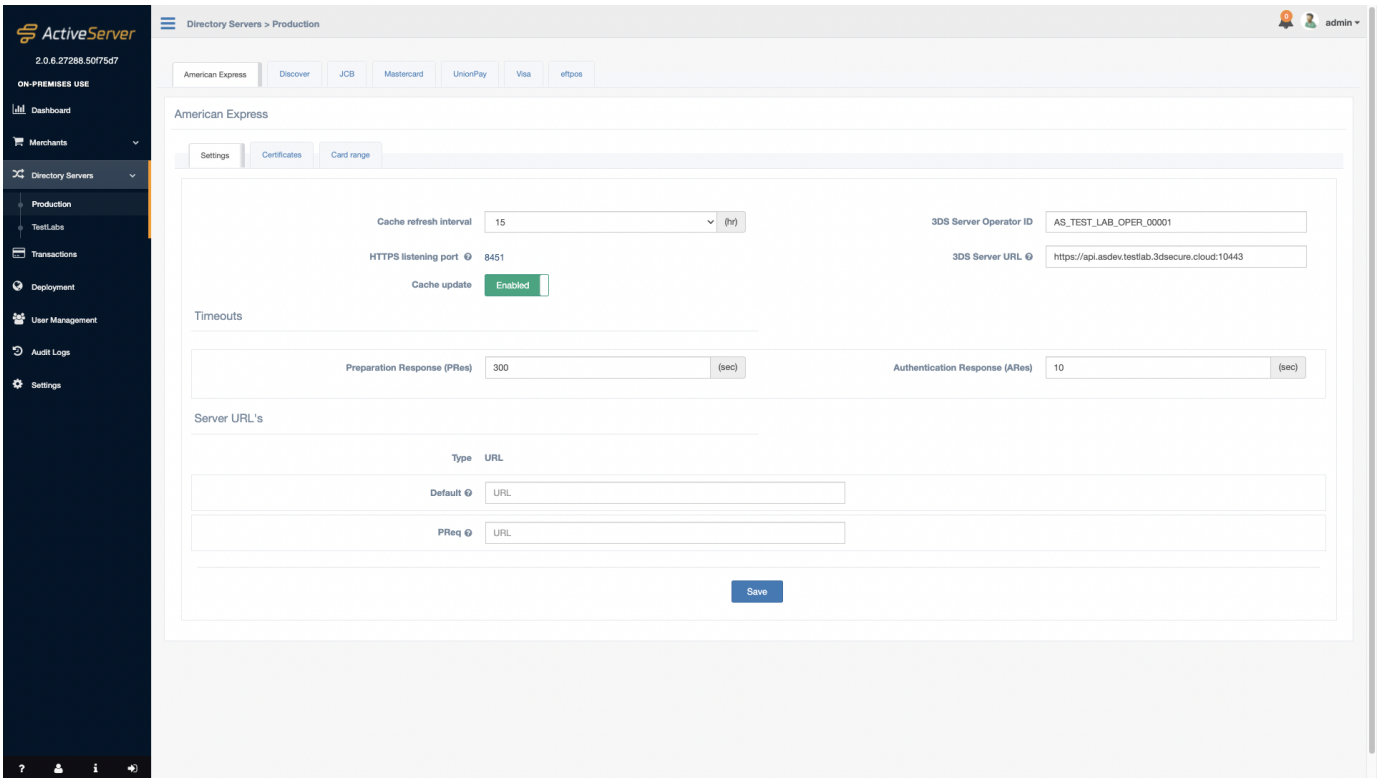
DS証明書の管理

ActiveServerでサポートされるすべての国際ブランドは、**Certificates**タブの**Directory Servers**ページで管理できます。

> **ユーザーアクセス**
>
> ユーザーがDS証明書を管理するには、**System admin**ロールが必要です。



国際ブランド証明書を管理するには：

詳細を表示するには、ページ上部の適切な**国際ブランド**タブを選択します。

クライアントとサーバー証明書

3DS2.0の認証ではDSへのインバウンドとアウトバンドの接続は相互SSL認証される必要があり、**ActiveServer**はそれぞれHTTPS**サーバー**と**クライアント**として役割を担います。

ActiveServerから国際ブランドのDSに接続する際にはActiveServerはクライアントとしての役割を担います。AReqを送信する際に**クライアント証明書**を使用しAResを受信します。

もし認証においてチャレンジが必要な場合は**ActiveServer**はサーバーとしての役割を担います。チャレンジ終了後にDSからRReqを受信し最終的な認証結果をActiveServerに通知されます。この接続を相互認証するためにサーバー証明書が使用されます。

証明書は、通常、証明書署名リクエスト（CSR）を提供した後に国際ブランドからダウンロードできます。

証明書の表には以下の情報が記載されています:

- **Certificate Information** - インストールされている証明書の情報
- **Status** - CAによってサインされたか否か。インストールされているCA証明書によってサインされている場合は**Valid**、されていない場合は**Invalid**を表示します。
- **Validity** - 証明書の有効期限
- **Issuer** - 証明書をサインしたCAのイシュアー名
- **Hash algorithm** - 証明書をサインする際に使用されたハッシングアルゴリズム
- **Export | Delete** - 証明書のダウンロードと削除

以下のクライアント証明書の管理ができます：

CSRの作成

CSRの生成を支援するため、**ActiveServer**は***Create CSR***ボタンからこの機能を提供しています。ただし、ご希望であれば、Java keytoolのような別の方法を使用して、手動でこのプロセスを実行することもできます。

証明書の内容は、国際ブランドの要件に応じて入力する必要があります。以下のオプションが利用可能です。

- **Key size** - リクエストのキーのサイズ（ビット単位）
- **Common Name** – 証明書に使用されるホスト名。通常、完全修飾ドメイン名が使用されます。サーバー証明書の場合はこれは**ActiveServer**のホスト名になります。クライアント証明書の場合は通常サーバー証明書と同じになりますが、国際ブランドによっては違う場合もあります。**Common Name**の値はデフォルトでDSに設定されている**3DS Server URL**のドメイン名になります。
- **Organization** – 企業または組織の法的な名前

- **Organization Unit** – グループの部署または部門の名前
- **City** – 企業がある市区町村
- **Province** – 企業がある都道府県
- **Two letter country code** – 国の2文字の略称
- **Hash algorithm** - CSRの署名に使用されるハッシュアルゴリズム

CSRの作成では、生の証明書コンテンツが作成され、 `.p10` 形式で***Download certificate***のボタンが提供されます。

> **重要**
>
> 各DSにはクライアントCSRとサーバーCSRの２つのみ保存できます。

CSRをエクスポート

CSRをエクスポートはCSRはCSRのコンテンツを `.csr` としてダウンロードします。ファイル名は*"Common Name"_"国際ブランド名".csr*のフォーマットになります。例： `api.testlab.3dsecure.cloud_JCB.csr` .

CSRを作成した後のみ***CSRをエクスポート***できます。

CSRを削除

CSRを削除した場合CSRのコンテンツとCSRを作成するのに使用された秘密鍵の両方を削除します。

CSRを作成した後のみ***CSRを削除***できます。

> **警告**
>
> CSRを削除した場合削除した後に署名された証明書はインスールできなくなります。

証明書をインストール

証明書をインストールは署名された**certificate content**または**certificate file**をインストールできます。

サポートされている証明書のフォーマット：`.pfx`，`.p7b`，`.p12`，`.jks`，`.pem`。**ActiveServer**は各ファイルタイプを読み込みます。もし、ファイルにパスワードが必要な場合はCertificateのページでパスワードを入力して下さい。例えば：`.p12`はパスワードが必要なファイル形式ですので、インストールする際にパスワードを入力する必要があります。

ActiveServerは`.pfx`、`.p12`または`.jks`のファイルがインストールされた場合はファイルに含まれている秘密鍵を使用して証明書をインストールしようとします。もし、ファイルに秘密鍵が含まれていない場合は現在インストールされている秘密鍵を使用します。秘密鍵の作成の仕方についてはこちらを参照下さい。

もし、国際ブランドがクライアントとサーバーの接続に必要な証明書が1つだけの場合、**Server certificate is the same as the client certificate**オプションを選択できます。これにより、クライアントセクションとサーバーセクションの両方に証明書がインストールされます。

インストール (Install)

署名済みの証明書は、***Install***ボタンを使用することでインストールできます。

> **Warning**
>
> 一度に**1つ**のクライアント証明書のみを持つことができ、別の証明書を**インストール**または**インポート**すると、現在の証明書が上書きされます。

> **WAFを利用している場合、インストール時に問題が発生する可能性があります**
>
> Mastercardから`.p7b`証明書をインポートする際に、安全でないコンテンツを含む.p7bファイルのバイナリコンテンツが原因で、**403エラー**がWAFによって返却される場合があります。回避策は、ファイルをPEMエンコード形式に変換することです。次のコマンドを実行することで変更できます。(オープンソースのopensslを利用)
>
> ```
> openssl pkcs7 -print_certs -in input.p7b -out output.cer -inform der
> ```

エクスポートと削除 (Export & Delete)

クライアントとサーバーの証明書は、バックアップのためにエクスポートしたり、必要に応じて証明書テーブルから削除アイコンを選択して削除したりできます。

証明書は、次の2つの形式で**エクスポート**できます。

- **PKCS12キーストア（秘密鍵を含む.p12)** - 証明書と関連する秘密鍵を含む `.p12` ファイルを作成します。オプションで、ファイルのパスワードを含めることができます。
- **証明書のみ（.pem)** - 証明書のみを含む `.pem` ファイルを作成します。

Deleteは、システムから証明書を削除する前に、証明書の削除を確認するようユーザーに求めるプロンプトを表示します。

> ⚠ **警告**
>
> 証明書の削除は永続的であり、最初にバックアップとして証明書をエクスポートすることをお勧めします。

CA証明書

CA証明書は、サーバー/クライアント証明書のCA署名者を検証し、それらが有効なCAからのものであることを確認するために使用されます。 CA証明書はサーバー/クライアント証明書をインストールした際ににCAチェーンが見つかった際に自動的にインストールされます。または、手動でインストールする事も可能です。

関連するCAがインストールされていない場合、クライアントまたはサーバー証明書の**Status**は**Not Valid**です。 CAを削除すると、以前のインストールも無効になります。

[***証明書のインストール***

]ボタンには、ローカル証明書ファイルを検索するプロンプトが表示されます。 表示される証明書情報と機能は、クライアントおよびサーバー証明書にCA証明書の**エイリアス**値を追加したものになります。

証明書管理に外部ツールを使用する (Using External Tool)¶

選択したツールを使用して、CSRと秘密キーを生成できます。 **OpenSSL**を使用した例を以下に示します。

OpenSSLがインストールされていることを確認し、ターミナルを開いて以下を実行します。

1. RSA秘密鍵を作成します

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

2. CSRを生成し、プロンプトに従ってCSRの詳細を入力します

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

3. CSRが国際ブランドによって署名されたら、提供された署名済み証明書と国際ブランドCA証明書チェーンを生成された秘密キーと結合します

```
1 openssl pkcs12 -export -out certifcate.p12 -inkey privateKey.key -in
```

4. 証明書をインストールする。

