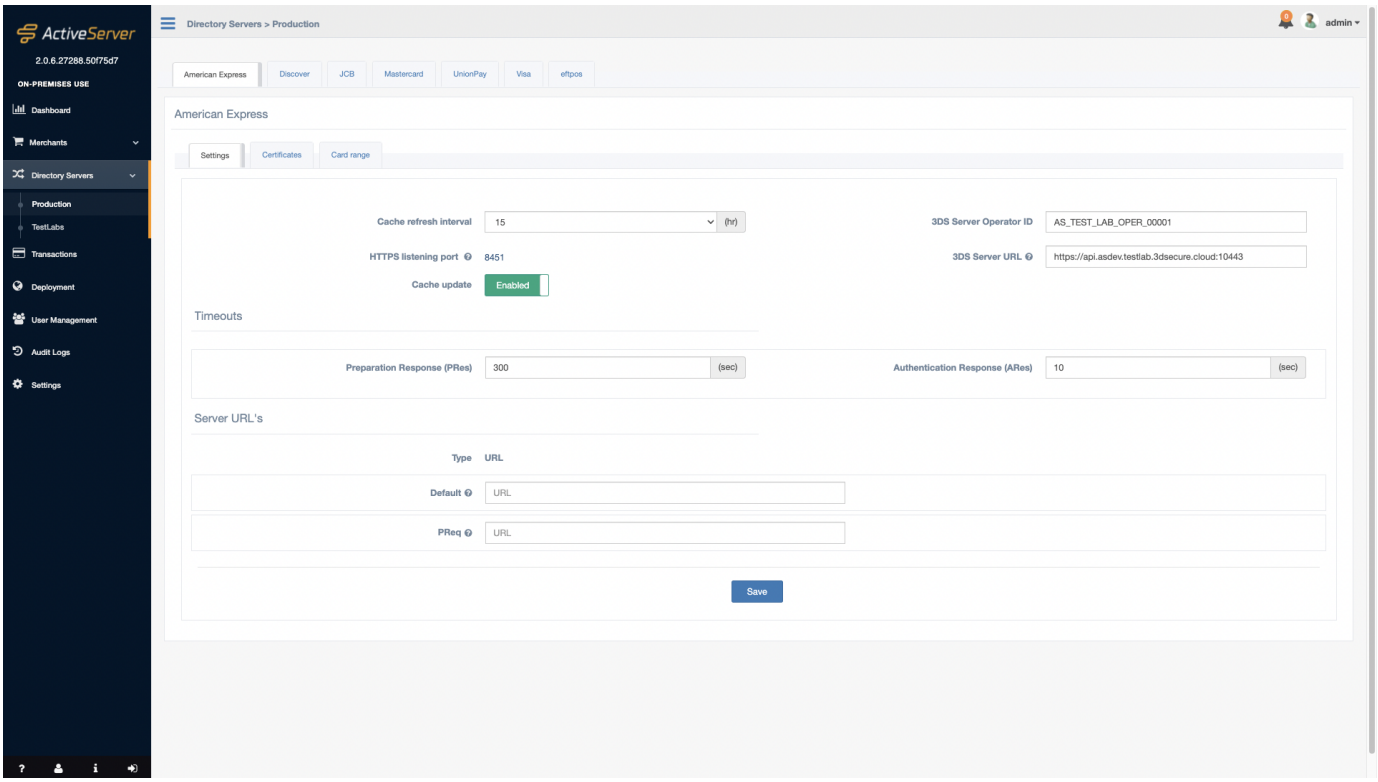
Manage DS certificates

All the certificates for card schemes supported by ActiveServer can be managed from the **Directory Servers** page, under the **Certificates** tab.

User Access

A user requires the **System admin** role to manage DS certificates.



To manage a card scheme's certificates:

Select the appropriate **card scheme** tab at the top of the page to display its details.

Client and server certificates

In a 3DS2 authentication, both inbound and outbound traffic with the DS is required to be mutually authenticated, meaning **ActiveServer** will act as both a HTTPS **client** and a **server** in different processes.

ActiveServer acts as a client when it connects to the card scheme's DS to send the initial AReq. The **client certificate** is used to verify and authenticate **ActiveServer** in this flow, with this connection also being used to receive the resulting ARes from the DS when it is ready.

If an authentication requires a challenge, **ActiveServer** will act as a server when it is time for the DS to notify us of the authentication results in the RReq and respond with the RRes.

This certificate/s is downloaded from the card scheme, usually after providing a Certificate Signing Request (CSR).

Each certificate has the following table information displayed:

- **Certificate information** - certificate details of the installed certificate.
- **Status** - displays whether the certificate has been signed by a trusted CA. **Valid** indicates the CA Certificate has been successfully installed in **ActiveServer**, whereas **Not Valid** indicates no associated CA Certificate can be found.
- **Validity** - start and end dates for certificate validity.
- **Issuer** - name of the CA issuer that signed the certificate.
- **Hash algorithm** - hash algorithm used for the certificate signature.
- **Export | Delete** - Export or delete a certificate.

The following processes can be carried out for client certificate management:

Create CSR

To assist with the generation of a CSR, **ActiveServer** provides this functionality via the ***Create CSR*** button. However, it also possible to do this process manually if you prefer, using an external method such as ***OpenSSL***.

The certificate content should be filled in as appropriate for the card schemes requirements, with the following options available:

- **Key size** - key size of the request, measured in bits .
- **Common Name** – hostname that will use the certificate, usually a fully-qualified domain name. For the server certificate, this must be the hostname of **ActiveServer**. For the client certificate, this will usually be the same as the sever certificate, however some card schemes may have different requirements. This value will be pre-filled with the **3DS Server URL** setting of the DS if it available.

- **Organization** – legal name of your company or organization.
- **Organization Unit** – departmental or division name for your group.
- **City** – city where your company is located.
- **Province** – province or state where your company is located.
- **Two letter country code** – two-character abbreviation for your country.
- **Hash algorithm** - hash algorithm used to sign the CSR.

Creating a CSR will generate the CSR content and an associated private key to be stored in the database.

> **Important**
>
> Only one CSR for the client certificate and one CSR for server certificates can be stored at a time, per card scheme.

Download CSR

Download CSR will export the CSR content to a `.csr` file in the following file name pattern: *"Common Name"_"Card Scheme".csr*, e.g. `api.testlab.3dsecure.cloud_JCB.csr`.

Download CSR will only be available if a CSR has already been created in the system.

Delete CSR

Delete CSR will delete both the CSR content and associated private key from the system.

Delete CSR will only be available if a CSR has already been created in the system.

> **Warning**
>
> Deleting a CSR is permanent and will invalidate any outstanding CSRs waiting to be signed, meaning they will be unable to be installed.

Install Certificate

Install Certificate will install a signed certificate, either using the raw **certificate content** or **certificate file**.

The certificate can be in one of the following formats: `.pfx`, `.p7b`, `.p12`, `.jks`, `.pem`. **ActiveServer** will try each file type one by one when installing. If the file type requires a password, such as `.p12`, enter the password on the install certificate page.

ActiveServer will attempt to use a private key if it is included with a `.pfx`, `.p12` or `.jks` file, otherwise it will use an existing private key created by the system if it is available. See below for information on creating your own private key.

If the card scheme provider only requires one certificate for client and server connections, the **Server certificate is the same as the client certificate** option can be selected. This will install the certificate to both the client and server sections.

Install certificate will save the certificate to the database, then remove any outstanding CSR content from the system, as well as the local private key if an external private key was provided. After this a new CSR and private key are able to be created if required.

Important: Restart required for new certificates

To use newly installed certificates, the **ActiveSever** instance **must be restarted** to refresh the DS connectors.

Warning

It is only possible to have **one** client and **one** server certificate at a time, **installing** another certificate will cause the current certificate and private key to be overwritten.

Potential WAF issue when importing certificate

Sometimes when importing a `.p7b` certificate from Mastercard, a **403 error** may be thrown by your WAF due to the binary content of the .p7b file containing unsafe content. A workaround is to convert the file to a PEM encoded format. This can be done using an open source tool such as **openssl** with the following command:

```
`openssl pkcs7 -print_certs -in input.p7b -out output.cer -inform der`"
```

Export and delete

Client and server certificates are able to be exported for backup purposes or deleted if required by selecting the relevant icon from the certificate table.

Certificates can be **Exported** in two formats:

- **PKCS12 keystore (.p12 including private key)** - creates a `.p12` file including the certificate and associated private key. Optionally a password can be included for the file.
- **Certificate only (.pem)** - creates a `.pem` file only containing the certificate.

Delete will show a prompt, asking the user to confirm deletion of the certificate, before removing it from the system.

Warning

Deleting the certificate is permanent, with exporting the certificate as a backup being recommended first.

CA Certificate

CA Certificates are used to verify the CA signer of the server/client certificate for this provider and ensure they are from a valid CA. CA certificates will be automatically added if found in the certificate chain during server/client certificate upload, otherwise they should be added manually to validate installed certificates.

If an associated CA has not been installed, the **Certificate Status** of the client or server certificate will be **Not valid**. Deleting a CA will also invalidate previously installed certificates.

The ***Install Certificate*** button will show a prompt to search for a local certificate file. Certificate information and functionality shown is the same as described in the Client and Server certificate section, with the addition of the CA Certificate **Alias** value.

Using external tool for certificate management

You can use your chosen tool for generating a CSR and private key. An example using **OpenSSL** is provided below.

Ensure **OpenSSL** is installed and open a terminal to perform the following:

1. Create a RSA Private Key

```
openssl genrsa -out privateKey.key 2048
```

2. Generate the CSR and follow the prompts to enter CSR details

```
openssl req -new -key privateKey.key -out yourCSR.csr
```

3. Once the CSR has been signed by the card scheme, combine the provided signed certificate and card scheme CA certificate chain with the generated private key

```
openssl pkcs12 -export -out certifcate.p12 -inkey privateKey.key -in
signedcertificate.crt -certfile ca-chain.crt
```

4. Install the certificate as normal