLS Postman Collection, you need to assign your Transport Certificate from the Thredd CA in Postman. If you downloaded the Transport Certificate when you requested it from Thredd Portal, skip the download instructions and assign it in Postman.

Note: You must complete this process for each of the host URLs and use the same certificates each time. The hosts are: api.uat.threddpay.com and api-uat.thredd.com

If you need to download the Transport Certificate:

1. Locate your application in the **Applications** screen in **System Admin**. Select the **Actions** menu (icon of three dots) and select **Review Application**. A screen appears with tabs for the **OAuth Clients** and **Certificates** for your application.
2. Select the **Certificates** tab. The Certificates screen displays any certificates available for your Application and the **Status**, **KID**, **Key Type** such as **TRANSPORT** or **SIGNING**, and **Issued** and **Expiry** dates.
3. Locate the **TRANSPORT** Key Type. Select the **Actions** menu (icon of three dots) and select **Download Certificate**.
4. This triggers a download for the Transport Certificate file. Locate the file on your computer – you will need to add this into your Postman Collection.

To assign your certificates to the Postman Collection:

1. Select the gears icon in the top-left corner of Postman and select **Settings**.
2. Select **Certificates**.
3. Select **Add Certificate**.
4. Enter the host into the **Host** field. See the **Note** for the list of hosts.
5. Under CRT file, select **Select File** and navigate to the file that you downloaded from Thredd Portal. Double-click to add the file.
6. Under KEY file, select **Select File** under KEY file and navigate to your KEY file. This is the file that you created locally on your machine when you requested it in Thredd Portal. Double-click to add the file.
7. Select **Add**.

When you have added the certificates, your Postman should display the details for each host (HOST, CRT file, KEY file) in **Certificates** screen.

Step 4: Set variables in Postman

Before you can use the Postman Collection, you must set the following variables in Postman. This ensures that when you make an API call, Postman references these variables and uses the stored values in your API calls. You must not hard-code static values in your API calls.



Variable	Description
client_id	This is either a UUID or a URL with a UUID on the end. If it is an URL you must include the full URL. Example: https://{{domain}}/{{restOfUrl}} If it is a UUID, you must include only the UUID. Example: abcdef01-0124-4567-ffaa-fedcba098731
kid	The Key ID of the Signing Certificate – only if you are using Private Key JWT authentication. If you are using Client Secret authentication, then leave this blank.
private_key	The Signing Certificate Private Key – only if you are using Private Key JWT authentication. You will have created this when you requested the Signing Certificate in Thredd Portal. You must include the first and last lines: -----BEGIN PRIVATE KEY----- and -----END PRIVATE KEY----- If you are using Client Secret authentication, then leave this blank.

Step 5: Check your settings and finish setting up

Make sure that you have configured the following settings for your API calls:

- You have set the correct Base URLs for the API Hub.
- You have obtained an access token.
- Including the access token in the authorisation header.
- Adding an X-Region header – this is mandatory and determines the region that you want to connect to; select one of the following options:
 - [0](#) for the default environment
 - [1](#) for the EMEA environment
 - [2](#) for the APAC environment

Once you have completed these steps, you are ready to make your first API call. For more information, see the [Thredd API Hub](#).

Preparing your Production environment

Once an Admin has verified that your testing in the UAT environment is satisfactory, you can prepare for migrating to a Production environment. You will need to repeat all of these steps for Production, such as requesting new certificates or access tokens, and ensuring that environment-specification configurations are correct. See [Updating Postman](#).



Updating Postman

You will need to update Postman if you delete an existing client application and create a new one. An Admin for your organisation must:

- For Client Secret client authentication: Request a new access token (TLS and mTLS).
- For Private Key JWT client authentication: Request a new Signing Certificate from Thredd via Thredd Portal, and request a new access token (TLS and mTLS).
- For mTLS connections only: Request a new Transport Certificate from Thredd via Thredd Portal, and request a new access token.

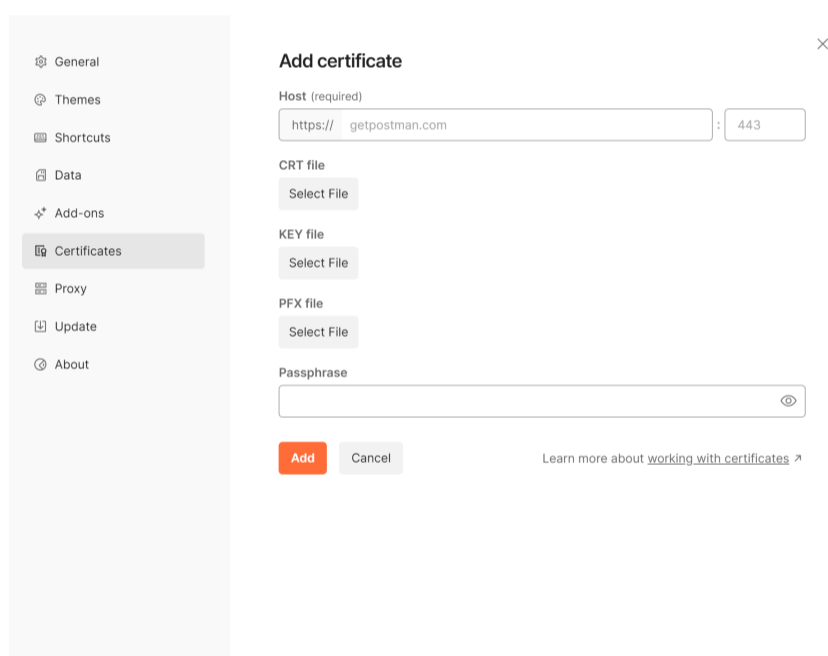
Users who will make calls to Thredd's APIs using the Postman Collection, must update their Postman Collection with the new credentials.

1. Update any access tokens or certificates.
2. Update the variables for the Client ID and other credentials.
3. Use the Postman Collection to obtain a new access token.

Adding Certificates to Hosts for mTLS connections in Postman

If you use an mTLS connection to Thredd REST APIs, you must include a Transport Certificate with the individual hosts. Complete these steps:

1. Select the gears icon and **Settings**.
2. Select **Certificates** from the left-hand menu.
3. Select **Add Certificate**.



4. Enter a host name.
5. Add the CRT and Key files for the host.
6. Select the **Add** button.

For more details, see [Using the Postman Collection to call REST APIs over mTLS](#).

Base URLs

When calling Thredd endpoints, you must include the Base URLs within your REST tool. The Base URLs are the hosts that accept the certificates. The Cards API and API Hub hosts have unique URLs that are different between UAT and Production.

UAT environments use the following Base URLs:

- **Cards API:** <https://api.uat.threddpay.com/api/v1>
- **API Hub over TLS:** <https://uat-api.thredd.com>
- **API Hub over mTLS:** <https://uat-mtls.thredd.com>
- **Thredd Certificate Authority:** <https://matls-auth.directory.sandbox.threddid.com>

Production environments use the following Base URLs:

- **API Hub over TLS:** <https://api.thredd.com> (for all PRD environments)
- **API Hub over mTLS:** <https://mtls.thredd.com> (for all PRD environments)



- **Cards API:** <https://coreapi.threddpay.com>



Section 4: SOAP

You should read this section to understand setting up certificates for SOAP web services.

Topics covered in this section:

- [Creating Organisation Transport Certificates for SOAP Web Services](#)



Creating Transport Certificates for SOAP Web Services

This page describes how you create Transport Certificates for accessing Thredd's SOAP Web Services. As Thredd's SOAP Web Services are secured using Mutual Transport Layer Security (MTLS), your *client application* must present a trusted Transport Certificate for authentication.

The instructions for obtaining certificates are the same as those for clients using Thredd's REST APIs.

To obtain certificates, you must first set up a Super Admin user in Thredd Portal, create an application and OAuth Client, and then request certificates for it. See [Requesting certificates for applications using mTLS](#).

Convert the certificate and key to PKCS#12 syntax

If required, for example you are using Windows services, you can convert the public certificate and private key to the PKCS#12 syntax. For converting, you can use OpenSSL commands. In the PKCS#12 syntax, the files are in the .pfx format. For more details, see [RFC 7292: PKCS #12: Personal Information Exchange Syntax v1.1](#).



Section 5: EHI Setup

You should read this section to understand the set up steps for EHI.

Topics covered in this section:

- [Setting up EHI for TLS connections and signed payloads](#)
- [Setting up EHI for mTLS connections](#)



Setting up EHI for TLS connections and signed payloads

The External Host Interface (EHI) offers a way to exchange transactional data between the Thredd processing system and the Program Manager's externally-hosted systems. All transaction data processed by Thredd is transferred to the external host system via EHI in real time.

EHI provides two main functions:

- a real-time transaction notification data feed
- payment authorisation control

Follow these steps to set up the External Host Interface (EHI) to communicate with Thredd using TLS and signed payloads.

Overview of TLS in EHI

In the EHI communication flow, your application is the Server and Thredd is the Client.

When Thredd (the Client) connects to your Server, you must present a Server Certificate to Thredd during the TLS handshake. This allows Thredd (the Client) to verify your certificate. The Server certificate is a TLS certificate (the more modern version of an SSL certificate).

Thredd's EHI messages are digitally signed to ensure their integrity and authenticity. These messages are sent over HTTPS for transport layer encryption using Transport Layer Security (TLS).

Thredd's Public Key Infrastructure (PKI) provides a JSON Web Key Set (JWKS) endpoint that you can use to validate EHI payload signatures. This allows the Server to receive signed payloads over the secure TLS connection.

Digital Signature Flow

The following diagram depicts the Digital Signature Flow; note that CSR = Certificate Signing Request.

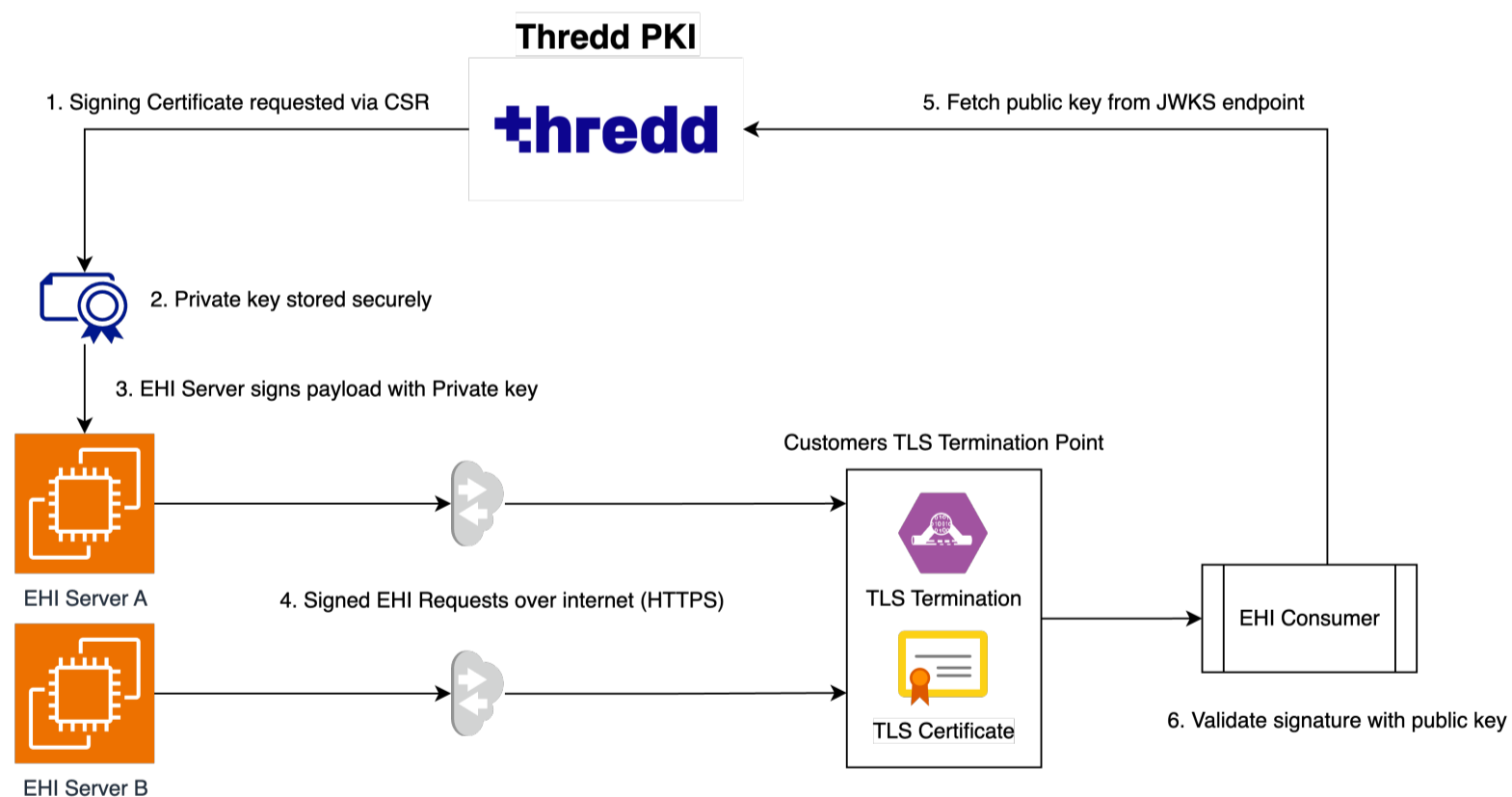


Figure: Thredd PKI Signing Flow

Configuring your Server to use EHI with signed payloads

Summary of the steps you need to complete:

1. Obtain a Server Certificate from a Certificate of Authority vendor and install the certificate on your EHI listening endpoint.
2. Log in to Thredd Portal and obtain the JWKS endpoint details.
3. Integrate with the JWKS endpoint.
4. Implement signature validation logic.



5. Test your EHI endpoint for TLS communication.
6. Provide the EHI endpoint to and inform Thredd that you are ready to use the EHI application.

Prerequisites

- You must have the Super Admin or Organisation Admin role for your organisation in Thredd Portal.
- You must have access to Thredd Portal.

Step 1: Obtain and install a Server Certificate

You first need to set up the Server certificate on your EHI server.

1. Obtain a Server Certificate from a Certificate of Authority vendor of your choice, such as Verizon or Digicert or Amazon Web Services.
2. Install the Server certificate on your EHI listening endpoint.
3. Ensure that your Server presents this certificate to Thredd during the TLS handshake.

To understand how to do this in your specific case, refer to your Client library documentation.

Step 2: Log in to Thredd Portal and obtain the JWKS endpoint details

Thredd's Public Key Infrastructure (PKI) provides a JSON Web Key Set (JWKS) endpoint that you can use to validate EHI payload signatures.

A Super Admin user must log in to the Thredd Portal and obtain the JWKS endpoint details.

To view your JWKS endpoint in Thredd Portal:

1. Log in to Thredd Portal.
2. Navigate to the **System Admin** menu, select **Organisation**, and then select **EHI Configuration**.
3. The EHI Configuration page opens. Locate the JWKS endpoint and copy its URI.

The address of the JWKS endpoint should be similar to the following:

https://jwks.threddid.com/{thredd_org_id}

Note: You must replace the placeholder `{thredd_org_id}` in the example JWKS endpoint URL with your unique organisation ID.

Step 3: Integrate with the JWKS endpoint

Every organisation registered with Thredd's IAM platform is assigned a unique, public JWKS endpoint. You must configure your application to securely access and consume the JSON Web Key Set (JWKS) from Thredd's JWKS endpoint, which exposes all of Thredd's signing keys.

Use the endpoint that corresponds with the environment that you are configuring:

UAT:

<https://jwks.threddid.com/daf9e877-fa87-44b0-9934-8a7a39503586/jwks.json>

Production:

<https://jwks.threddid.com/a7841563-97ec-4885-ac7b-cdb87e9d1024/jwks.json>

Your security and network teams must ensure that the network environment of the application (that receives EHI messages) has unrestricted outbound access to the provided JWKS endpoint.

Your system must be able to periodically query this public endpoint to retrieve the current public signing keys. Thredd recommends that you implement caching logic to store the keys temporarily to reduce the need to call this endpoint, and define a clear refresh mechanism.

Step 4: Implement signature validation logic

To verify the authenticity and integrity of payloads, for example verifying that payload data has not been altered, you should implement logic to validate payload signatures. Signature validation is conducted at the application layer.

An overview of the logic is:



1. Locate and parse the signature – find the digital signature and associated Key ID (*kid*) in the *X-Thredd-Signature* header.
2. Retrieve the matching public key – use the *kid* from the signature to look up the corresponding Public Key from the JWKS endpoint or cached JWKS data.
3. Validate the signature – use the Public Key that you retrieved from the JWKS endpoint or cached data to cryptographically verify the digital signature against the EHI payload content.
 1. Successful validation: Guarantees the integrity and authenticity of the payload.
 2. Validation failure: Immediately rejects the notification and logs a critical security incident.

Step 5: Test your EHI endpoint for TLS communication

When you are ready, test the EHI endpoint with the online SSL Labs tool and OpenSSL. This is to ensure that you can successfully communicate with EHI over TLS.

When you have completed testing, provide the EHI endpoint to Thredd.

Note: You must not provide the EHI endpoint if you have not completed testing.

Test using SSL Labs

1. Go to the URL of the tool: <https://www.ssllabs.com/ssltest/>
2. Enter the URL for testing in the SSL Labs test screen test page. For example, api.thredd.com. The results appear similar to the following:

The screenshot shows a web browser window displaying the SSL Labs report for api.thredd.com. The report is titled "SSL Report: api.thredd.com" and was assessed on Wednesday, 05 Mar 2025 at 12:08:19 UTC. A "Scan Another >>" link is visible. The report contains a table with three rows, each representing a server. All three servers received an "A" grade.

	Server	Test time	Grade
1	51.24.40.63 ec2-51-24-40-63.eu-west-2.compute.amazonaws.com Ready	Wed, 05 Mar 2025 12:04:26 UTC Duration: 77.515 sec	A
2	18.168.174.19 ec2-18-168-174-19.eu-west-2.compute.amazonaws.com Ready	Wed, 05 Mar 2025 12:05:43 UTC Duration: 80.185 sec	A
3	18.133.217.216 ec2-18-133-217-216.eu-west-2.compute.amazonaws.com Ready	Wed, 05 Mar 2025 12:07:03 UTC Duration: 75.356 sec	A

SSL Report v2.3.1

Figure: Passing Tests on SSL Labs

An "A" Grade results in the test passing. However, "B" does not result in a pass, and can indicate that the problem is due to a missing certificate on your Server.

Test using OpenSSL

You should run the following command that triggers the TLS handshake for the communication between Server (you) and the Client (Thredd). You receive a response for the Server certificate.

```
openssl s_client -connect ehi.yourdomain.com:443
```

Server Certificate result

The results for the Server certificate appear as follows:



```
CONNECTED(00000006)
depth=2 C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert Global Root G2
verify return:1
depth=1 C=US, O=DigiCert Inc, CN=DigiCert Global G2 TLS RSA SHA256 2020 CA1
verify return:1
depth=0 C=GB, L=London, O=UK Limited, CN=*.com
verify return:1
---
```

Figure: Server Certificate Result Including Depth Settings

If [Depth 0,1,2](#) all show [verify return:1](#) this indicates that the EHI servers trust the Server Certificate. There could be an issue with Server certificate result, if the results are different; for example, if there is no [verify return:1](#) for the depth settings. The Depth settings mean the following:

Depth Setting	Description
Depth 0 = Root	The Root certificate has been sent.
Depth 1 = Intermediate:	The Leaf certificate has been sent.
Depth 2 = Root:	The Root certificate has been sent.

Note: You might need to check the EHI listener endpoint configuration if the SSL Labs test does not give an "A" Grade response.

Testing requests without a certificate

For additional testing, you can test a request to the EHI endpoint that does not contain a certificate. Using cURL, you see a 400 or 403 response as in the following example.

```
# Request
curl -v https://api.yourdomain.com/ehi/endpoint/api
# Partial Response
<html>
<head><title>400 No required SSL certificate was sent</title></head>
<body>
<center><h1>400 Bad Request</h1></center>
<center>No required SSL certificate was sent</center>
```

Step 6: Inform Thredd that you are ready to use the EHI application

Once you have completed testing, provide the EHI endpoint to Thredd.

Note: You must not provide the EHI endpoint if you have not completed testing.

Thredd confirms to you when you are ready to continue to setting up your integration with EHI. See one of the following guides to learn more about EHI:

- [EHI Guide \(JSON version\)](#)
- [EHI Guide \(XML version\)](#)



Setting up EHI for mTLS connections

The External Host Interface (EHI) offers a way to exchange transactional data between the Thredd processing system and the Program Manager's externally-hosted systems.

All transaction data processed by Thredd is transferred to the external host system via EHI in real time.

An EHI with mTLS (EHIm) setup comprises an integration with an external payments network that itself is authenticated using Mutual TLS (mTLS).

EHI provides two main functions:

- a real-time transaction notification data feed
- payment authorisation control

mTLS relies on a system of digitally-signed certificates issued by a trusted third party called a Certificate Authority (CA). The CA proves the authenticity of the public key and the identity of the Server presenting the key.

Here, you can follow the steps to set up the External Host Interface (EHI) to communicate with Thredd using mTLS.

Note: If you are currently presenting a certificate that has been issued by Raidiam Connect, you must replace it with a certificate that Thredd has issued and migrate your EHIm setup to Thredd. See the EHIm Migration Guide.

Overview of mTLS in EHI

In the EHIm (EHI + mTLS) communication flow, your application is the *Server* and Thredd is the *Client*. Both parties must prove their identities to each other for authentication by presenting their respective certificates to each other, and verifying this during the mTLS handshake.

The high-level communication flow is as follows:

1. The Client (Thredd) connects to the Server (you).
2. The Server (you) presents your Server certificate to the Client (Thredd). The Server certificate is a TLS certificate (the more modern version of an SSL certificate).
3. The Client (Thredd) verifies your certificate.
4. The Client (Thredd) presents its certificate.
5. The Server (you) verifies the Client's (Thredd's) certificate by matching it against the information held in Thredd's Certificate Authority's Trust Chain.

Once both certificates are verified, the Server grants access. The Client and Server then exchange information over the secure connection.

In order to enable this you need to install the Thredd EHIm Trust Chain on your EHI Server. The EHI Server must validate the Client Certificates that Thredd sends to you. This ensures that a Chain of Trust is established.

A Trust Chain is a collection of trusted root certificates that are installed on the Server, which includes the root anchor (top level) and intermediate Certificate Authority (CA) certificates. An end entity uses these to establish a Chain of Trust, by checking the certificate chain, and validates the Client Certificate. The Server uses the Certificate Chain to prove the legitimacy of a Client Certificate by tracing it back to a single Root CA.

What is mTLS?

Mutual Transport Layer Security (Mutual TLS or mTLS) is an enhanced security protocol that ensures that both the Client and the Server verify each other's identity before establishing a secure communication channel. This is a digital handshake where both parties must provide their official identity. To learn more about mTLS, see [Mutual TLS explained](#).

Configuring your Server to use EHI over mTLS

You need to obtain, install and store the certificates on your EHI Server, which form the trust store that the Chain of Trust refers to when validating the path and certificates.

To ensure this, you must:

- Obtain a Server Certificate from a Certificate of Authority vendor, such as Verizon or Digicert or Amazon Web Services, and install it on your EHI listening point.



- Download and install Thredd's EHI Trust Chain on your Server, and store the CA's Root and Issuing Certificates on your mTLS termination point. You must configure your systems to ensure that your Server presents this certificate to the Client (Thredd) during the mTLS handshake.
- Ensure that your mTLS termination point has access to Thredd's Online Certificate Status Protocol (OCSP) responder.
- Test the EHI endpoint.
- Provide the EHI endpoint to and inform Thredd that you are ready to use the EHI application.

Prerequisites

- You must have the Super Admin or Organisation Admin role for your organisation in Thredd Portal.
- You must have access to Thredd Portal.
- You must have obtained and installed a Server Certificate from a Certificate of Authority vendor, such as Verizon or Digicert or Amazon Web Services.

Step 1: Download and install Thredd's EHI Trust Chain

You must install the EHI Trust Chain on your Server and configure your systems to ensure that your Server presents its Server Certificate to the Client (Thredd) and validates against the EHI Trust Chain during the mTLS handshake.

To download the Trust Chain:

1. Log in to Thredd Portal.
2. Navigate to and select **System Admin**, select **Organisation**, and then select **EHI Configuration**.
3. The EHI Configuration page opens. Locate the certificate/trust chain file under **Certificate Chain**, and download it. If a certificate chain is not available to download, contact Thredd for support.

Next, configure your mTLS termination point (your load balancer, proxy, or server) settings to validate against the Thredd EHI Trust Chain. Point it to or copy the contents of the Thredd EHI Trust Chain file.

Before you proceed, you must ensure that:

- Your mTLS termination point is validating against the Thredd EHI Trust Chain file.
- You store the CA's Root and Issuing Certificates on your mTLS termination point.

Refer to your server's technical documentation for information about how to complete this task.

Step 2: Ensure that your mTLS termination point has access to Thredd's Online Certificate Status Protocol (OCSP) Responder

Thredd recommends that you configure your mTLS termination point (load balancer, proxy, or server) to carry out certificate status checks. For example, to check if the certificate has been revoked.

This is optional. However, if you want to ensure that it can do this, your mTLS termination point must have access Thredd's Online Certificate Status Protocol (OCSP) Responder. The OCSP Responder provides both clients and Thredd with the ability to verify the validity of certificates issued within Thredd's private PKI hierarchy, and in real time.

Check the settings for your mTLS termination point:

1. Make sure that your mTLS termination point can access Thredd's OCSP Responder at: <https://ocsp.threddid.com>
2. Verify and change the settings if necessary. For example, by adding the OCSP Responder to the allowlist.
3. Refer to your server's documentation for information about how to check and change your mTLS termination point settings.

Step 3: Test the EHI endpoint

Once you are ready, test the EHI endpoint with the online SSL Labs tool and OpenSSL. This is to ensure that you can successfully communicate with EHI over mTLS.

Once you have completed testing, provide the EHI endpoint to Thredd, as per step 4.

Note: You must not provide the EHI endpoint if you have not completed testing.



Test using SSL Labs

1. Go to the URL of the tool: <https://www.ssllabs.com/ssltest/>
2. Enter the URL to be tested in the SSL Labs test screen test page. For example, mtls.thredd.com. The results appear similar to the following:

The screenshot shows the SSL Labs report for api.thredd.com. The report is titled "SSL Report: api.thredd.com" and was assessed on Wed, 05 Mar 2025 12:08:19 UTC. A "Scan Another >>" link is visible. The main table contains the following data:

	Server	Test time	Grade
1	51.24.40.63 ec2-51-24-40-63.eu-west-2.compute.amazonaws.com Ready	Wed, 05 Mar 2025 12:04:26 UTC Duration: 77.515 sec	A
2	18.168.174.19 ec2-18-168-174-19.eu-west-2.compute.amazonaws.com Ready	Wed, 05 Mar 2025 12:05:43 UTC Duration: 80.185 sec	A
3	18.133.217.216 ec2-18-133-217-216.eu-west-2.compute.amazonaws.com Ready	Wed, 05 Mar 2025 12:07:03 UTC Duration: 75.356 sec	A

SSL Report v2.3.1

Figure: Passing Tests on SSL Labs

An "A" Grade results in the test passing. However, "B" will not result in a pass, and can indicate that the problem is due to a missing an Immediate or Root certificate on your Server.

Test using OpenSSL

You should run the following command that triggers the TLS handshake for the communication between Server (you) and the Client (Thredd). You receive responses for the Server and Client certificates.

```
openssl s_client -connect ehi.yourdomain.com:443
```

Server Certificate result

The results for the Server certificate appear as follows:

```
CONNECTED(00000006)
depth=2 C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert Global Root G2
verify return:1
depth=1 C=US, O=DigiCert Inc, CN=DigiCert Global G2 TLS RSA SHA256 2020 CA1
verify return:1
depth=0 C=GB, L=London, O=UK Limited, CN=*.com
verify return:1
---
```

Figure: Server Certificate Result Including Depth Settings

If Depth 0,1,2 all show `verify return:1` this indicates that the EHI servers trust the Server Certificate. There could be an issue with Server certificate result, if the results are different; for example, if there is no `verify return:1` for the depth settings. The Depth settings mean the following:

Depth Setting	Description
Depth 0 = Root	The Root certificate has been sent.



Depth Setting	Description
Depth 1 = Intermediate:	The Leaf certificate has been sent.
Depth 2 = Root:	The Root certificate has been sent.

You should configure your Server's TLS setup so that it sends the Server Certificate and the Intermediate Certificate. If the Server sends only the Server certificate then the Chain of Trust is not complete; the mTLS connection is not set up between the Server and the Client.

Client Certificate result

For validating the Client Certificate, certificates that Thredd users appear under [Acceptable client certificate CA names](#). There may be an issue if the [Acceptable client certificate CA names](#) section is empty, where Thredd's CAs are not listed.

```

-----END CERTIFICATE-----
subject=C=GB, L=London, O=, CN=,com
issuer=C=US, O=DigiCert Inc, CN=DigiCert Global G2 TLS RSA SHA256 2020 CA1
---
Acceptable client certificate CA names
C=GB, O=Thredd UK Limited, OU=Thredd Directory, CN=Thredd Root CA - G1
C=GB, O=Thredd UK Limited, OU=Thredd Directory, CN=Thredd Issuing CA - G1
Client Certificate Types: RSA sign, DSA sign, ECDSA sign
Requested Signature Algorithms: ECDSA+SHA256:ECDSA+SHA384:ECDSA+SHA512:UNDEF:Ed25519:Ed448:RSA-PSS+SHA256:RSA-P
SS+SHA384:RSA-PSS+SHA512:RSA-PSS+SHA256:RSA-PSS+SHA384:RSA-PSS+SHA512:RSA+SHA256:RSA+SHA384:RSA+SHA512:ECDSA+SH
A224:ECDSA+SHA1:RSA+SHA224:RSA+SHA1:DSA+SHA224:DSA+SHA1:DSA+SHA256:DSA+SHA384:DSA+SHA512
Shared Requested Signature Algorithms: ECDSA+SHA256:ECDSA+SHA384:ECDSA+SHA512:Ed25519:Ed448:RSA-PSS+SHA256:RSA-
PSS+SHA384:RSA-PSS+SHA512:RSA-PSS+SHA256:RSA-PSS+SHA384:RSA-PSS+SHA512:RSA+SHA256:RSA+SHA384:RSA+SHA512:ECDSA+S
HA224:RSA+SHA224:DSA+SHA224:DSA+SHA256:DSA+SHA384:DSA+SHA512
Peer signing digest: SHA256
Peer signature type: RSA-PSS
Server Temp Key: X25519, 253 bits
---
SSL handshake has read 6235 bytes and written 443 bytes
Verification: OK
---
New, TLSv1.2, Cipher is ECDHE-RSA-AES128-GCM-SHA256
Protocol: TLSv1.2
Server public key is 2048 bit

```

Figure: Client Certificate Response

Note: Listing the CA authority is optional. If this is empty, it might not mean that the mutual part of the certificate validation has failed.

Note: You might need to check the EHI listener endpoint configuration if the following has occurred:

- If the SSL Labs test does not give an "A" Grade response.
- The Server Certificate does not show [verify return:1](#) at all depths.
- The Acceptable client certificate CA names are not sent.

Testing requests without a certificate

For additional testing, you can test a request to the EHI endpoint that does not contain a certificate. Using cURL, you see a 400 or 403 response as in the following example.

```

# Request
curl -v https://api.yourdomain.com/ehim/endpoint/api
# Partial Response
<html>
<head><title>400 No required SSL certificate was sent</title></head>
<body>
<center><h1>400 Bad Request</h1></center>
<center>No required SSL certificate was sent</center>

```

Testing requests with a certificate

You can also test a request to the EHI endpoint that contains a certificate. Using cURL, you see a 200 response as in the following example.

```

# Request
curl --cert ./exampleClientCert.pem --key ./exampleClientCert.key \
-X POST \
-H "Content-Type: application/json" \
-d '{"foo": "bar"}' \
-v https://api.yourdomain.com/ehim/endpoint/api

```



Partial Response
200 OK

Step 4: Inform Thredd that you are ready to use the EHI application

Once you have completed testing, provide the EHI endpoint to Thredd.

Note: You must not provide the EHI endpoint if you have not completed testing.

Thredd will confirm to you when you are ready to continue to setting up your integration with EHI. See one of the following guides to learn more about EHI:

- [EHI Guide \(JSON version\)](#)
- [EHI Guide \(XML version\)](#)

Message Architecture between Thredd and the Customer EHI

The following shows the message architecture between Thredd and your EHI components (the EHI Customer). The EHI listener endpoint and the mTLS termination point are part of your EHI components.

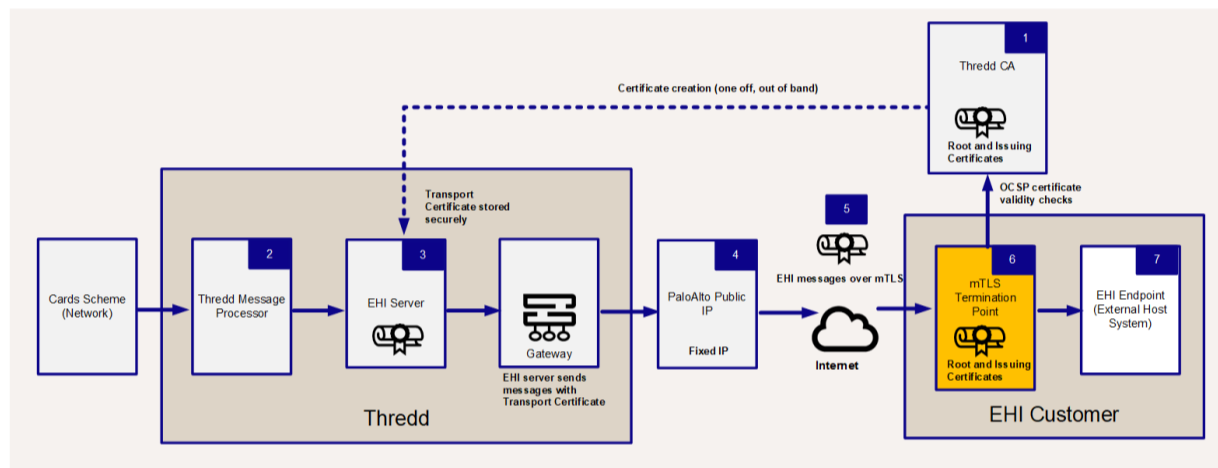


Figure: Message Architecture Between Thredd and EHI

1. Thredd uses the *Certificate Authority*, Thredd Certificate Authority, to create a Transport Certificate to enable communication later over mTLS.
2. Thredd's Message Processor sends and processes EHI messages that are received from the Cards Schemes (Networks) to Thredd's EHI servers.
3. Thredd sends the EHI messages via a gateway and adds the Transport Certificate.
4. EHI requests travel across the Internet with the associated Transport Certificate attached. Note that Thredd's outgoing firewall has a fixed IP address.
5. The Transport Certificate is presented to your (the Server's) incoming mTLS termination point during the connection handshake.
6. Your systems validate the Transport Certificate by checking it against the CA chain of trust (Root and Issuing Certificates).
7. When the mTLS handshake is complete, you receive EHI messages on your EHI endpoint (External Host System).

Adding Security Controls

You can choose to implement additional optional controls to enhance mTLS security. These include the:

- **IP Address Allowlist** – Thredd EHI messages are sent from a firewall with a fixed IP address. Optionally, you can add this IP address to your allowlist.
- **Certificate Pinning** – the mutual communication, between your Server and Thredd (the Client), is secured by Thredd's Transport Certificate. Your Server only trusts the Transport Certificate if it the Thredd CA has issued it. However, you can choose to also implement Certificate Pinning. Certificate Pinning blocks attempted requests made with the incorrect certificates. For more information, see [Certificate and Public Key Pinning | OWASP Foundation](#).



Section 6: Configuring an identity provider for SSO

You should read this section to understand the setup steps for Single Sign-On (SSO) with your chosen identity provider.

Topics covered in this section:

- [Configuring SSO with Google \(SAML\)](#)
- [Configuring SSO with Okta \(SAML\)](#)
- [Configuring SSO with Okta \(OIDC\)](#)



Configuring SSO with Okta (SAML)

If your organisation uses Okta as an IdP, you can use Okta for configuring SSO for accessing various Thredd services, for example, Thredd Portal. This guidance describes the steps for using the 2.0 version of Security Assertion Markup Language (SAML) protocol for setting up SSO.

As a client, you would already have an account on the Okta Administration Console.

Note: Setting up SSO is not mandatory, but Thredd recommends it.

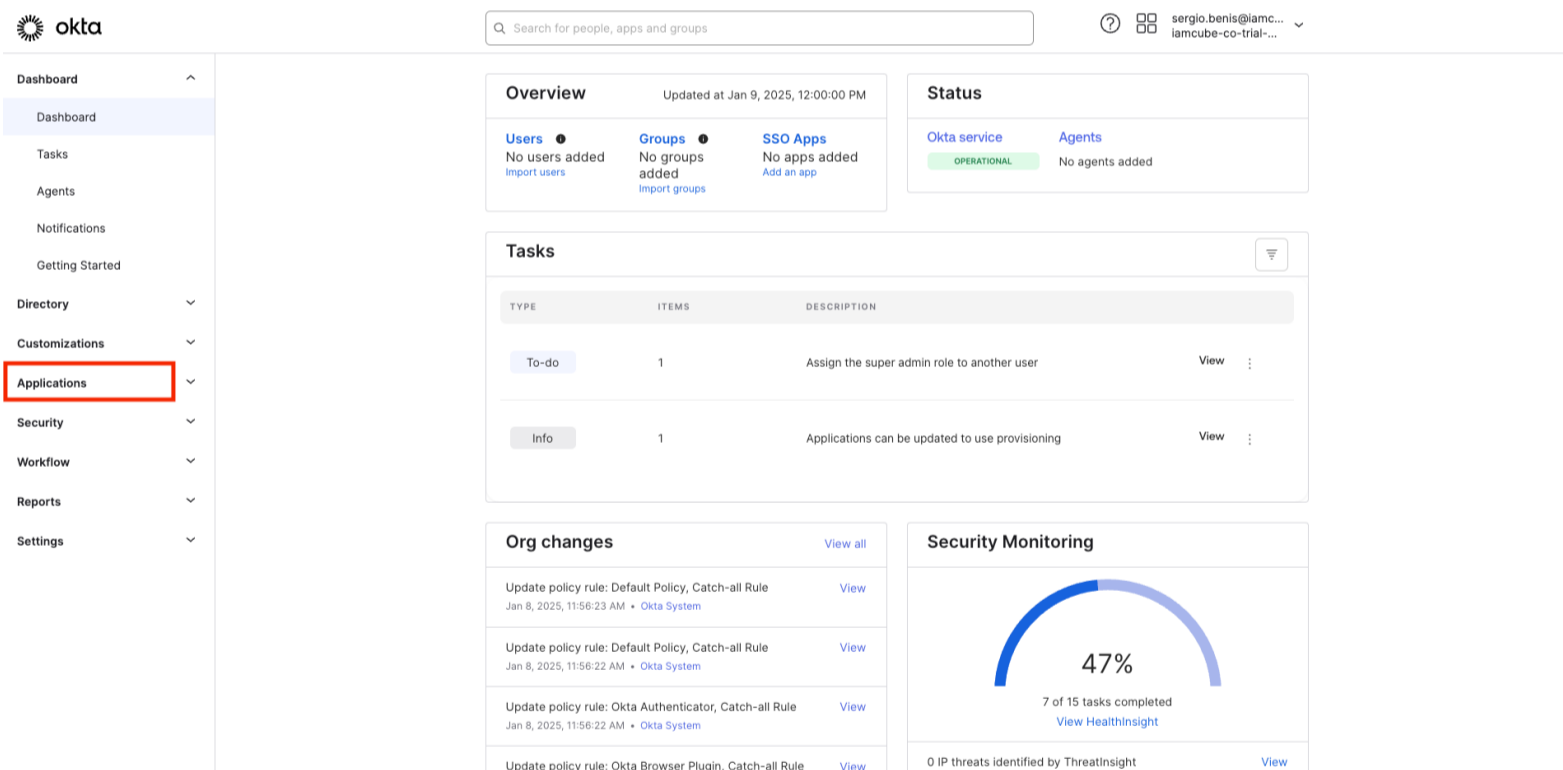
Summary of the steps

The steps involve:

- Creating a SAML app for your SSO connection to Thredd services.
- Adding configurations from Thredd including the SSO URLs and the Entity ID.
- Mapping fields associated with the users defined by Okta with those used by your app.
- Sharing the Metadata URL with Thredd.

Configuring SSO

1. Log in to the Okta Administration console.
2. From the left-hand menu, select **Applications**.



3. Select **Create Application**.
4. Select **SAML 2.0** and select **Next**. The next page appears.



Create a new app integration



Sign-in method

[Learn More](#)

- OIDC - OpenID Connect**
Token-based OAuth 2.0 authentication for Single Sign-On (SSO) through API endpoints. Recommended if you intend to build a custom app integration with the Okta Sign-In Widget.
- SAML 2.0**
XML-based open standard for SSO. Use if the Identity Provider for your application only supports SAML.
- SWA - Secure Web Authentication**
Okta-specific SSO method. Use if your application doesn't support OIDC or SAML.
- API Services**
Interact with Okta APIs using the scoped OAuth 2.0 access tokens for machine-to-machine authentication.

Cancel

Next

6. Enter a name for the app that accesses Thredd services in **App name** and select **Next**. The next page appears.

7. Add the provided URLs in **Single Sign-on URL (ACS URL)** and **Audience URL (entity ID)**.

8. Scroll down on the same page and configure the following Attribute Statements:

- a. Enter an attribute name in the **Name** column.
- b. Select a value in the **Value** column.
- c. To add another entry, select the **Add Another** button.
- d. Repeat steps a and b.



Attribute Statements (optional)

[LEARN MORE](#)

Name	Name format (optional)	Value
firstname	Unspecified ▼	user.firstName ▼
lastname	Unspecified ▼	user.lastName ▼ ✕
email	Unspecified ▼	user.email ▼ ✕

[Add Another](#)

9. Select **Next**.

10. In the displayed page, select **This is an internal app that we created** and select **Finish**.

11. In the displayed **Metadata Data** details that appear, share the Metadata URL with Thredd.

You can then assign the application to the users or groups who will be using the Thredd services.



Configuring SSO with Google (SAML)

If your organisation uses Google, you can configure Google as an IdP provider to provide SSO access to various Thredd services. For example, you can use SSO to access Thredd Services, such as Thredd Portal. This guidance describes the steps for using the 2.0 version of Security Assertion Markup Language (SAML) protocol for setting up SSO. As a client, you would already have an account on the Google Admin Console.

Note: Setting up SSO is not mandatory, but Thredd recommends it.

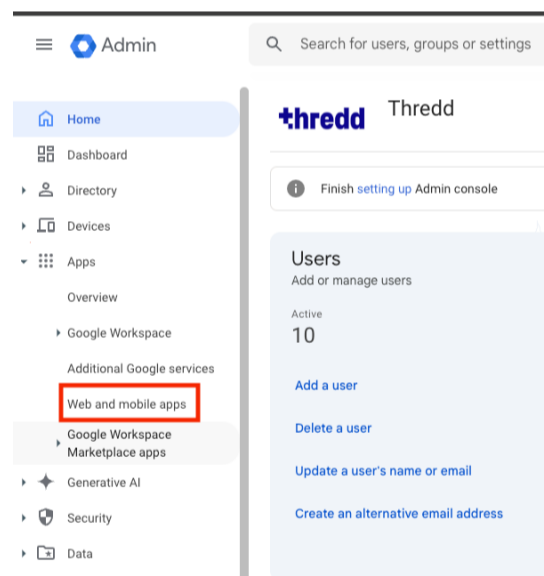
Summary of steps

The steps involve:

- Creating a SAML app for your SSO connection to Thredd services.
- Choosing either to download IdP metadata or to add configurations from Thredd. If you add configurations from Thredd, you include the SSO URL and the Entity ID.
- Mapping fields associated with the users defined by Google to those used by your app.
- Assigning access permission on your app.

Configuring SSO

1. Log in to the Google Admin console.
2. Select **Apps > Web and mobile apps**.



3. Select **Add app** and select **Add custom SAML app**.
4. Enter a name for the app that accesses Thredd services in **App name** and select **Continue**. The next page appears.




App details
Enter details for your custom SAML app. This information is shared with app users. [Learn more](#)

App name
Thredd Services

Description

App icon
Attach an app icon. Maximum upload file size: 4 MB



CANCEL CONTINUE

5. To download the metadata, select **Download Metadata** and save the file. Then share the file with Thredd. Proceed to step 7.

6. To include entity ID and URL details:

a. Add the URL in **SSO URL**.

b. Add in the Entity ID in **Entity ID**. A certificate and a SHA-256 fingerprint appear. These are generated automatically on the console.

To configure Single Sign-On (SSO) for SAML apps, follow your service provider's instructions [Learn more](#)

Option 1: Download IdP metadata

DOWNLOAD METADATA

OR

Option 2: Copy the SSO URL, entity ID and certificate

SSO URL
https://accounts.google.com/o/saml2/idp?idpid=C024b6hqr

Entity ID
https://accounts.google.com/o/saml2?idpid=C024b6hqr

Certificate
Google_2028-7-26-74014_SAML2_0
Expires 26 Jul 2028

```

---BEGIN CERTIFICATE---
MIIDdDCCAlYgAwIBAgjGAYmc8W61MA0GCSqGSIb3DQEBCwUAMHsxFDASBgNVBAoTC0d0vb2dsZSBj
b2dsZSBGb3JlV29yazELMAkGA1UEBhMCVVMxEzARBgNVBAGTCkNhbGlib3JuaWEwHhcNMjMwNzI4
b2dsZSBGb3JlV29yazELMAkGA1UEBhMCVVMxEzARBgNVBAGTCkNhbGlib3JuaWEwHhcNMjMwNzI4

```

SHA-256 fingerprint
13:14:C9:7F:D2:D4:A6:E0:A8:BE:EA:F9:03:60:79:56:86:6D:E0:11:40:EF:FD:1B:2F:46:CA:60:5E:2D:BE:53

BACK

CANCEL CONTINUE

7. Select **Continue**.

8. Configure attribute mapping.

a. Select the **Add Mapping** button.

b. Select a group in **Google directory attributes** and choose a group in **App attributes**.

c. To add another entry, select the **Add Mapping** button again and repeat the step for choosing both attributes.

The following shows an example.



Attributes
Add and select user fields in the Google Directory, then map them to service provider attributes. Attributes marked with * are mandatory. [Learn more](#)

Google directory attributes		App attributes	
Basic Information > First name	→	firstname	×
Basic Information > Last name	→	lastname	×
Basic Information > Primary email	→	email	×

[ADD MAPPING](#)

Group membership (optional)
Group membership information can be sent in the SAML response if the user belongs to any of the groups that you add here.

Google Groups		App attribute
Search for a group	→	Groups

BACK CANCEL [FINISH](#)

9. Select **Finish**.

10. Once completed, select access permission options (see the following procedure).

Setting Access Permission Options

You can set access permission options for the app based on anyone who holds a Google account, membership of specific Google groups, and organisational units. An organisational unit is a named organisation within Google.

1. To provide access to anyone who holds a corporate Google account, select **All users in this account** on the left hand menu. Then choose **ON for everyone** in the main screen.

Apps > Web and mobile apps > Thredd Services > Service status

Thredd Services

All users in this account

Showing settings for users in all organisational units

Service status:

Service status:

ON for everyone

OFF for everyone

Most changes take effect within a few minutes. [Learn more](#)

CANCEL SAVE

Organisational units

2. To provide access to members of a selected group:

- a. Select **Groups** on the left hand menu.
- b. Select a group.
- c. Select **ON for everyone** in the main screen.



The screenshot shows the 'Service status' configuration page for 'all organisational units'. On the left sidebar, the 'Groups' menu item is highlighted with a red box. The main content area shows the service status for 'all organisational units' with the following options:

- ON for everyone
- OFF for everyone

Below the radio buttons, there is an information icon and the text: 'Most changes take effect within a few minutes. [Learn more](#)'. At the bottom right of the main content area, there are 'CANCEL' and 'SAVE' buttons.

3. To provide access to members of specific Organisational units:

- a. Select **Organisational units** on the left hand menu.
- b. Select an organisational unit.
- c. Select **ON for everyone** in the main screen.

The screenshot shows the 'Service status' configuration page for a specific organisational unit, 'Thredd SSO'. On the left sidebar, the 'Organisational units' menu item is highlighted with a red box, and 'Thredd SSO' is selected. The main content area shows the service status for 'Thredd SSO' with the following options:

- ON
- OFF

Below the radio buttons, there is an information icon and the text: 'Override will override the settings inherited from the parent org unit. Most changes take effect within a few minutes. [Learn more](#)'. At the bottom right of the main content area, there are 'CANCEL' and 'OVERRIDE' buttons.



Configuring SSO with Okta (OIDC)

If your organisation uses Okta, you can configure Okta as an IdP provider to provide SSO access to various Thredd services. For example, you can use SSO to access Thredd services such as Thredd Portal. This guidance describes the steps for using the OpenID Connect (OIDC) protocol for setting up SSO.

Note: Setting up SSO is not mandatory, but Thredd recommends it.

Overview

The steps involve:

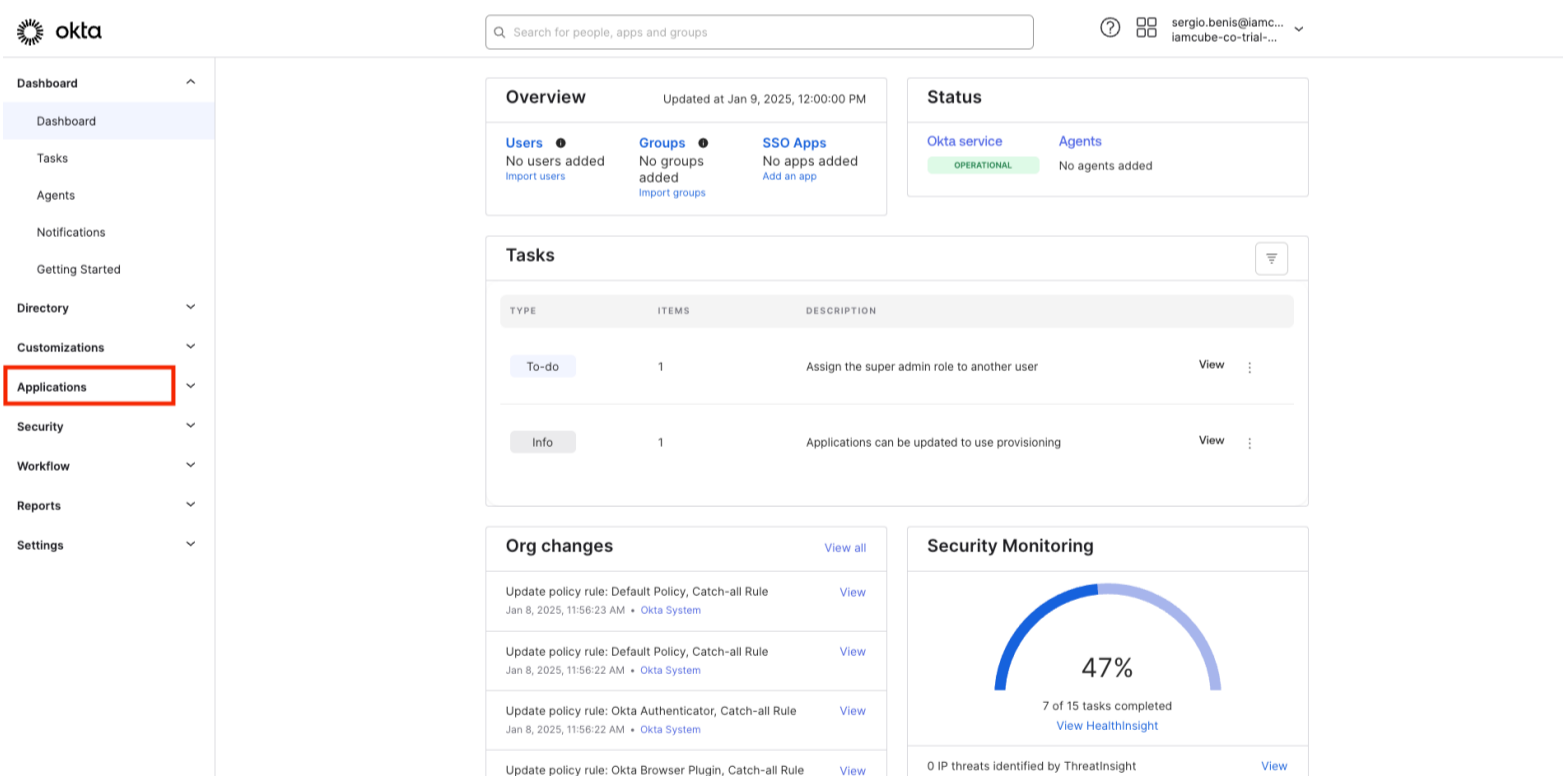
- Creating an app and app integration.
- Setting URL and refresh token settings.
- Specifying your access control requirement.
- Sharing authentication details with Thredd, via the Client ID/Client Secret method or the [private_key_jwt](#) authentication method.

Note: Thredd recommends using the [private_key_jwt](#) authentication method.

Thredd will provide you with a Sign-in Redirect URI for creating a web application integration.

Configuring SSO

1. Log in to the Okta Administration console.
2. Select **Applications** from the left-hand menu.



3. Select **Create Application**.
4. In **Create a new application integration**, select **OIDC - OpenID Connect** and **Web Application**.



X

Create a new app integration

Sign-in method

[Learn More](#)

- OIDC - OpenID Connect**
Token-based OAuth 2.0 authentication for Single Sign-On (SSO) through API endpoints. Recommended if you intend to build a custom app integration with the Okta Sign-In Widget.
- SAML 2.0**
XML-based open standard for SSO. Use if the Identity Provider for your application only supports SAML.
- SWA - Secure Web Authentication**
Okta-specific SSO method. Use if your application doesn't support OIDC or SAML.
- API Services**
Interact with Okta APIs using the scoped OAuth 2.0 access tokens for machine-to-machine authentication.

Application type

What kind of application are you trying to integrate with Okta?

Specifying an application type customizes your experience and provides the best configuration, SDK, and sample recommendations.

- Web Application**
Server-side applications where authentication and tokens are handled on the server (for example, Go, Java, ASP.Net, Node.js, PHP)
- Single-Page Application**
Single-page web applications that run in the browser where the client receives tokens (for example, Javascript, Angular, React, Vue)
- Native Application**
Desktop or mobile applications that run natively on a device and redirect users to a non-HTTP callback (for example, iOS, Android, React Native)

Cancel

Next

5. Select **Next**.
6. Enter a name for your application integration in **Application integration name**.
7. Select the **Refresh Token** check box.
8. Enter the URL value of the sign-in redirect URIs in the **Sign-in redirect URIs** section.

The screenshot shows the Okta admin console interface for creating a new web application integration. The page title is "New Web App Integration". Under "General Settings", the "App integration name" is "Thredd Integration". The "Proof of possession" section has "Require Demonstrating Proof of Possession (DPoP) header in token requests" unchecked. The "Grant type" section has "Client acting on behalf of itself" selected, with "Authorization Code" and "Refresh Token" checked under "Core grants". The "Sign-in redirect URIs" section is highlighted with a red box, showing a text input field with the value "https://thredd-will-provide-this-value" and an "Add URI" button. The "Allow wildcard * in sign-in URI redirect" checkbox is unchecked.



9. Select how you want to control access to Thredd applications from your organisation. The following example shows that access is available to all users in your organisation, and where there is immediate access.

The screenshot shows the Okta Admin Console interface. On the left is a navigation menu with items like Dashboard, Directory, Customizations, Applications, Security, Workflow, Reports, and Settings. The main content area is titled 'Assignments' and contains two sections: 'Controlled access' and 'Enable immediate access (Recommended)'. The 'Enable immediate access' section has a radio button selected, and a tooltip is displayed next to it. The tooltip text reads: 'To ensure optimal app performance at scale, Okta End User Dashboard and provisioning features are disabled. Learn more about Federation Broker Mode.' Below the tooltip are 'Save' and 'Cancel' buttons. A red box highlights the 'Enable immediate access' section and the tooltip.

10. Share details with Thredd.

- If you are using a Client ID/Client secret, share this detail with Thredd using your preferred secure method.
- If you prefer Thredd's recommended authentication method of `private_key_jwt`, complete the following steps to share the private key.

Share the Private Key JWT

1. Select the application that you have just created under **General**.
2. Select **Edit** next to **Client Credentials**.



Client Credentials Edit

Client ID Copy
Public identifier for the client that is required for all OAuth flows.

Client authentication Client secret Public key / Private key

Proof Key for Code Exchange (PKCE) Require PKCE as additional verification

CLIENT SECRETS

Generate new secret

Creation date	Secret	Status
Jan 9, 2025 Copy	Active ▼

3. Select **Public Key / Private Key**, the page updates to show the public key configuration options:



Client authentication Client secret Public key / Private key

Proof Key for Code Exchange (PKCE) Require PKCE as additional verification

PUBLIC KEYS

Configuration Save keys in Okta Use a URL to fetch keys dynamically

KID	Status	Created
No public keys are configured. Click Add key to get started.		

[Add key](#)

[Save](#) [Cancel](#)

4. Select **Add key**. The **Add a public key** screen appears.
5. Select **Generate new key**.
6. When the private key is shown, select PEM.
7. Copy the value to your clipboard and save it.

You must share the private key with Thredd. The following shows the on-screen instructions for generating and copying a private key.



Add a public key

Paste your own public key or automatically generate a new key pair.

```
Clear Generate new key  
{  
  "kty": "RSA",  
  "e": "AQAB",  
  "kid": "1j9wEsKGprj20d5e4yIuEoeZ1fczIcAHcWwzzmDrzHo",  
  "n": "1qUoFm8h1-jtvGnofsBGM7X_GinhSri0WeRgd5aLzqU3kThUNFOJfMGe0j1vh74Rte2stN0Vn_GqIAVHRGj_1uX4"  
}
```

Private key - Copy this!

The private key appears only once for enhanced security. Copy this key and store it somewhere safe for use later.

```
JSON PEM Copy to clipboard  
-----BEGIN PRIVATE KEY-----  
MIIEvgIBADANBgkqhkiG9w0BAQEFAAQCAQAwgAgwAgEAAoIBAQDWPsgWbyHX6O28  
aeh+wEYztf8aKeFKuI5Z5GB31ovOpTeROFQ0U418wZ46OW+HvhFN7ay03RWf8aog  
BUdEaP/W5fhiyCZXRWJHfHnJPaXEWNRbxaCoNIYyZEoSebbq1Kf921NVZNGQdEq3  
y3MU1rpgne+Po0z0qD0bqLgFUIv7/f0ZRc4SBtv9h1gxec9pphqAFvaua5tjF8fN  
Qx9t06Sfm9B6NYtPR+0aZgwZH6W1Mwdr5RJCP12Hw/rEo/N64nuRmQ8Cha5gMO8  
uJh68XtXRxfqT1Yvx38pKJxnsPVuHOv5SX/jt1/kblx2oZpj20vGKLDmjJUtdV/  
OJ0QV7abAgMBAECggEAMPwzGF+XZSti6g9vgFHIE7ASvn1VUZSp5A,jzHQet82SQ  
OGOXD/QKmf6j6hzGf7+Y0mU194bHEX/3V+RsfcfKr1P/ai fMXD1Y8wCM2Kjph1RR  
bno9LnYCNEjgALRnUsTHS+98Zq4iB2oKzUQwiM5ybW9Nx0WY3/LvMzs/d/MIZ2MG  
F147h6t/Nqf9VG7FWW0I6FmJpmv1iCjMPRm8hySbDhnOSLu0d7Jn0HUtxkvE fouW  
K/RyceNfEPKpYw+B5c4hG1JDtXz4Lor4tjWGfdnRTfwHjEbEfwYK+x063ioj6IP1  
-----END PRIVATE KEY-----  
Done Cancel
```

8. Select Done.



General FAQs

The following detailed list of FAQs focuses on the Thredd Secure Framework and its components, and related topics. These components include Thredd Portal and Thredd CA.

Overview

What is the Secure Framework?

The Thredd Secure Framework is a combination of components that enable secure access to Thredd's resources using a common identity store. These include Thredd Portal, Thredd CA, and mTLS termination (for those clients using mTLS). It ensures secure communication and access control through features like mTLS, OAuth, and certificate-based authentication. For more information, see [Thredd Secure Framework](#).

What is mTLS and how is it used in the Secure Framework?

mTLS (Mutual Transport Layer Security) is a security protocol used to establish trust between clients and servers.

- Transport Certificates issued by Thredd CA are used to establish secure connections.
- mTLS Termination requires on-premise infrastructure to establish Trust Chains to ensure certificates originate from legitimate sources.
- It is used in the External Host Interface (EHI) setup, and works in the background for the SOAP and REST API setups.

For more information about Mutual TLS, see [Mutual TLS explained](#).

What is the role of Cloudfity in the Secure Framework?

Thredd previously used Cloudfity as a Software as a Service (SaaS) capability to act as an Identity Provider (IdP), OAuth OpenID Provider (OP), and Policy Decision Point (PDP). However, Thredd has enhanced its Secure Framework to allow clients to manage access to Thredd services via Thredd Portal. As a result, Thredd has phased out use of Cloudfity within the Thredd Secure Framework. For more information, see [Thredd Secure Framework](#).

Can I still use or request a VPN setup?

No, Thredd does not support new VPN setups. You must connect securely to Thredd using the Secure Framework.

Certificates

What is Thredd CA, and what certificates does it provide?

Thredd CA is Thredd's Certificate Authority responsible for creating and managing certificates. Users with an Admin role can request and revoke certificates from within Thredd Portal. It provides:

- **Transport Certificates:** For secure connections between resources.
- **Signing Certificates:** For private_key_jwt authentication, signed messages, and non-repudiation.
- **Root and Issuing Certificates:** Bundled in the EHI Trust Chain, for verifying Transport Certificates in EHI setups that use mTLS.

What certificates are required for different Thredd applications?

The certificates required for various Thredd applications are as follows:

- **REST API:** Requires Signing Certificates for TLS and mTLS connections; mTLS also requires Transport Certificates.
- **SOAP API:** Requires only Transport Certificates.
- **External Host Interface (EHI):** Requires Root and Issuing Certificates (available in Thredd Trust Chain).
- **Thredd Portal:** Pre-installed Transport Certificates.
- **Smart Client:** Bundled Transport Certificates in the installer.

For more information, see the following sections that are related to requesting certificates in the Connecting to Thredd Guide:

- [Creating an Application and OAuth Client](#)
- [Requesting certificates for applications using TLS](#)
- [Requesting certificates for applications using mTLS](#)
- [Setting up EHI for mTLS connections](#)



Single Sign-On

How does SSO work in the Secure Framework?

SSO capabilities are provided through Thredd Portal, allowing federated authentication. This enables users to authenticate once and gain access to multiple Thredd resources instead of logging in separately for each service. You can also configure your own IdP provider for SSO.

See the following section within the Connecting to Thredd Guide: [Setting up Single Sign-On access for users](#)

External Host Interface

How do I set up mTLS for EHI?

Step 1: Set up certificates.

You need a Server Certificate and the EHI Trust Chain to enable mTLS.

1. Set up Server Certificates:
 - a. Obtain a Server Certificate from a trusted Certificate Authority (CA) vendor, such as Verizon, Digicert, or Amazon Web Services.
 - b. Install the following certificates on your EHI listening endpoint:
 - Server or Leaf Certificate: Issued to individual servers by a CA. This certificate is the "leaf" of the hierarchical tree of trust.
 - Intermediate or Issuing Certificate: Represents the CA that issued the server certificate.
 - Root Certificate: The trusted root of the CA chain.
2. Set up Client certificates. Thredd presents its Transport Certificate during the connection handshake. Your system must validate this certificate against the CA Thredd Trust Chain (Root and Issuing Certificates).

Step 2: Test the mTLS Connection.

1. Test the connection to ensure the mTLS handshake is successful.
2. Verify that your system can receive and process EHI messages securely.

Step 3: Add Optional Security Enhancements

To further secure your mTLS setup, consider implementing the following:

1. IP Address Allowlist: Add Thredd's fixed IP address to your allow list to ensure only authorised traffic is accepted.
2. Certificate Pinning: Implement certificate pinning to block requests made with incorrect certificates.

For more information, see [Setting up EHI for mTLS connections](#).

Postman Setup

How do I use Postman to test the Secure Framework?

Prerequisites

- Install Postman on your system.
- Create an application in Thredd Portal.

Main steps

1. Ensure that you have the necessary certificates generated and signed by the Thredd Certificate Authority (CA).
2. Add variables from Thredd CA to the variables section in the Postman collection. You can obtain this information from your application's details in Thredd Portal.
3. If using mTLS, add certificates for mTLS and repeat this process for all required hosts.
4. Set the variables in the **Variables** tab.
5. Check your settings and finish setting up.
6. Test making an API call using the Postman Collection.

For a full step-by-step guidance, see one of the following:

- [Using the Postman Collection to call REST APIs over TLS](#)
- [Using the Postman Collection to call REST APIs over mTLS](#)



More information

Where can I find more information?

For more details, refer to the following resources:

- [Setup preparation and steps](#) – for connecting to Thredd API Hub (REST APIs) and Thredd Portal via TLS or mTLS connections
- [API Hub website](#) – API Reference and guides for Thredd's REST APIs
- [Thredd Portal: Card and Transaction Management Guide](#)
- [External Interface \(EHI\) Guide \(JSON\)](#)
- [External Interface \(EHI\) Guide \(XML\)](#)



Glossary

This page provides a list of glossary terms used in this guide.

C

Card Scheme (Network)

Card network, such as Discover, MasterCard, or Visa, responsible for managing transactions over the network and for arbitration of any disputes.

Certificate Authority

A Certificate Authority is an entity that validates the identities of entities (such as individuals, organizations, or websites) and binds them to cryptographic key pairs through the issuance of digital certificates.

Confidential Client

A client that can maintain the confidentiality of their credentials (e.g., client implemented on a secure server with restricted access to the client credentials), or capable of secure client authentication using other means. For more details, refer to RFC 6749 for the OAuth 2.0 Authorization Framework.

E

External Host Interface (EHI)

The external system to which Thredd sends real-time transaction-related data. The Program Manager uses their external host system to hold details of the balance on the cards in their programme and perform transaction-related services, such as payment authorisation, transaction matching and reconciliation.

I

Identity Provider (IDP)

An IDP is a system entity that creates, maintains, and manages identity information for principals and also provides authentication services for relying on applications within a federation or distributed network.

M

Mutual Transport Layer Security (mTLS)

mTLS is a method for mutual authentication that ensures the parties at each end of a network connection are who they claim to be by verifying that they both have the correct private key. The information within their respective TLS certificates provides additional verification.

O

OAuth OpenID Provider

An OAuth OpenID Provider (OP) is an entity that has implemented the OpenID Connect and OAuth 2.0 protocols. OPs can also be referred to by the role it plays, such as: a security token service, an identity provider (IDP), or an authorization server.

P

Program Manager

A customer who manages a card program. The program manager can create branded cards, load funds and provide other card or banking services to their end customers.

S

Scope

The permissions related to the resources your application is registered to access. Each scope relates to specific endpoints. For example, the cards.encrypted scope gives permission to use the Get Encrypted Data endpoint.

Secure Connectivity Framework

The Secure Connectivity Framework is an umbrella set of rules and standards for identity management, verification, and assurance within a sector. The framework establishes common principles, definitions, and Open Standards for data sharing to create the foundations of a trusted data-sharing ecosystem



Smart Client

Smart Client is a user interface for programme managers to manage their account on the Thredd Platform. Smart Client is installed as a desktop application.

T

Thredd CA

Thredd CA (Certificate Authority) acts as the Certificate Authority for issuing certificates. You can create applications for your organisation, as well as request Transport and Signing certificates.

Transport Certificate

A Transport Certificate (or TLS Certificate) is a data file that contains important information for verifying a server's or device's identity, including the public key, a statement of who issued the certificate (TLS certificates are issued by a Certificate Authority), and the certificate's expiration date.



Document History

This section provides details of what has changed since the previous document release.

Version	Date	Reason	Revised by
Version 1.1	08/05/2026	Removing support for Raidiam Connect (for mTLS connections) and Cloudfity.	GF
Version 1.1	06/02/2026	Added new Scopes appendix, which describes each of the different scopes returned from the DCR Endpoint.	JB
	23/01/2026	Added section on how to connect to Thredd using AWS and VPN. See Connecting with AWS and VPN .	JB
	23/01/2026	Added a prerequisites list in the Generating and Obtaining a Software Statement Assertion (SSA) section and linked it with related information elsewhere in the guide.	ER
	14/08/2025	Added section on how to manually create users, including how to assign restricted codes and Thredd Portal roles. See Manually Create Users .	JB
	05/06/2025	New FAQ section. See FAQs .	KD
Version 1.0	25/03/2025	First version.	WS



Contact Us

Please contact us if you have queries relating to this document. Our contact details are provided below.

Thredd UK Ltd.

Company registration number 09926803

Support Email: occ@thredd.com

Telephone: +44 (0) 203 740 9682

Our Head Office

33 Kingsway

London

WC2B 6UF

UK

Technical Publications

If you want to contact our technical publications team directly, for queries or feedback related to this guide, you can email us at:

docs@thredd.com.