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7月14日木曜日

APEX 22.1の永続認証を使ってみる

Oracle APEX 22.1より**永続認証（Persistent Authentication）**という機能が追加されています。

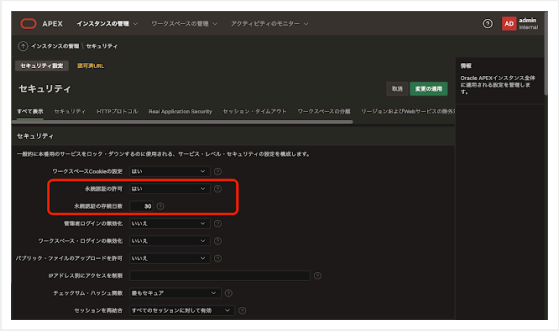
Oracle APEX Office HoursのNew in APEX 22.1 (Part 5): UI, PWA and Mobileで、この機能を開発したChristian Rokittaさんが解説をしています。

Oracle APEXが扱うクッキーについては、Oracle APEXが使用するクッキーという記事で解説をしています。この記事で説明しているクッキーに加えて、新たに認証情報を保持する永続クッキーが追加されています。追加されたクッキーの名前は、セッションの維持に使用するクッキー（デフォルトでは**ORA_WWV_APP_application_id**という形式）の末尾に**$P**が付加されます。

APEXのセッションが無効になった後に（セッションのタイムアウト、URLに含まれるセッションIDの不一致、ブラウザの終了など）、永続クッキーの値を使ってセッションを新規作成します。サインイン画面が再表示されることはありません。

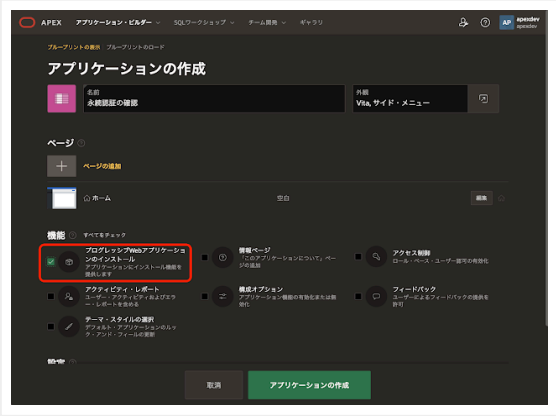
永続認証の機能を利用する準備として、Oracle APEXの**管理サービス**の**インスタンスの管理**より、**セキュリティ**の**永続認証の許可**を**はい**に変更する必要があります。

永続認証の許可を**はい**にすると、**永続認証の存続日数**の指定が現れます。デフォルトは**30**日です。



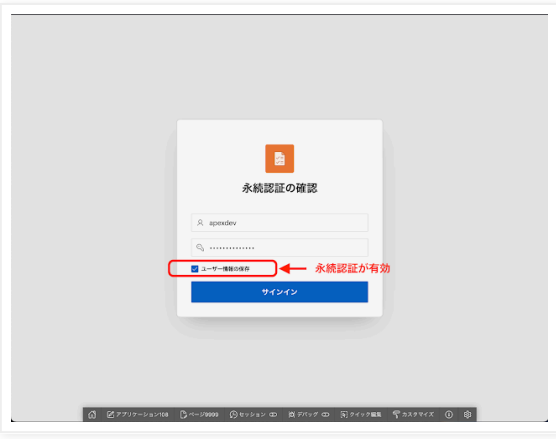
インスタンス・レベルで永続認証を許可するだけで、APEXアプリケーションの永続認証が有効になるわけではありません。APEXアプリケーションのログイン・ページに、永続認証を有効にする処理を組み込む必要があります。

Oracle APEX 22.1では、アプリケーションの作成時に**プログレッシブWebアプリケーションのインストール**にチェックが入っていると、永続認証に関わる処理がログイン・ページに組み込まれます。

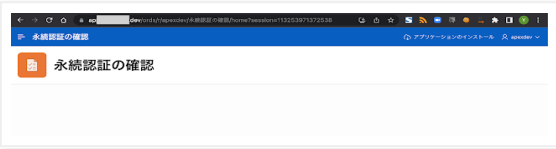


認証の確認を行なうだけなので、特別な構成はせず**アプリケーションの作成**を行います。作成したアプリケーションを実行すると、以下のログイン画面が表示されます。

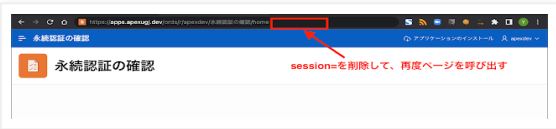
ユーザー情報の保存にチェックを入れた上で**サインイン**を行います。サインインに成功した時点で永続認証が有効になります。



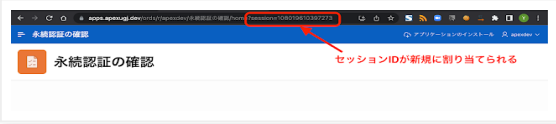
サインインが成功し、アプリケーションの画面が開きます。



URLに**session=**という形式で**セッションID**が含まれています。この**session=をURLから削除**し再度アクセスします。



永続認証が有効な場合、サインイン画面は表示されずに**新規にセッションが開始**します。URLの引数session=のセッションIDが変わっています。

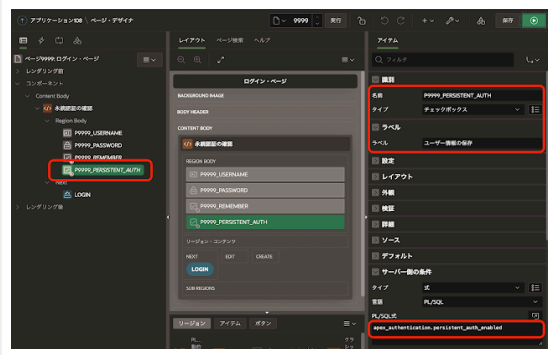


これと似た動きをする機能に**セッションの再結合**（こちらの記事で説明しています）があります。**セッションの再結合**の場合は、URLにセッションIDが含まれていない、もしくはsession=0が与えられているときに、クッキーに設定されているセッションIDだけでセッションを継続します。セッションを新規に開始するといった動作ではありません。

ページ・デザイナでログイン・ページを開き、永続認証の実装を確認します。

ページ・アイテム**P9999_PERSISTENT_AUTH**が、永続認証を有効にするチェックボックスです。**ラベル**は**ユーザー情報の保存**となっています。

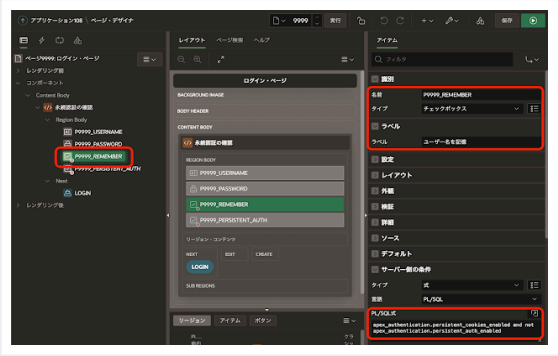
サーバー側の条件として、**PL/SQL**の**式**apex_authentication.persistent_auth_enabledが指定されています。インスタンス・レベルで永続認証が有効になっているときに限り、このチェックボックスが表示されます。



APEX 22.1以前にあった**ユーザー名を記憶**のチェックボックスは、サーバー側の条件が以下のように変わっています。

apex_authentication.persistent_cookies_enabled and not apex_authentication.persistent_auth_enabled

インスタンスの設定で、**永続認証の許可**が**いいえ**で**ワークスペースCookieの設定**が**はい**である場合は、以前と同じく**ユーザー名を記憶**のチェックボックスが表示されます。



ページ・アイテム**P9999_PERSISTENT_AUTH**の値は、プロセスloginで呼び出されるAPEX_AUTHENTICATION.LOGINの引数として与えられます。引数**p_set_persistent_auth**はAPEX 22.1で新たに追加されました。

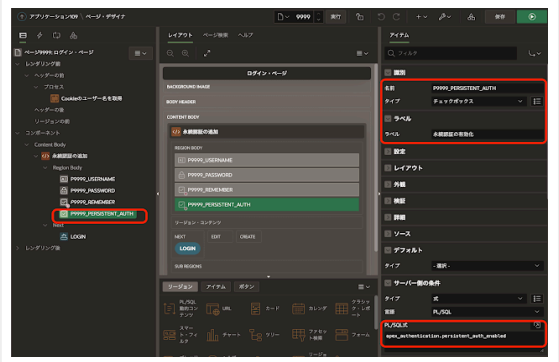
```
apex_authentication.login(
   p_username => :P9999_USERNAME,
   p_password => :P9999_PASSWORD,
   p_set_persistent_auth => nvl(:P9999_PERSISTENT_AUTH, 'N') = 'Y' );
```

p_set_persistent_authがtrueである場合、認証情報を含んだ永続クッキー（名前の末尾が$P）がブラウザにセットされます。

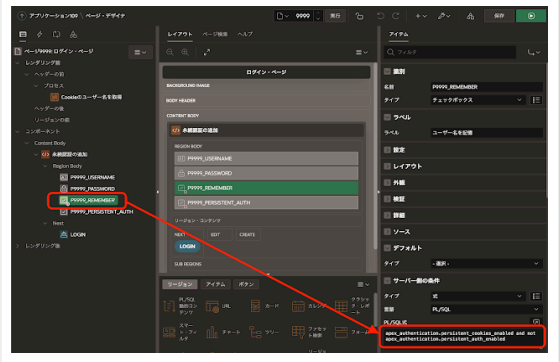
プロシージャAPEX_AUTHENTICATION.LOGINが呼び出されないと永続認証は有効になりません。アプリケーションのログイン・ページを表示しない認証スキーム、例えばソーシャル・サインイン（Open ID Connect、OAuth2）やSAMLの場合はAPEX_AUTHENTICATION.LOGINは呼ばれないことから、現時点では永続認証を許可することはできません。

ログイン・ページに永続認証の機能が組み込まれるのは、アプリケーション作成ウィザードで機能としてPWAを選択したときなので、PWAが有効でないと永続認証が使えないような印象があります。実際は、PWAが有効化されていなくても永続認証は有効にできます。

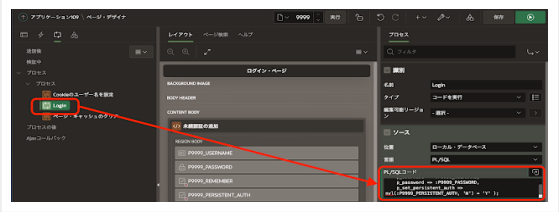
手作業でページ・アイテム**P9999_PERSISTENT_AUTH**を作成します。



ページ・アイテム**P9999_REMEMBER**の**サーバー側の条件**を変更します。



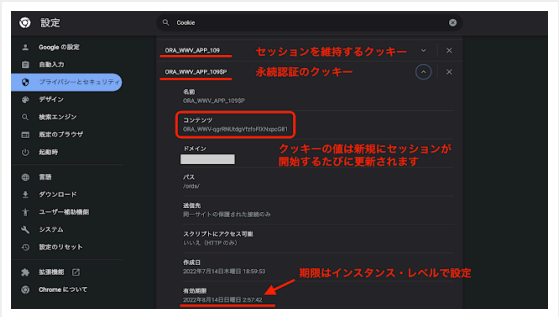
プロセス**login**の**ソース**の**PL/SQLコード**を変更します。



以上で、アプリケーションの永続認証の対応は完了です。

永続認証が扱うクッキーを、ブラウザより確認します。クッキー名はセッションを維持するクッキー名に**$P**が付加された名前になっています。**有効期限**は**インスタンスの管理**の**セキュリティ**にて、**永続認証の存続日数**として設定した日数を反映しています。

クッキーの値（画面ではコンテンツ）は、新規にセッションが開始する度に更新されます。もし仮にクッキーの値が漏洩しても、そのクッキーの値を使えるのは最長でもセッションがタイムアウトするまでになります。



最後にアプリケーション・プロセスが実行されるタイミングを確認しました。確認のために以下のDDLを実行し、表TEST_POST_AUTHを作成しました。

```
-- create tables
create table test_post_auth (
    id                             number generated by default on null as identity
                                   constraint test_post_auth_id_pk primary key,
    app_id                         number,
    point                          varchar2(80 char),
    created                        date not null,
    created_by                     varchar2(255 char) not null,
    updated                        date not null,
    updated_by                     varchar2(255 char) not null
)
;


-- triggers
create or replace trigger test_post_auth_biu
    before insert or update
    on test_post_auth
```

```
    for each row
begin
    if inserting then
        :new.created := sysdate;
        :new.created_by := coalesce(sys_context('APEX$SESSION','APP_USER'),user);
    end if;
    :new.updated := sysdate;
    :new.updated_by := coalesce(sys_context('APEX$SESSION','APP_USER'),user);
end test_post_auth_biu;
/

-- load data
```

create_test_post_auth.sql hosted with ❤️ by GitHub view raw

アプリケーション・プロセスを作成します。**プロセス・ポイント**を**新規インスタンス(新規セッション)開始時**とします。実行するコードは以下です。

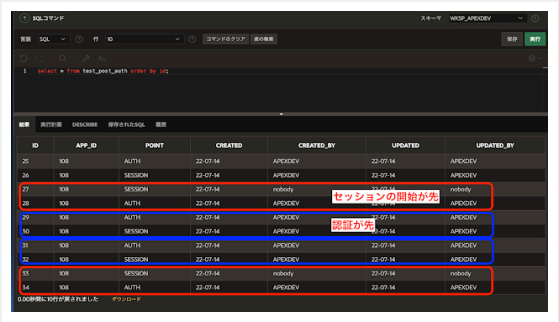
```
begin
   insert into test_post_auth(app_id, point) values(:APP_ID, 'SESSION');
   commit;
end;
```

もうひとつアプリケーション・プロセスを作成します。**プロセス・ポイント**を**認証後**とします。実行するコードは以下です。

```
begin
   insert into test_post_auth(app_id, point) values(:APP_ID, 'AUTH');
   commit;
end;
```

アプリケーション・プロセスの実行結果を、表TEST_POST_AUTHの内容から確認します。

select * from test_post_auth order by id;



ログイン画面で認証し新規セッションが開始されるときは、プロセス・ポイントの**新規インスタンス(新規セッション)開始時**が先に呼び出されます。認証の前に呼び出されているため、APP_USERに値は設定されずnobodyになります。その後に**プロセス・ポイント**の**認証後**のコードが実行されます。

永続認証によって新規セッションが開始される場合は、**プロセス・ポイント**の**認証後**が先に呼び出されています。その後に**新規インスタンス(新規セッション)開始時**のコードが実行されています。

すでに作成済みのAPEXアプリケーションに、プロセス・ポイントが**新規インスタンス(新規セッション)開始時**または**認証後**となっているアプリケーション・プロセスが作成されている場合は、永続認証を有効にすることによる影響の有無を確認した方が良いでしょう。

例えば**プロセス・ポイント**の**新規インスタンス(新規セッション)開始時**のコードでAPP_USERは必ずnobodyであることを前提としていたり、**プロセス・ポイント**の**認証後**の前に必ず**新規インスタンス(新規セッション)開始時**のコードが実行されることを前提としているコードがあると、問題が発生する可能性があります。

パッケージAPEX_AUTHENTICATIONには永続認証のサポートに伴って、以下の２つのプロシージャが追加されています。

プロシージャREMOVE_CURRENT_PERSISTENT_AUTHを呼び出すことにより、実行中のアプリケーションの永続認証の設定を無効にすることができます。また、管理者はプロシージャREMOVE_PERSISTENT_AUTHを呼び出すことにより、指定したユーザーの永続認証の設定を無効化することができます。

APEX 22.1の新機能である永続認証の説明は以上になります。

完

Yuji N. 時刻: 20:04

共有

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