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。

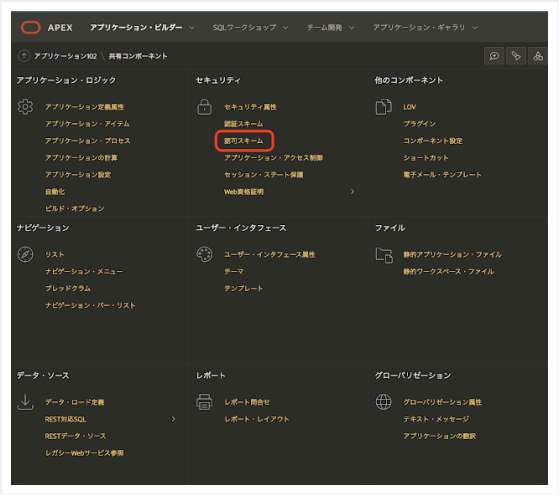
2020年12月31日木曜日

アクセス制御の実装サンプル解説(6) - 認可スキーム

こちらの記事の継続で、本シリーズの最後の記事になります。

今までは認可スキームありきで、各種のページ・タイプへアクセス制御を実装してきました。この記事では、認可スキームについて説明します。

認可スキームは共有コンポーネントとして登録されます。登録済みの認可スキームを確認してみましょう。**共有コンポーネント**から**認可スキーム**を開きます。



コントリビューション権限、**リーダー権限**、**管理権限**の３つが登録済みです。

この認可スキームは**アプリケーション作成ウィザード**の**機能**で、**アクセス制御**に**チェック**を入れているとウィザードによって、アプリケーションに作成されます。

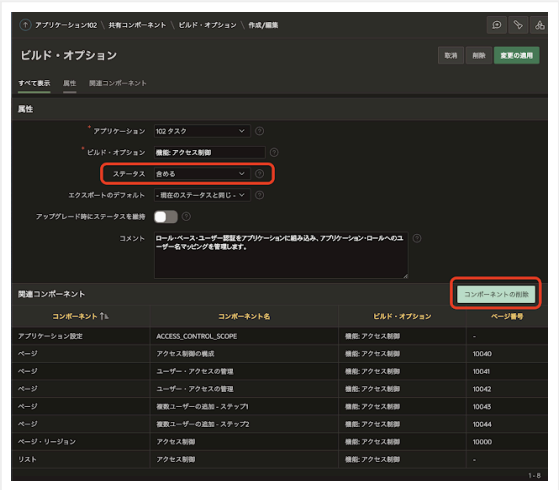
または、**ページ作成ウィザード**で**ページ・タイプ**として**機能**を選択し、**アクセス制御**を追加することもできます。

横道にそれますが、導入されたアクセス制御のコンポーネントは**ビルド・オプション**の**機能: アクセス制御**に紐づけられています。

共有コンポーネントに**ビルド・オプション**があります。

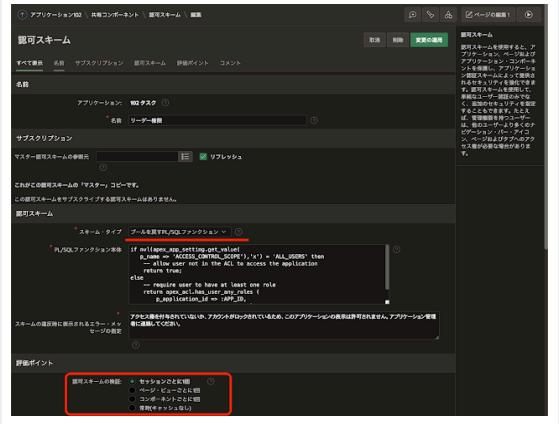
これを開くと、ビルド・オプションの一覧が確認できます。アプリケーション作成ウィザード、ページ作成ウィザードで追加できる機能は（ログイン・ページを除き)、ビルド・オプションに紐づけられています。**機能: アクセス制御**をクリックして開いてみます。

アクセス制御に関連するコンポーネントの確認や、無効化(**ステータス**を**除外**へ変更)、機能の削除(**コンポーネントの削除**）が可能であることがわかります。



さて、**認可スキーム**です。実体が分かりやすいのは**リーダー権限**です。こちらを開いてみます。

リーダー権限は、**スキーム・タイプ**としては**ブールを戻すPL/SQLファンクション**として実装されています。



スキーム・タイプは他に以下の種類があります。

- EXISTS SQL問合せ
- NOT EXISTS SQL問合せ
- ブールを戻すPL/SQLファンクション
- ロールまたはグループ内にある
- ロールまたはグループ内にない
- 式1のアイテムがNOT NULL
- 式1のアイテムがNULL
- 式1のアイテムの値 != 式2
- 式1のアイテムの値 = 式2
- 式1のプリファレンスの値 != 式2
- 式1のプリファレンスの値 = 式2

これらは還元すると、**ブールを戻すPL/SQLファンクション**になります。つまり、真偽値を返すPL/SQLファンクションが認可スキームです。戻り値がTRUEであればアクセスが許可され、FALSEであればアクセスが許可されません。

この認可スキームとなるファンクションが呼び出されるタイミングが**評価ポイント**であり、認可スキームの検証の頻度が高いとパフォーマンス面ではマイナスになりますが、権限の変更の反映が早くなります。

例えば**認可スキームの検証**が**セッションごとに1回**であると、サインインした後、再度サインインするまで権限の状態は変わりません。一般にOracle APEXのアプリケーションではサインアウトを意図的に行うことは少ないので、セッションのタイムアウトまで一旦許可されたアクセス権限は、そのまま維持されることになります。

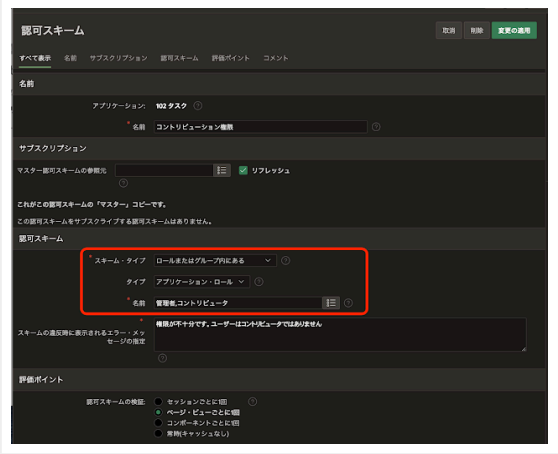
ページ・ビューごとに1回であれば、サインアウト/サインインを行うことなく、新たにページを表示するたびにアクセス権限が評価され、変更された権限が適用されます。

リーダー権限として定義されているコードを見てみましょう。

```
if nvl(apex_app_setting.get_value(
  p_name => 'ACCESS_CONTROL_SCOPE'),'x') = 'ALL_USERS' then
   -- allow user not in the ACL to access the application
   return true;
else
   -- require user to have at least one role
   return apex_acl.has_user_any_roles (
      p_application_id => :APP_ID,
      p_user_name      => :APP_USER);
end if;
```

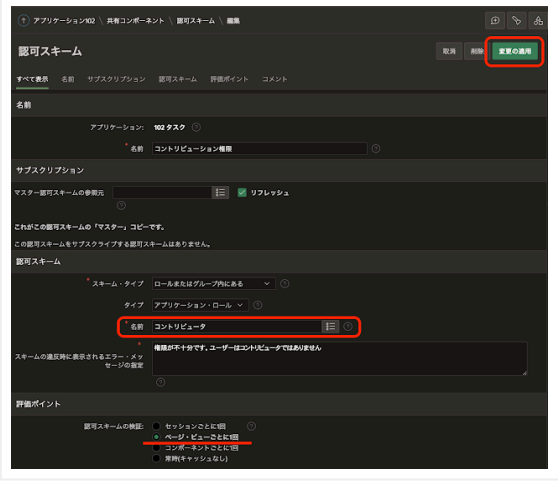
アプリケーション定義の**ACCESS_CONTROL_SCOPE**が**ALL_USERS**であればTRUEを返す、それ以外は**apex_acl.has_user_any_rolesの結果**(何かロールが登録されていればTRUE)を返しています。

次に**コントリビューション権限**を見てみましょう。



コントリビューション権限の**スキーム・タイプ**として、**ロールまたはグループ内にある**が設定され、**タイプ**が**アプリケーション・ロール**、**名前**が**管理者、コントリビュータ**となっています。つまり、ユーザーがアプリケーション・ロールとして管理者かコントリビュータ・ロールを持っているとコントリビューション権限がTRUEとなります。

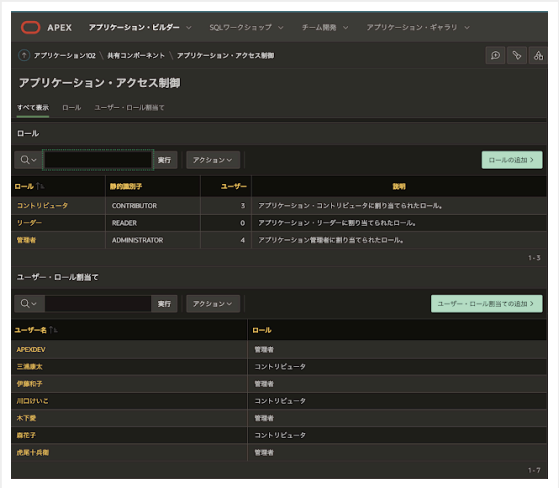
今回の要件では、管理者はデータの編集を行なわず(自分が担当者の場合に限定)、コントリビューション権限は持たないことにしているので、**名前**から**管理者**を除きます。



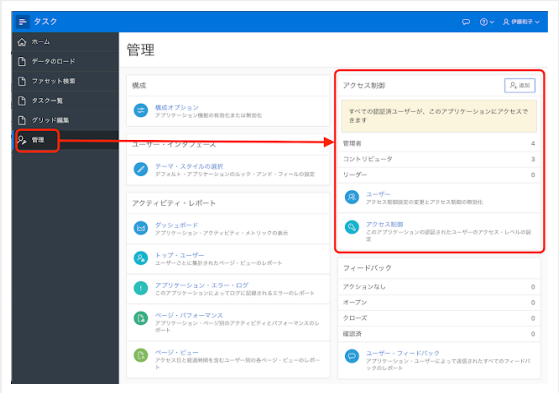
管理権限は、こちらの**名前**が**管理者**のみ(アプリケーション・ロールとして管理者を持っている)の認可スキームです。コントリビューション権限と大差ないので確認は省きます。

次に、アプリケーション・ロールの設定を確認します。**共有コンポーネント**から**アプリケーション・アクセス制御**を開きます。

アプリケーションに登録されているアプリケーション・**ロール**と**ユーザー・ロール割当て**として、ユーザーに割り当てられたアプリケーション・ロールを確認することができます。



共有コンポーネントの**アプリケーション・アクセス制御**を開いて、**ユーザー・ロール割当て**を実施するには、Oracle APEX開発環境にログインする必要があります。つまりOracle APEXの開発者アカウントが必要ですが、**機能**として**アクセス制御**がアプリケーションに組み込んであれば、サイド・メニューの**管理**より**アクセス制御の設定を変更**したり、ユーザーへの**アプリケーション・ロールの追加/削除**をすることができます。



機能: アクセス制御によって提供されるアクセス制御は、これを導入しないとアプリケーションのアクセス制御ができない、といったものではありません。

そうではなく、以下の作業をアプリケーションに行います。

- アプリケーション・ロールとして、管理者、コントリビュータ、リーダーを登録する。
- アプリケーション定義にACCESS_CONTROL_SCOPEを登録し、ACL_ONLYを設定する。
- 認可スキームとして管理権限(アプリケーション・ロールの管理者を持っている)、コントリビューター権限(アプリケーション・ロールの管理者かコントリビュータを持っている)、リーダー権限(アプリケーション定義がALL_USERSであるか、または、何かひとつでもアプリケーション・ロールを持っている）の３つを登録する。

- 管理権限を持ったユーザーによって、アプリケーション定義ACCESS_CONTROL_SCOPEの変更と、ユーザーへのアプリケーション・ロールの割当てを可能とする画面を、アプリケーションに登録する。

アプリケーションを開発する側としては、自力で同様の機能をアプリケーションに組み込むこともできますが、提供されている機能が開発するアプリケーションの要件を満足するのであれば、わざわざ開発する必要はなく、そのまま利用することで開発工数の削減になります。アプリケーション自体に組み込まれた設定であるため、組み込まれた後の認可スキームやページは自由に改変可能です。

ウィザードのよって組み込まれるアクセス制御の機能は、そのアプリケーションに閉じた作業です。複数のアプリケーションに跨ったアプリケーション・ロールの登録やユーザーへのロールの割り当てといった機能は含まれていません。そのような機能が必要な場合は、認可スキームを作成する必要があります。

伝統的な企業アプリケーションであれば、LDAPを認証/認可のためのサーバーとして使用していることが多いと思います。Oracle APEXではLDAPの扱いを容易にするためのAPI、APEX_LDAPを提供しています。これらのAPIを使った認可スキームの実装については、また記事を改めて行えたらと思います。

今回の認可に関するシリーズは以上になります。

完

Yuji N. 時刻: 16:00

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