

Salesforce における全従業員ユーザーに対する多要素認証（MFA）の適用について (2026/06/11)

【概要】

2026 年 6 月 22 日に Salesforce により順次全ユーザーの多要素認証（MFA）が必須化されます（<https://help.salesforce.com/s/articleView?id=005321561&type=1>）。MFA 必須化に伴い、アシロボのブラウザ操作コマンドによる Salesforce へのログイン手順が変わります。新しい手順では、Salesforce の API からログイン URL を発行し、その URL 経由でブラウザログインを行います。そのため、Salesforce の API を利用するようにシナリオを修正する必要があります。

本ドキュメントでは、Salesforce の API を利用するための OAuth 設定手順と、アシロボでの連携方法についてご案内いたします。

なお、Salesforce ログインの新しい手順を行うためには、「JWT エンコード」という新しい RPA コマンドが、アシロボのアップデートによって、2026 年 6 月 11 日に追加が予定されています。本手順では、追加予定の「JWT エンコード」コマンドを利用します。

【手順】

新手順は、Salesforce の API 利用準備を含め、大きく分けて 3 つのセクションがあります。

1. 証明書および秘密鍵の作成
2. Salesforce の OAuth 設定によって API 利用を許可
3. アシロボのシナリオ修正

1. 証明書および秘密鍵の作成

アシロボから Salesforce へログインするには、証明書（server.crt）と秘密鍵（client.pfx）という 2 つのファイルが必要になります。これらは、Salesforce の API 利用時に、API 利用者の身分証明のために利用します。下記の手順の通り、PowerShell を利用して作成します。

※ PowerShell は Windows で標準インストールされているソフトウェアです。

1.1 PowerShell を起動します

「Win」＋「R」で表示されるダイアログに「powershell」と入力して Enter を押下します。

1.2 証明書を作成します

Powershell に下記の文字列を入力して Enter を押下します。

下記は、自己署名証明書を作成し、ファイルとして書き出すコマンドです。

```
$cert = New-SelfSignedCertificate `
-Subject "CN=SalesforceJWT" `
-CertStoreLocation "Cert:\CurrentUser\My" `
-KeyExportPolicy Exportable `
-KeySpec Signature `
-KeyAlgorithm RSA `
-KeyLength 2048 `
-HashAlgorithm SHA256 `
-NotAfter (Get-Date).AddYears(100)
Export-Certificate `
-Cert $cert `
-FilePath ".\server.cer"
certutil `
-encode ".\server.cer" ".\server.crt"
```

項目	内容
New-SelfSignedCertificate	自己署名証明書を作成するコマンド。証明書は Windows の証明書ストアに保存されるため、ファイルとして保存するために「Export-Certificate」と「certutil」を利用する。証明書データは\$cert という参照 ID に一時的に保持する。
-Subject "CN=SalesforceJWT"	New-SelfSignedCertificate のオプション。 証明書の所有者名を「SalesforceJWT」に設定する。
-CertStoreLocation "Cert:\CurrentUser\My"	New-SelfSignedCertificate のオプション。 作成した証明書を、Windows の現在のユーザーの「個人」証明書ストアに保存する。
-KeyExportPolicy Exportable	New-SelfSignedCertificate のオプション。 秘密鍵をエクスポートできるように許可する。
-KeySpec Signature	New-SelfSignedCertificate のオプション。 鍵の用途を「デジタル署名用」に指定する。JWT 認証に必要な設定である。
-KeyAlgorithm RSA	New-SelfSignedCertificate のオプション。 暗号化アルゴリズムに「RSA」を使用する
-KeyLength 2048	New-SelfSignedCertificate のオプション。 鍵の長さを 2048 ビット（一般的な長さ）に設定する。

-HashAlgorithm SHA256	New-SelfSignedCertificate のオプション。 ハッシュアルゴリズムに「SHA-256」を指定する。
-NotAfter(Get-Date).AddYears(100)	New-SelfSignedCertificate のオプション。 証明書の有効期限を現在の日付から「100 年後」に設定する。有効期限は必要に応じて、設定してください。
Export-Certificate	証明書（公開鍵）をファイルとして書き出すコマンド。
-Cert \$cert	Export-Certificate のオプション。 変数 \$cert に格納した証明書を指定する
-FilePath ".\server.cer"	Export-Certificate のオプション。 現在操作中のフォルダに server.cer という名前で保存する。
certutil	ファイル形式を Salesforce が対応する crt 形式に変換するコマンド
-encode ".\server.cer" ".\server.crt"	certutil のオプション。 証明書ファイル（server.cer）から server.crt を作成

1.3 秘密鍵を作成します

秘密鍵パスワードは必要に応じて変更してください。

秘密鍵（client.pfx）はログインに利用するファイルです。漏洩しないようにご注意ください。

<pre>\$pwd = ConvertTo-SecureString ` -String "ここに秘密鍵のパスワードを記載してください" ` -Force ` -AsPlainText Export-PfxCertificate ` -Cert \$cert ` -FilePath ".\client.pfx" ` -Password \$pwd</pre>	
項目	内容
ConvertTo-SecureString	指定のパスワードを安全な文字列（暗号文字列）に変換するコマンド。暗号文字列は\$pwd という参照 ID に一時的に保持する。
-String "ここに秘密鍵のパスワードを記載してください"	ConvertTo-SecureString のオプション。 設定したいパスワードの平文を指定する。
-Force	ConvertTo-SecureString のオプション。
-AsPlainText	平文を暗号文字列に変換するときに指定する。
Export-PfxCertificate	秘密鍵を指定の証明書から作成するコマンド。 PFX（PKCS #12）形式のファイルとして出力する

-Cert \$cert	Export-PfxCertificate のオプション。 証明証データを指定する。
-FilePath ".\client.pfx"	Export-PfxCertificate のオプション。 秘密鍵を client.pfx というファイル名で保存する
-Password \$pwd	Export-PfxCertificate のオプション。 秘密鍵にパスワードを設定する

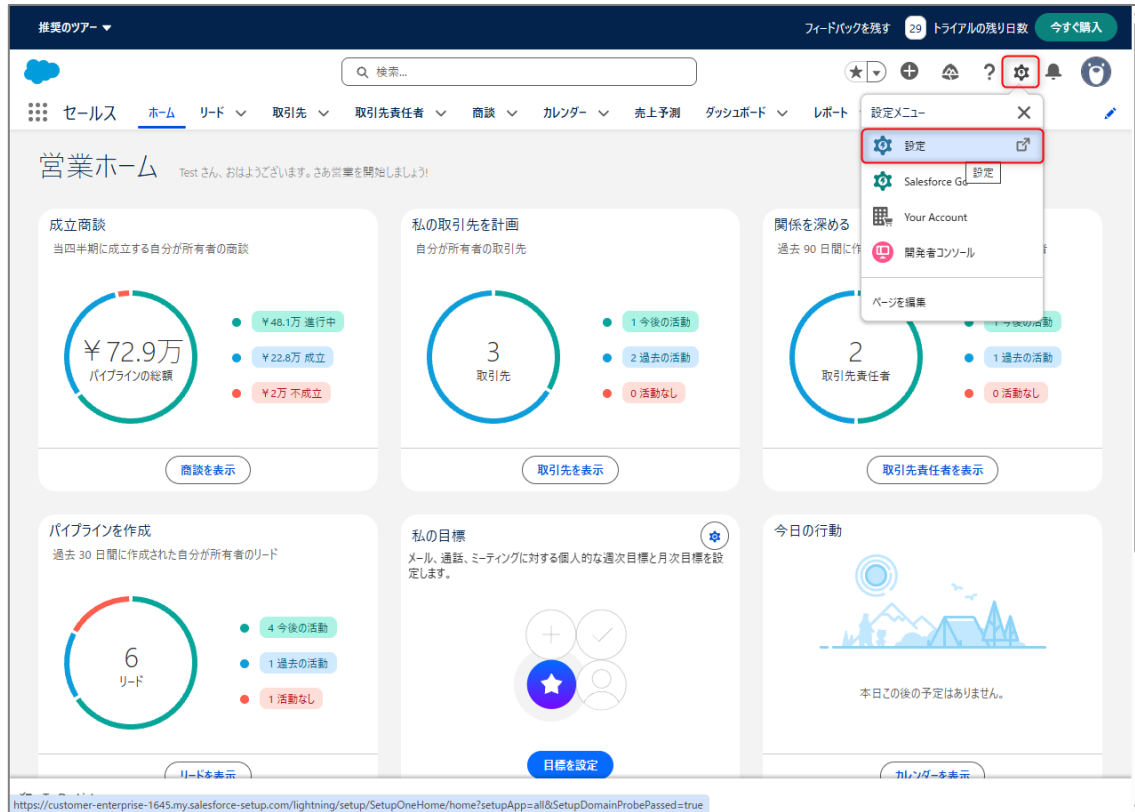
ここで作成した証明書 (server.crt) と秘密鍵 (client.pfx) は後述の手順で利用しますので、ファイルの場所をメモしてください。

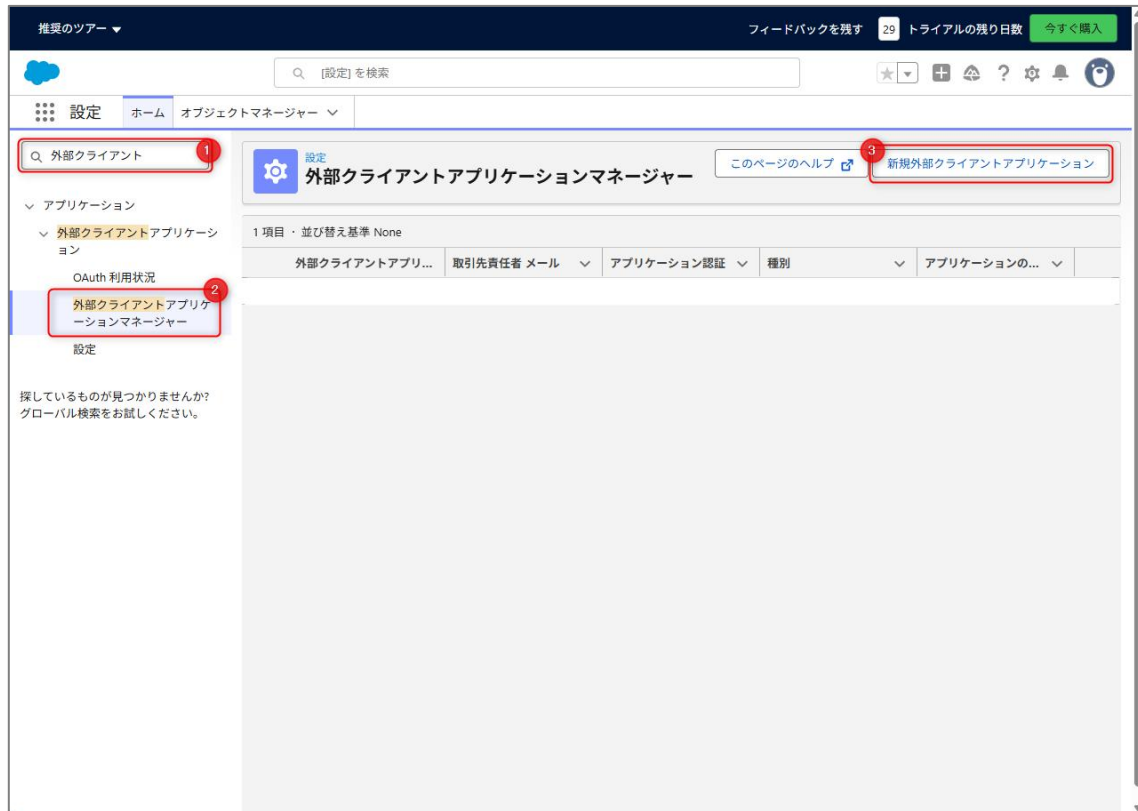
また、秘密鍵のパスワードはシナリオ内で設定する必要があるので、メモしてください。

2. Salesforce の OAuth 設定によって API 利用を許可

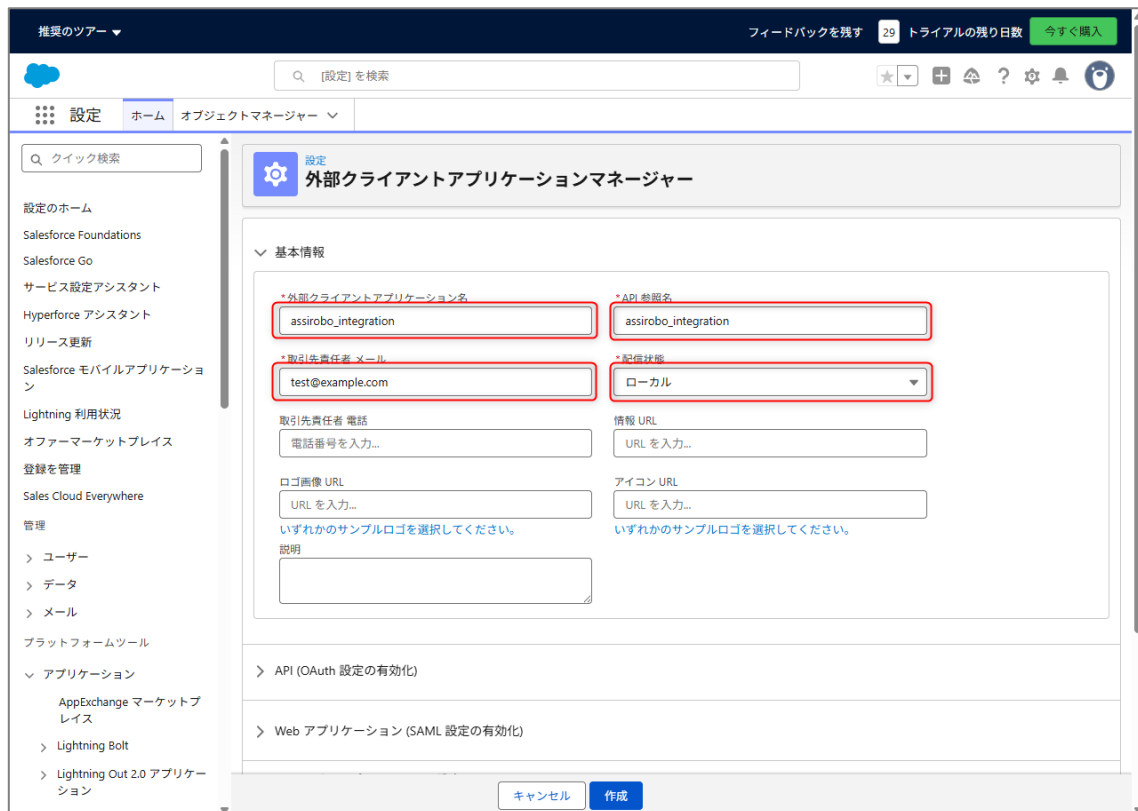
アシロボとの連携に必要な外部クライアントアプリケーションの作成を行います。

2.1 Salesforce の画面にて外部クライアントアプリケーションの作成画面に移動します





2.2 基本情報を入力します



項目	内容
外部クライアントアプリケーション名	外部クライアントアプリケーションマネージャーに表示する外部クライアントアプリケーション名
API 参照名	プログラムからアプリケーションを参照するときの名前
取引先責任者メール	Salesforce がアプリケーション提供者またはそのサポートチームへの連絡時に使用する連絡先メール
配布状態	ローカル組織の外部クライアントアプリケーションの場合は「ローカル」に設定。 配布用の外部クライアントアプリケーションの場合は「パッケージ化済み」に設定。 今回は、ローカル組織の外部クライアントアプリケーションであるため「ローカル」に設定

2.3 API（OAuth 設定）を有効化します

The screenshot shows the Salesforce OAuth Settings page. The left sidebar contains navigation links for various Salesforce features. The main content area is titled 'API (OAuth 設定の有効化)' and includes several sections:

- 1** A red box highlights the 'OAuth を有効化' checkbox, which is checked.
- 2** A red box highlights the 'アプリケーション設定' section, specifically the 'コールバック URL' field, which contains 'https://localhost'.
- 3** A red box highlights the 'OAuth 範囲' section, specifically the 'ID URL サービスにアクセス (id, profile, email, address, phone)' checkbox, which is checked.
- 4** A red box highlights the 'Web ブラウザーを使用してユーザーデータを管理 (web)' checkbox, which is checked.
- 5** A red box highlights the 'JWT ベアラーフローを有効化' checkbox, which is checked.

Below the 'OAuth 範囲' section, there are two red annotations:

- Web ブラウザーを使用してユーザーデータを管理 (web)
- いつでも要求を実行 (refresh_token, offline_access)

Below the 'JWT ベアラーフローを有効化' section, there is a red annotation:

- 証明書をアップロード

At the bottom of the page, there are two buttons: 'キャンセル' and '作成'.

項目	内容
コールバック URL	認証が成功した後にユーザーのブラウザがリダイレクトされる URL。アシロボでは https://localhost を設定
OAuth 範囲	アシロボから Salesforce へログインできるようにするには、下記を追加。 ● Web ブラウザーを使用してユーザデータを管理 (web) ● いつでも要求を実行 (refresh_token, offline_access)
フローの有効化	アシロボが対応する JWT ベアラーフローにチェック

2.4 手順 1 で作成した証明書 (server.crt) をアップロードします

設定のホーム

Salesforce Foundations

Salesforce Go

サービス設定アシスタント

Hyperforce アシスタント

リリース更新

Salesforce モバイルアプリケーション

Lightning 利用状況

オフマーケットプレイス

登録を管理

Sales Cloud Everywhere

管理

> ユーザー

> データ

> メール

プラットフォームツール

> アプリケーション

AppExchange マーケットプレイス

> Lightning Bolt

> Lightning Out 2.0 アプリケーション

アプリケーションマネージャー

> インテリジェントアプリケーション

> パッケージ

> モバイルアプリケーション

> 外部クライアントアプリケーション

OAuth 利用状況

API (OAuth 設定の有効化)

☒ OAuth を有効化

アプリケーション設定

*コールバック URL

*OAuth 範囲

利用可能な OAuth 範囲

ID URL サービスにアクセス (id, profile, email, address, phone)

API を使用してユーザーデータを管理 (api)

フルアクセス (full)

Connect REST API リソースにアクセス (chatter_api)

Visualforce アプリケーションにアクセス (visualforce)

一意のユーザー識別子にアクセス (openid)

選択した OAuth 範囲

Web ブラウザーを使用してユーザーデータを管理 (web)

いつでも要求を実行 (refresh_token, offline_access)

☐ すべてのトークンを調査

☐ ID トークンを設定

フローの有効化

☐ クラウドログイン情報フローを有効化

☐ 認証コードおよびログイン情報フローを有効化

☐ デバイスフローを有効化

☒ JWT ベアラーフローを有効化

証明書アップロード

またはファイルをドロップ

証明書を選択

☐ トークン交換フローを有効化

キャンセル 作成

外部クライアントアプリケーションの設定が完了しましたので、「作成」をクリックします。

推奨のツアー ▼ フィードバックを残す 28 トライアルの残り日数 今すぐ購入

設定 ホーム オブジェクトマネージャー ▼

クイック検索

設定のホーム

Salesforce Foundations

Salesforce Go

サービス設定アシスタント

Hyperforce アシスタント

リリース更新

Salesforce モバイルアプリケーション

Lightning 利用状況

オフマーケットプレイス

登録を管理

Sales Cloud Everywhere

管理

> ユーザー

> データ

> メール

プラットフォームツール

アプリケーション

AppExchange マーケットプレイス

> Lightning Bolt

> Lightning Out 2.0 アプリケーション

アプリケーションマネージャー

> インテリジェントアプリケーション

> パッケージ

> モバイルアプリケーション

外部クライアントアプリケーション

OAuth 利用状況

外部クライアントアプリケーションマネージャー

基本情報

* 外部クライアントアプリケーション名

アシロポ連携用

* API 参照名

アシロポ連携用

* 取引先責任者 メール

test@example.com

* 配信状態

ローカル

取引先責任者 電話

電話番号を入力...

情報 URL

URL を入力...

ロゴ画像 URL

URL を入力...

アイコン URL

URL を入力...

いずれかのサンプルロゴを選択してください。

説明

API (OAuth 設定の有効化)

☒ OAuth を有効化

アプリケーション設定

* コールバック URL

https://localhost

* OAuth 範囲

利用可能な OAuth 範囲

ID URL サービスにアクセス (id, profile, email, address, phone)

選択した OAuth 範囲

Web ブラウザーを使用してユーザーデータを管理 (web)

API を使用してユーザーデータを管理 (api)

いつでも要求を実行 (refresh_token, offline_access)

キャンセル 作成

2.5 OAuth ポリシーを設定します

推奨のツアー ▼ フィードバックを残す 28 トライアルの残り日数 今すぐ購入

設定 ホーム オブジェクトマネージャー ▼

アプリケーション

AppExchange マーケットプレイス

> Lightning Bolt

> Lightning Out 2.0 アプリケーション

アプリケーションマネージャー

> インテリジェントアプリケーション

> パッケージ

> モバイルアプリケーション

外部クライアントアプリケーション

OAuth 利用状況

外部クライアントアプリケーションマネージャー

設定

> 接続アプリケーション

> 機能設定

> Slack

> Data Cloud

> Heroku

> MuleSoft

> Einstein

> オブジェクトおよび項目

> イベント

外部クライアントアプリケーションを管理

assirobo_integration

このページのヘルプ 無効化

パートナーアプリケーションのセキュリティコントロールを確認してロックしてください。 コントロールを確認

取引先責任者 メール

test@example.com

アプリケーション認証

すべてのユーザーは自己承認可能

種類

ローカル

アプリケーションの状況

有効

ポリシー 設定 パッケージのデフォルト

ポリシーを設定して、この Salesforce 組織の外部クライアントアプリケーションおよびプラグインをカスタマイズします。

> アプリケーションポリシー

OAuth ポリシー

プラグインポリシー

許可されているユーザー

管理者が承認したユーザーは事前承認済み

OAuth 開始 URL

URL を入力...

カスタム範囲

現在、OAuth カスタム範囲が定義されていません。アプリケーションに割り当てる前にここで定義できます。

Apex プラグインクラス

クラスを検索

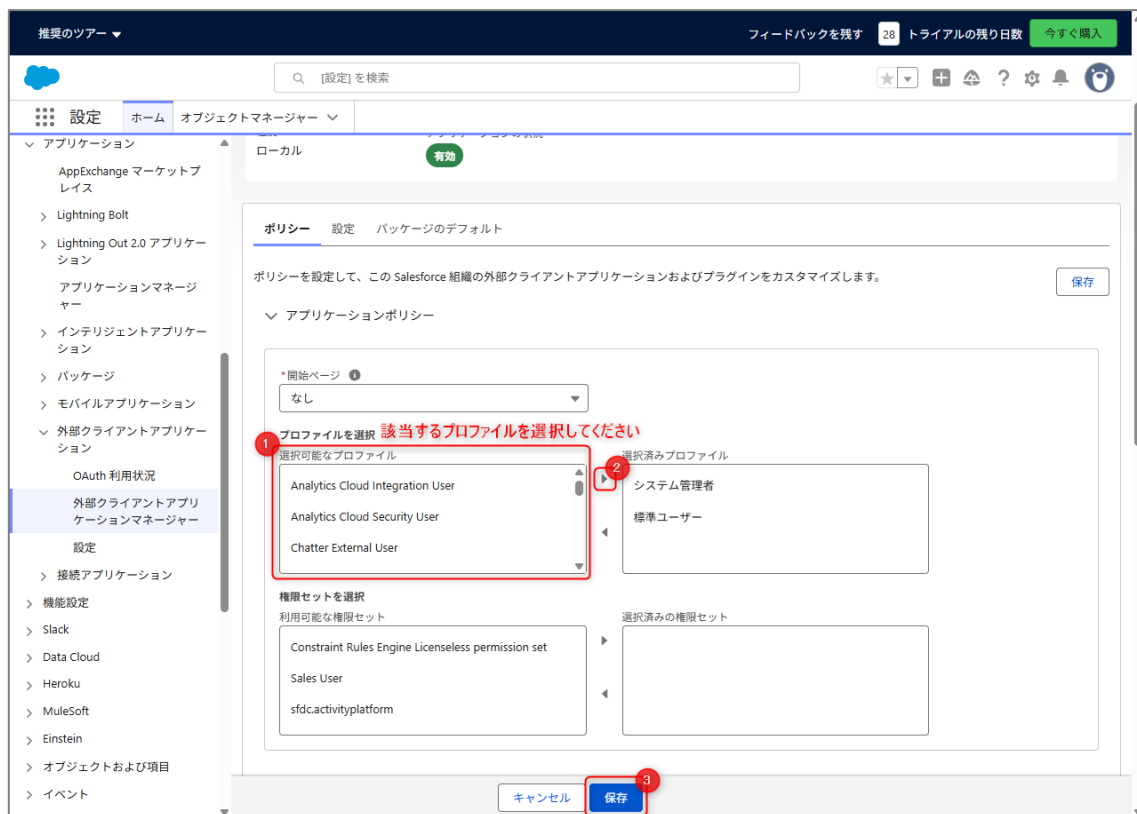
キャンセル 保存

項目	内容
許可されているユーザー	アシロボからのログインを許可するユーザを設定する。 「すべてのユーザーは自己承認可能」と「管理者が承認したユーザーは事前承認済み」が選択可能だが、「すべてのユーザーは自己承認可能」は手動承認が必要であるため、「管理者が承認したユーザーは事前承認済み」を選択

2.7 アプリケーションポリシーを変更します

アシロボからのログインを許可したユーザーに割り当てられているプロファイルを設定してください。

※プロファイルの確認方法は2枚目の画像に記載しています。



ユーザプロフィールの確認方法

推奨のツアー ▼ フィードバックを残す 21 トライアルの残り日数 今すぐ購入

設定 ホーム オブジェクトマネージャー ▼

Q クイック検索

設定のホーム

- Salesforce Foundations
- Salesforce Go
- サービス設定アシスタント
- Hyperforce アシスタント
- リリース更新
- Salesforce モバイルアプリケーション
- Lightning 利用状況
- オファーマーケットプレイス
- 登録を管理
- Sales Cloud Everywhere
- 管理
- ▼ ユーザー
 - キュー
 - プロフィール
 - ユーザー**
 - ユーザー管理設定
 - ロール
 - 公開グループ
 - 分析グループ
 - 権限セット
 - 権限セットグループ

設定 ユーザー

ユーザー Test

Salesforce では確認されたドメインからのユーザーが承認済みのメールのみを送信するようになりました。配信失敗を回避するには、今すぐメールアドレスを確認してください。 [ヘルプに移動](#)

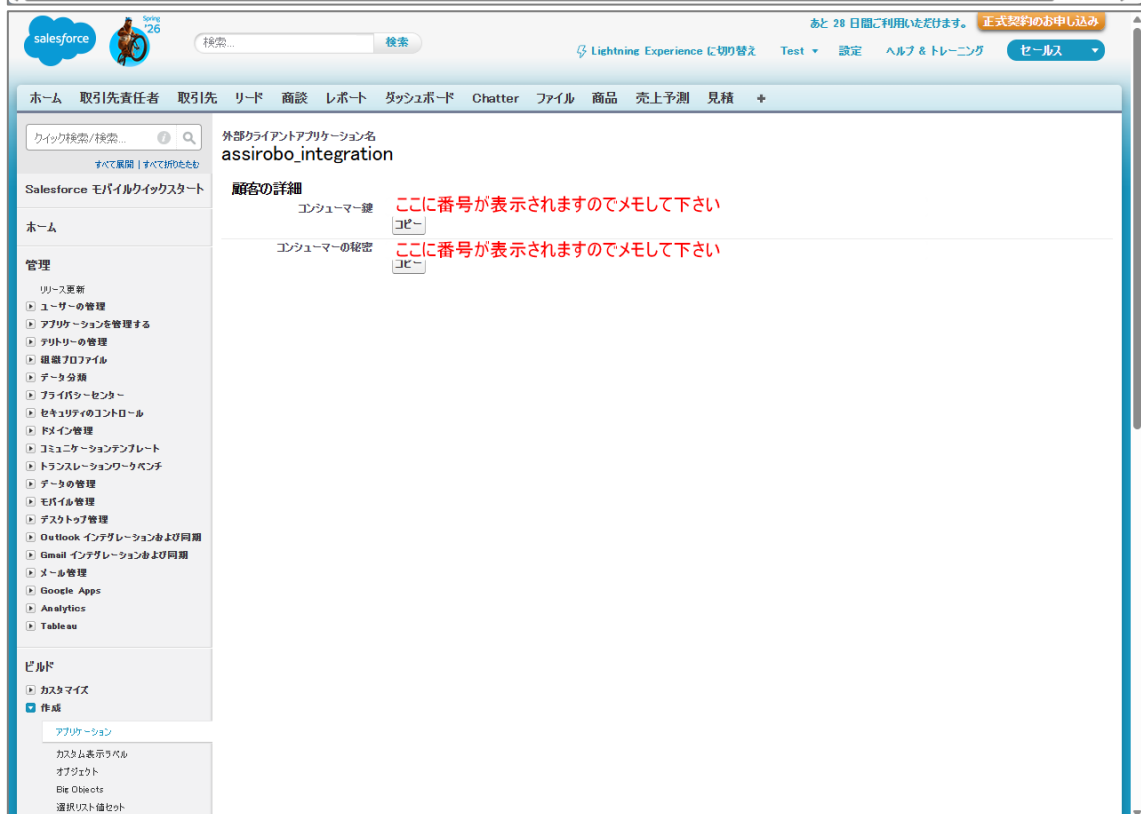
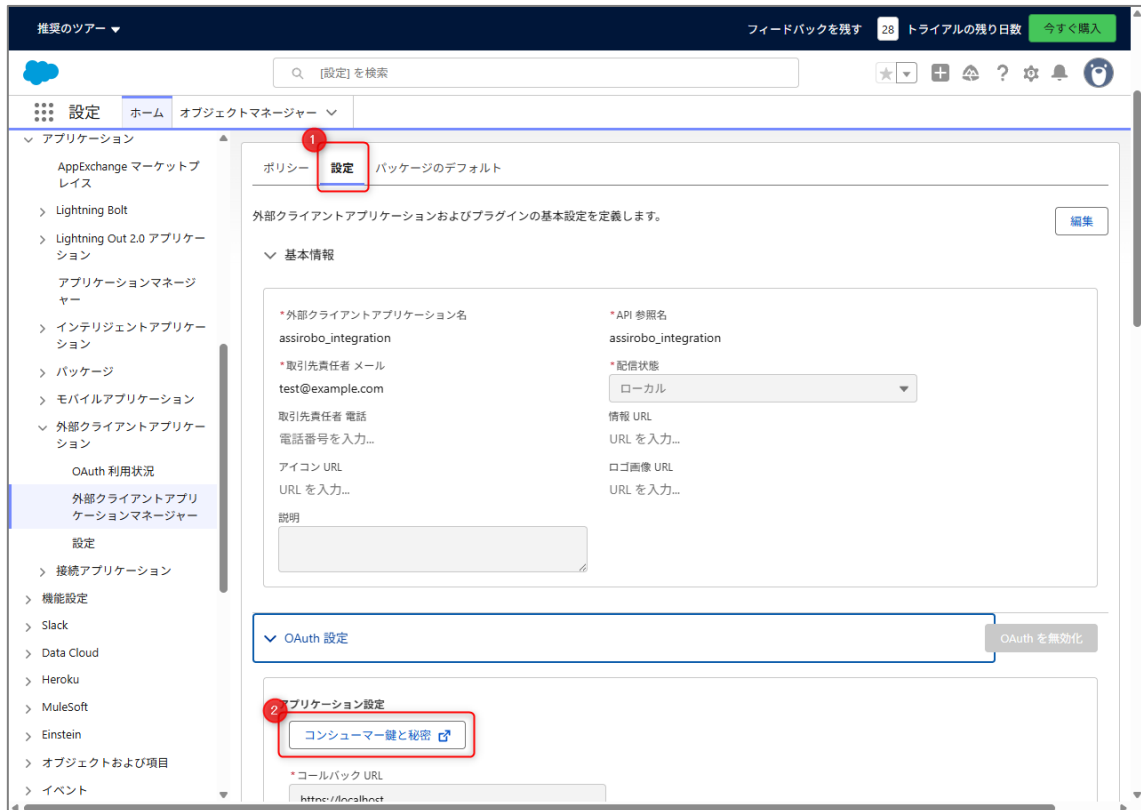
[権限セットの割り当て](#) | [権限セットの割り当てで有効化が必要](#) | [権限セットグループの割り当て](#) | [権限セットライセンスの割り当て](#) | [Lightning Data 購入の割り当て](#) | [非公開グループ](#) | [公開グループのメンバーシップ](#) | [キューのメンバー](#) | [ユーザーのスキル](#) | [チーム](#) | [ロール権限のマネージャー](#) | [OAuth アプリケーション](#) | [サードパーティアプリケーションのリンク](#) | [組み込み Authenticator](#) | [インストール済みモバイルアプリケーション](#) | [外部システムの認証設定](#) | [ロギング履歴](#) | [ユーザープロファイルのアカウント](#)

ユーザーの詳細 [編集](#) [共有](#) [パスワードの変更](#) [概要を表示](#)

名前	Test	ロール
別名		ユーザーライセンス Salesforce
メール		プロフィール システム管理者
ユーザー名		有効 ☒
ニックネーム		マーケティングユーザー ☒
役職		オフラインユーザー ☐
会社名		フローユーザー ☐
部署		モバイル転送の登録 ☒
デバイス		アクセシビリティモード (Classic のみ) ☐
住所	JP	デバッグモード ☐
タイムゾーン	(GMT+09:00) 日本標準時 (Asia/Tokyo)	グラフィックの高コントラストパレット ☐
地域	日本語 (日本)	スクロール中に Lightning ページを読み込む ☒
言語	日本語	Apex 警告メールの送信 ☐
代理承認者		設定画面をデフォルトページに設定 ☐
マネージャー		クイックアクセスメニュー ☒
承認申請メールを受信	自分承認者である場合のみ	開発モード ☐
統合 ID		開発モードでデフォルトを表示 ☐
アプリケーション登録: ワンタイムパスワード認証	接続	キャッチアップ診断 ☐
アプリケーション登録: Salesforce Authenticator	切断	売上予測を許可 ☒
セキュリティキー (U2F または WebAuthn)	登録	Checkout の有効化 ☒
Lightning Login	登録	MFU 更新なし ☐
...		ツールボックス

2.8 外部クライアントアプリケーションの「コンシューマー鍵」「コンシューマーの秘密」を取得します。

※この情報は認証に利用しますので、漏洩しないようご注意ください。



3. アシロボのシナリオ修正

Salesforce へのログインには主に「Web API」コマンドと「JWT エンコード」コマンドを利用します。「JWT エンコード」コマンドは、Salesforce の API を利用する際の身分証明データを作成するために使います。身分証明データを標準形式（JSON Web Token）に変換するコマンドです。

JWT エンコードコマンドと WebAPI コマンドの設定画面、サンプルシナリオは下記の通りです。

Json Web Tokenに変換

データ

+

Key	iss		Value	ここにコンシューマー鍵		×
Key	sub		Value	\${ユーザ名}		×
Key	aud		Value	https://login.salesforce.com		×
Key	exp		Value	\${有効期限}		×

秘密鍵

C://Users/...

パスワード設定方法

パスワード参照IDから

パスワード参照ID

PFXパスワード

データ参照ID

エンコードされた認証情報

[詳細表示]

メモ

コンシューマー鍵と秘密鍵を設定してください

× キャンセル

✓ OK

項目	内容
データ	JWT 形式に変換するデータを指定する。 ※Salesforce が指定する項目については下記に記載しています。
秘密鍵	データ変換に利用する秘密鍵を指定する。 拡張子は下記に対応している。 .pem .key .pfx .p12
パスワード設定方法	パスワード設定の方法を指定する。 以下の設定方法から選択できる。 1. パスワード無し 2. 直接入力 3. パスワード参照 ID 手順 1.3 で指定したパスワードを設定する
データ参照 ID	JWT 変換したデータを保持するデータ参照 ID 名を定義する

Salesforce が指定する身分証明データの項目

項目	内容
iss	証明書を登録した外部クライアントアプリケーション（もしくは接続アプリケーション）の コンシューマー鍵
sub	ログインするユーザ名
aud	認証サーバの URL。利用している環境に合わせて設定 ※本番環境、Develop Edition https://login.salesforce.com ※Sandbox 環境 https://test.salesforce.com ※Experience Cloud https://site.force.com/customers
exp	認証情報の有効期限。時間の形式は UNIX 時間

Web API

メソッド	POST		
テンプレート	自由入力		
URL	https://login.salesforce.com/services/oauth2/token		
クエリパラメータ	+		
ヘッダ	+		
ボディ	☒ Form-Data ☐ JSON ☐ Text		
Form-Data	+		
Key	grant_type	Value	urn:ietf:params:oauth:grant-type:jwt-bearer
Key	assertion	Value	\${エンコードされた認証情報}
タイムアウト (秒)	10		
メモ	Salesforce側に認証リクエストを送信します		

【注意】 API仕様については各サービスにお問い合わせください。

× キャンセル ✓ OK

項目	内容
メソッド	リクエストメソッドを指定します。 Salesforce へは POST で送信する必要がありますので、POST を設定します。
URL	リクエストの送信先を設定します。環境に合わせて設定してください。 ※本番環境、Develop Edition https://login.salesforce.com/services/oauth2/token ※Sandbox 環境 https://test.salesforce.com/services/oauth2/token ※Experience Cloud https://site.force.com/customers/services/oauth2/token

ボディ	ここでは Form-Data を選択します。Text でも送信可能です。
Form-Data	Salesforce に送信するデータを指定します。 grant_type: urn:ietf:params:oauth:grant-type:jwt-bearer assertion: エンコードされた認証情報 ※grant_type の値は Salesforce の仕様により、値が決まっています。

1

☐ データの記憶（文字）

文字: ここにログインユーザ名

ユーザ名

2

☐ パスワードの記憶 （秘密鍵のパスワードを入力してください）

パスワード設定方法: 直接入力

秘密鍵/パスワード

3

☐ 現在の時刻を記憶

時刻の形式: [UNIX時間], 先頭の0削除: 無し

時刻

4

☐ 計算結果を記憶 （このシナリオでは5分間（300秒）、URLを有効化します。）

演算式: 時刻 + 300

有効期限

5

☐ JSON Web Token生成 （コンシューマー鍵を設定してください）

エンコードされた認証情報

6

☐ Web API （Salesforce側に認証リクエストを送信します）

メソッド: POST
URL: https://login.salesforce.com/services/oa ...

[応答ステータス] [応答ヘッダ] [応答データ]

7

☐ JSON値取得

取得元データ参照ID: [応答データ], 取得値の型: オブジェクト（Object）, キー名: access_token

アクセストークン

8

☐ JSON値取得

取得元データ参照ID: [応答データ], 取得値の型: オブジェクト（Object）, キー名: instance_url

インスタンスURL

9

☐ ブラウザ起動 （ログイン済みの状態でSalesforceが開きます）

ブラウザ: アシロボブラウザ
開始URL: インスタンスURL /secur/frontdoor.jsp?sid= アクセストークン
ウィンドウ最大化: 有り

ブラウザ

10

Salesforceのサイトが表示されるまでに時間がかかる場合があります。
必要に応じて待機を入れてください。

コマンド	内容
1. データを記憶	ログインユーザ名を設定します。
2. パスワードの記憶	秘密鍵のパスワードを設定します。
3. 現在の時刻を記憶	UNIX 時間という形式で現在時刻を取得します。 ※ログイン URL の発行には有効期限を設定する必要があり、その形式が UNIX 時間である必要があるため、UNIX 時間形式で取得します。
4. 計算結果を記憶	URL を有効化する時間を設定します。

	※UNIX 時間は数値です。有効化したい秒数を加算すると、その秒数有効化できます。例えば、5 分間 (300 秒) 有効にしたい場合は、+300 します。
5. JSON Web Token 生成	身分証明データを作成します。
6. Web API	身分証明データを利用して、Salesforce へログインします。
7. JSON 値取得	ログインに成功すると、アクセストークンと URL が返ってきますので、アクセストークンを取得します。
8. JSON 値取得	ログインに成功すると、アクセストークンと URL が返ってきますので、URL を取得します。
9. ブラウザ起動	アクセストークンと URL を利用して、ログイン済みの状態で Salesforce を開きます。

6 番目の WebAPI コマンドで Salesforce の API を利用します。API で JWT データを送信することで、ログイン用の URL を発行できます。ログイン済みの状態で Salesforce を開いた後は、これまで通りブラウザ操作コマンドで操作できます。現在のログインを行っている箇所を置き換えていただきますようお願い致します。

API 経由でログイン URL が発行されますので、その URL にブラウザ起動コマンド等でアクセスする形になります。URL には有効期限がありますので、必要に応じて 4 番目の計算結果を記憶コマンドを利用して有効期限を延長してください。