日日是Oracle APEX

Oracle APEXを使った作業をしていて、気の付いたところを忘れないようにメモをとります。

2022年6月17日金曜日

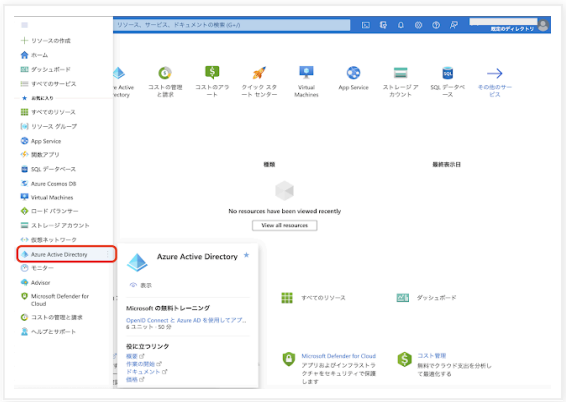
Azure ADを使ってAPEXアプリをSAMLで認証する

Azure ADを使って、Oracle APEXのアプリケーションをSAMLで認証させてみました。

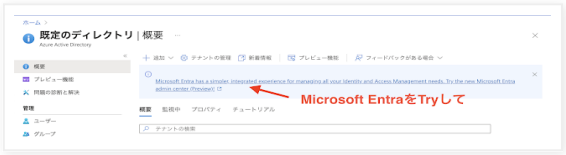
Oracle APEX側の準備は、おおむねOktaを使ってSAML認証の設定を行ったときと同じです。そのため、Azure AD側での作業を主に記述します。

以下、作業手順になります。

https://portal.azure.comにアクセスし、ナビゲーション・メニューから**Azure Active Directory**を開きます。



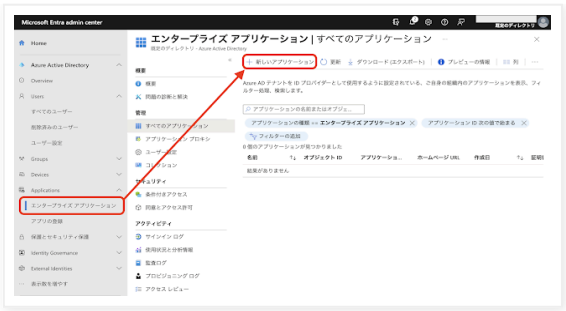
Microsoft EntraをTryしてみて、と通知が表示されているので、以降の作業はMicrosoft Entraで行ってみることにしました。



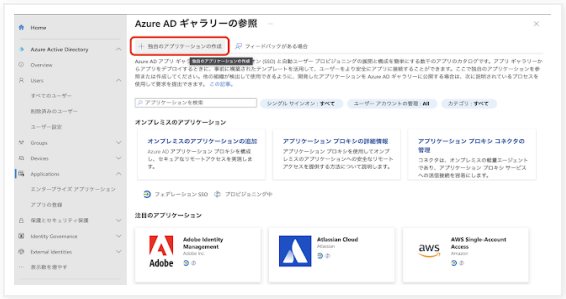
Microsoft Entra admin centerが開きます。

ナビゲーション・メニューの**Applications**の**エンタープライズアプリケーション**を開きます。

開いた画面で**＋新しいアプリケーション**をクリックします。



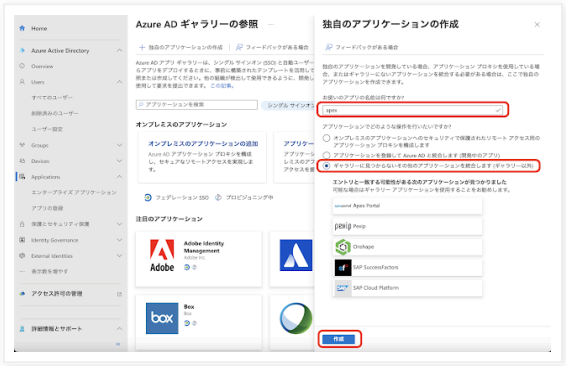
Oracle APEXは事前に連携がされているアプリケーションではないため、**＋独自のアプリケーションの作成**をクリックします。



右側にドロワーが開きます。**お使いのアプリの名前は何ですか？**という問い合わせに**apex**と入力します。Azure AD側ではアプリと言っていますが、Oracle APEXの場合は個々のアプリではなく、インスタンス全体をここでいうアプリとして登録します。そのため、ここではAPEXが動いているインスタンス名のような名前を指定することになるでしょう。

名前に**apex**と入力すると、**エントリと一致する可能性がある次のアプリケーションが見つかりました**、と候補が表示されます。選択できる候補は無いので、**ギャラリーに見つからないその他のアプリケーションを統合します（ギャラリー以外）**を選択します。

作成をクリックします。



エンタープライズアプリケーションとして**apex**が作成されます。

SAMLの設定は、**2。シングルサインオンの設定**に含まれています。今回の作業は、この他に**1。ユーザーとグループの割り当て**だけです。

2。シングルサインオンの設定の**作業の開始**のリンクをクリックします。

シングルサインオン方式の選択として、**SAML**を選択します。

SAMLによるシングルサインオンのセットアップとして、実施する手順が１から５まで順番に提示されます。

最初の**基本的なSAML構成**から実施します。**編集**をクリックします。

画面右にドロワーが開きます。

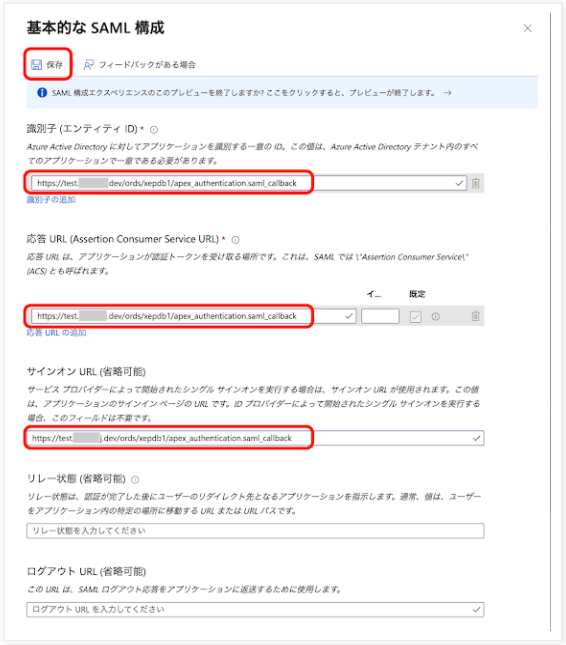
識別子(エンティティID)、**応答URL(Assertion Consumer Service URL)**、**サインオンURL(省略可能)**の３つすべてに、Oracle APEX側で定義されているSAMLコールバックURLを設定します。

以下のような形式のURLです。**apex_authentication.saml_callback**の部分は、どのインスタンスでも同じです。ベースとなるURLはOracle APEXが稼働している環境に合わせて変更します。

https://test.mydomain.dev/ords/xepdb1/apex_authentication.saml_callback

https://test.mydomain.dev/ords/apex_authentication.saml_callback

以上の設定を**保存**します。



２の**属性とクレーム**は、デフォルトのままにしておきます。

３の**SAML署名証明書**の**編集**をクリックします。

署名オプションを**SAML応答とアサーションへの署名**に変更します。**署名アルゴリズム**は**SHA-256**のまま変更しません。

以上の変更を実施し、**保存**します。

アクティブな証明書の操作メニューを開き、**PEM証明書のダウンロード**を実行します。**アプリケーション名.pem**、今回の例では**apex.pem**というファイル名で、PEM形式の証明書がダウンロードされます。

以上の作業を行い、ドロワーを閉じます。

Azure AD側のSAMLの設定は以上になります。

これからOracle APEX側の設定を行います。

Azure ADの**5**の**apexのセットアップ**（apexの部分はアプリケーション名で、作業によって変わります）の記述を参照します。

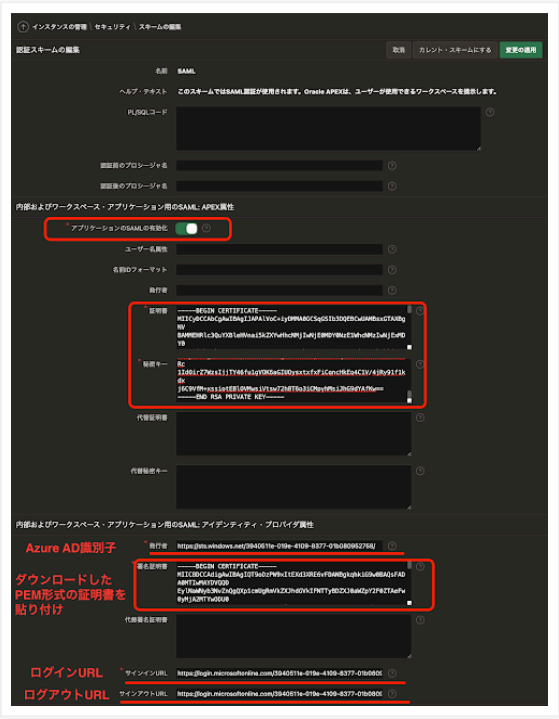
Oracle APEXの**管理サービス**にサインインし、**SAML**の構成画面を開きます。（ナビゲーション・パスは**インスタンスの管理**>**セキュリティ**>**認証制御**>**SAML**です。スクリーンショットはOktaの記事を参照してください。）

内部およびワークスペース・アプリケーション用のSAML: APEX属性の**アプリケーションのSAML有効化**を**ON**にします。**証明書**と**秘密キー**は、Oktaのときと同様にopensslを使って秘密キーと自己署名証明書を生成したものを貼り付けます。Oracle APEXの設定で必須なので指定しますが、SAML認証には使用されません。

内部およびワークスペース・アプリケーション用のSAML: アイデンティティ・プロバイダ属性を設定します。

発行者として**Azure AD識別子**、**署名証明書**はAzure ADから**ダウンロードした証明書**（今回であれば**apex.pem**の内容）、**サインインURL**として**ログインURL**、**サインアウトURL**として**ログアウトURL**を設定します。

以上で**変更の適用**をします。



ORDSのCORS設定で、アクセスを許可するOriginはhttps://login.microsoftonline.comでした。

ORDS 22.1では以下のコマンドを実行します。ordsコマンドのパスや構成ファイルの位置は、それぞれのインストールで異なります。

/usr/local/bin/ords --config /etc/ords/config config set security.externalSessionTrustedOrigins https://login.microsoftonline.com

ORDS 21.xでは以下のコマンドを実行します。

java -jar ords.war set-property security.externalSessionTrustedOrigins https://login.microsoftonline.com

変更を反映するには、ORDSの再起動が必要です。

以上でOracle APEX側の設定は完了です。

Azure AD側で**1。ユーザーとグループの割り当て**を行います。

＋ユーザーまたはグループの追加をクリックします。

ユーザーを選択します。現在Microsoft Entra admin centerを開いているユーザーは作成済みなので、そのユーザーを割り当てます。

選択されていませんのリンクをクリックします。

ENTERPRISE MOBILITY + SECURITY E5もしくは**AZURE AD PREMIUM P2**のサブスクリプションを購入するとグループの割り当てもできるとのことです。無料試用版も提供されています。今回の検証ではグループは使いません。

サインインを許可する**ユーザー**を**クリック**して**選択したアイテム**に移動させた後、**選択**をクリックします。

選択したユーザーで、**割り当て**を実行します。

以上でAzure ADとOracle APEXの双方の設定は完了です。

SAMLによるサインインを確認するために作成したアプリケーションにアクセスし、設定を確認します。

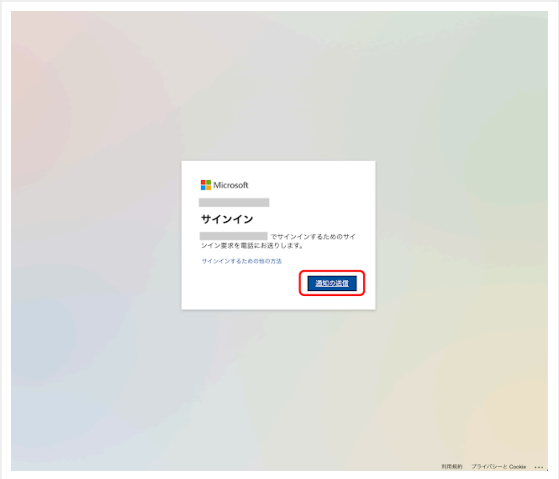
https://ホスト名/ords/PDB名/r/ワークスペース名/samltest/home

今回の例では、以下のURLにアクセスします。

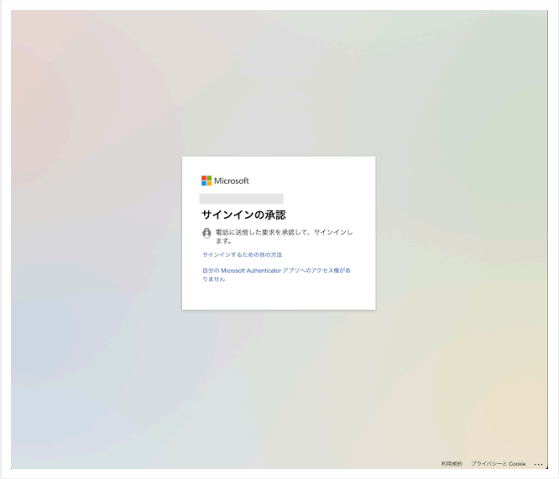
https://test.mydomain.dev/ords/xepdb1/r/apexdev/samltest/home

Microsoftのサインイン画面が表示されます。Microsoftとの契約状況（私はMIcrosoft 365の契約があります）やユーザーの設定状況によって、サインイン画面は違うのではないかと思います。

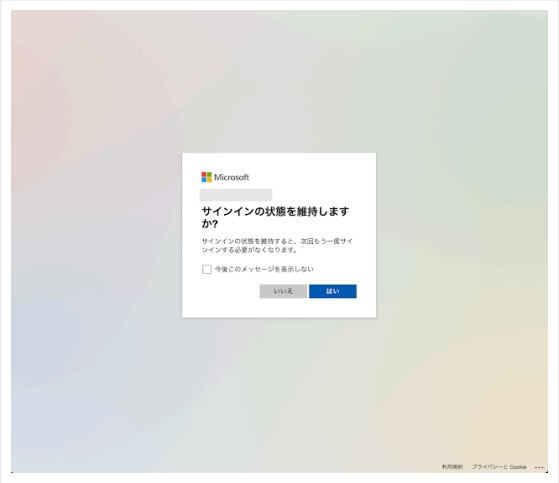
アカウントを選択すると、（私の場合は）**通知の送信**を行う画面が開きました。



Microsoft Authenticatorによる承認待ちになります。



サインインの状態を維持しますか？と確認されます。**はい**か**いいえ**のどちらかを選択します。



APEXのアプリケーションの画面が開きます。**APP_USER**の値が非常に長くなっています。

この値は、Azure AD側の**属性とクレーム**として設定されている**一意のユーザーID**になります。

APEXのアプリケーションのAPP_USERもシステムで一意でなければならない値なので、Azure AD側で一意性が保証されている値を設定する必要があります。

そのため、対応としてはAPP_USER自体を変更するのではなく、アプリケーション側でAPP_USERを表示に使用している部分を、人が見てわかる表示（姓名など）に変更します。

Azure ADのデフォルト設定では、SAML 2.0 Assertionに表示名（**Attribute**の**Name**属性が**http://schemas.microsoft.com/identity/claims/displayname**）と電子メール・アドレス（**Attribute**の**Name**属性が**http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress**）が含まれています。これを**認証スキーム**の**認証後のプロシージャ**で取り出し、アプリケーション・アイテム**G_DISPLAY_NAME**、**G_EMAILADDRESS**に設定します。

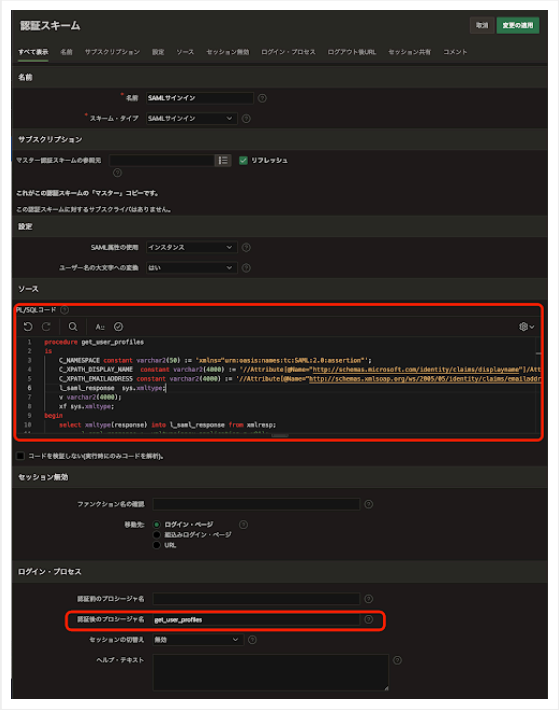
共有コンポーネントの**アプリケーション・アイテム**として、**G_DISPLAY_NAME**、**G_EMAILADDRESS**を作成します。

表示名と電子メール・アドレスの取り出しコードは、以下になります。

```
procedure get_user_profiles
is
    C_NAMESPACE constant varchar2(50) := 'xmlns="urn:oasis:names:tc:SAML:2.0:assertion"';
    C_XPATH_DISPLAY_NAME  constant varchar2(4000) := '//Attribute[@Name="http://schemas.microso
    C_XPATH_EMAILADDRESS constant varchar2(4000) := '//Attribute[@Name="http://schemas.xmlsoap.
    l_saml_response  sys.xmltype;
    v varchar2(4000);
    xf sys.xmltype;
begin
    l_saml_response := xmltype(apex_application.g_x01);
    xf := l_saml_response.extract(C_XPATH_DISPLAY_NAME, C_NAMESPACE);
    v := xf.getstringval();
    apex_util.set_session_state('G_DISPLAY_NAME', v);
    apex_debug.info('displayname = ' || v);
    xf := l_saml_response.extract(C_XPATH_EMAILADDRESS, C_NAMESPACE);
    v := xf.getstringval();
    apex_util.set_session_state('G_EMAILADDRESS', v);
    apex_debug.info('emailaddress = ' || v);
end get_user_profiles;
```

get_user_profiles.sql hosted with ❤️ by GitHub　　view raw

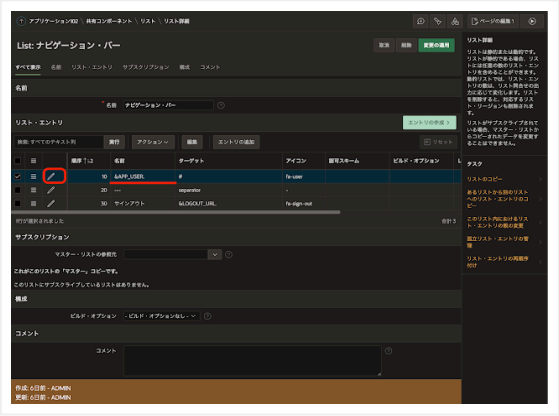
このPL/SQLコードを認証スキーム**SAMLサインイン**の**ソース**の**PL/SQLコード**に記述し、**ログイン・プロセス**の**認証後のプロシージャ名**として**get_user_profiles**を設定します。



ナビゲーション・バーの表示を変更するため、**共有コンポーネント**の**ナビゲーション・バー・リスト**を開きます。

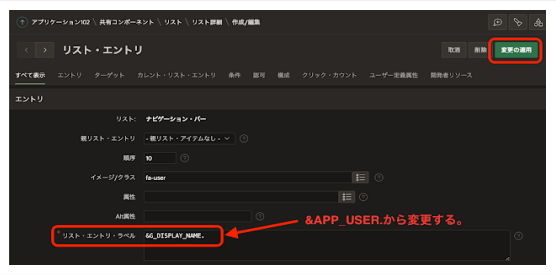
ナビゲーション・バーを開きます。

名前が**&APP_USER.**となっている**リスト・エントリ**の**鉛筆アイコン**をクリックします。編集画面が開きます。



リスト・エントリ・ラベルを&APP_USER.から&G_DISPLAY_NAME.に変更します。

変更の適用をクリックすると、ナビゲーション・メニューの変更は完了です。



ユーザー名が小文字のみで表示されないようにするには、同じページの**ユーザー定義属性**の**List Item CSS Classes**として設定されている**has-username**を削除します。

以上の変更を行い、再度、SAML認証の確認アプリケーションにアクセスします。

以上で、Azure ADを使ってAPEXアプリをSAMLで認証するための作業は終了です。

サブスクリプションの関係で、Azure ADでは認証後のプロシージャにてダイナミック・グループの割り当ては行っていません。おそらくAzure ADでもAssertionにグループ情報が含まれることになるので、Oktaの処理とおおむね同じコードで対応できると思います。

完

Yuji N. 時刻: 0:05

共有

‹ ホーム ›

ウェブ バージョンを表示

自己紹介

Yuji N.

日本オラクル株式会社に勤務していて、Oracle APEXのGroundbreaker Advocateを拝命しました。
こちらの記事につきましては、免責事項の参照をお願いいたします。

詳細プロフィールを表示