

日々是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。

2021年2月5日 金曜日

コンポーネントの表示制御について

コンポーネントの表示/非表示を制御する方法には、いくつか種類があります。それぞれ、簡単に紹介しようと思います。紹介する機能は、以下の4つです。

- ビルド・オプション
- 認可スキーム
- サーバー側の条件
- 動的アクションとクライアント側の条件



より正確には表示/非表示に限らず、**コンポーネントの有効化/無効化を行います**。目で見えて分かりやすい有効化/無効化の効果が表示/非表示なので、それを例にとっています。表示されないプロセスやその他のコンポーネントにたいしても、同様に効果があります。

ビルド・オプション

ビルド・オプションについては[以前に記事を書いています](#)。ここでは簡単に設定について紹介します。

ビルド・オプションは**共有コンポーネントのアプリケーション・ロジック**に含まれる**ビルド・オプション**を開いて作成します。



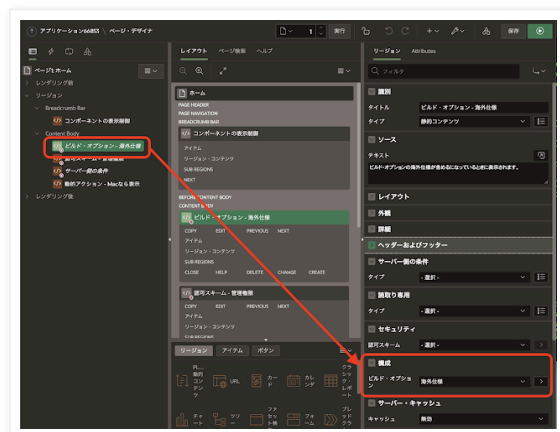
今回は海外仕様というビルド・オプションを作成しています。



表示、非表示の制御はステータスによって決まります。コンポーネントのビルド・オプションとして、海外仕様が設定されている場合、このステータスが**含む**であれば、リージョンは表示されます。逆に**除外**であれば、表示されません。



ほとんどのコンポーネントは、ビルド・オプションの設定を持っています。プロパティの構成に含まれる**ビルド・オプション**にて、コンポーネントにビルド・オプションを設定します。



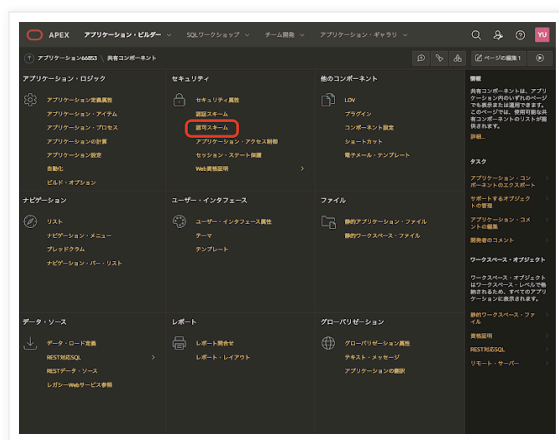
コンポーネントがリージョン、ページ・アイテムやボタンであれば、画面表示に関わるものなので表示/非表示が分かりやすい効果になります。プロセスなどのコンポーネントであれば、実行がされません。

ビルド・オプションにて除外されている場合は、そのコンポーネントがアプリケーション自体に存在しない、という扱いになります。

認可スキーム

認可スキームについても、[以前に記事を書いています](#)。こちらでも簡単に設定の紹介をします。

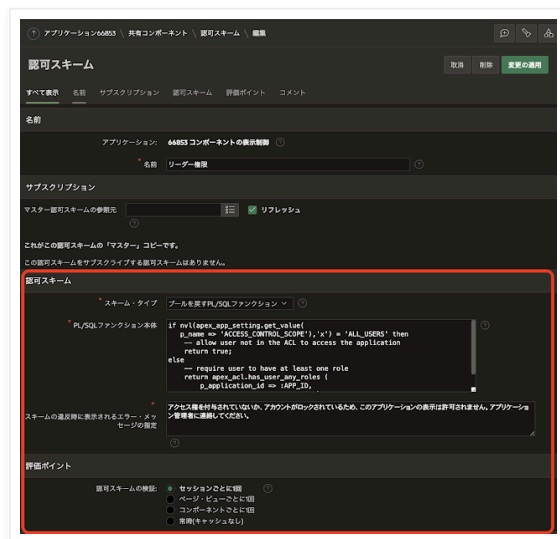
認可スキームはは**共有コンポーネント**の**セキュリティ**に含まれる**認可スキーム**を開いて作成します。



実体が分かりやすい**リーダー権限**を参照することで、認可スキームについて確認します。



リーダー権限は、**スキーム・タイプがブールを返すPL/SQLファンクション**になっています。他にも選択可能なスキーム・タイプはありますが、還元すると**認可スキームは、真偽値を返すPL/SQLのファンクションです**。

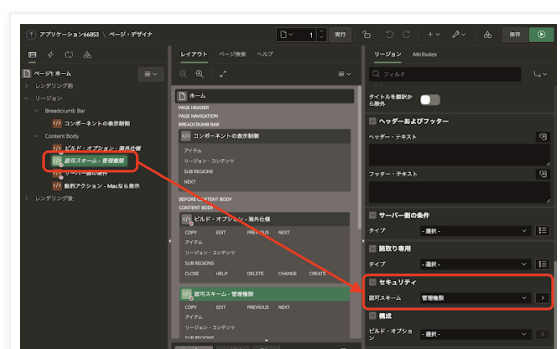


認可スキームで重要なのは、**評価ポイント**です。ビルド・オプションはステータスを含める、もしくは、除外にすると、その時点でアプリケーション全体の構成が変わります。認可スキームは、以下の4種類の評価ポイントがあり、その都度、評価されます。

- セッションごとに1回
- ページ・ビューごとに1回
- コンポーネントごとに1回
- 常時(キャッシュなし)

検証結果はキャッシュされるため（次に説明するサーバー側の条件に比べて）、パフォーマンス面で有利です。コンポーネントを有効にするか無効にするかを評価する際に、セッション・ステートに依存することが無ければ、認可スキームの利用を検討すべきです。

認可スキームの設定も、ほぼ全てのコンポーネントに含まれます。プロパティの**セキュリティ**に含まれる**認可スキーム**です。



認可スキームは、上記のコンポーネントへの設定以外にも、対話グリッドやフォームが行う操作の認可にも使われます。むしろ、こちらの使い方が本来の用途です。

サーバー側の条件

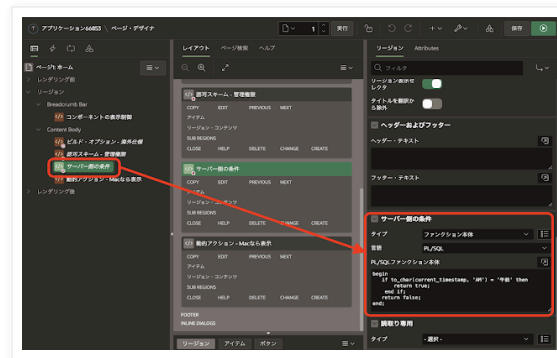
それぞれのコンポーネントを有効にするか無効にするか、そのコンポーネントが呼び出される際に評価する条件を設定できます。**サーバー側の条件も還元すると真偽値を返すPL/SQLファンクションです**。例えば、次のようなPL/SQLファンクション本体を設定すると、午前中だけリージョンが表示されます。

begin

```

if to_char(current_timestamp, 'AM') = '午前' then
    return true;
end if;
return false;
end;

```

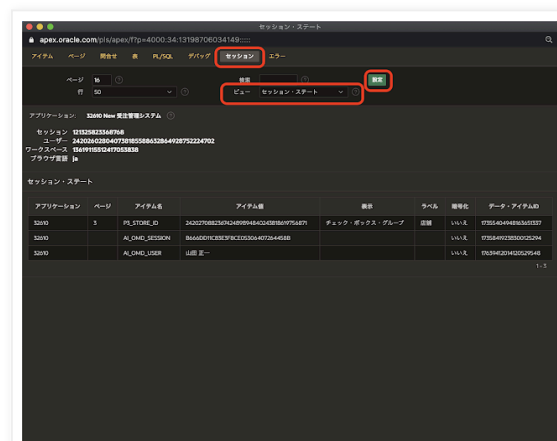


認可スキームとは異なり、コンポーネントが呼び出される度に評価されるため、その評価に**セッション・ステート**を使うことができます。セッション・ステートとは、APEXのセッションに紐づけられてデータベースに保存されている、アプリケーション・アイテムとページ・アイテムの値です。

セッション・ステートとして保持されている値は、**開発者ツール・バー**にある**セッション**をクリックし、表示されるデバッグ画面より、**ビュー**として**セッション・ステート**を選択し、**設定**をクリックすることで確認できます。

受注ID	受注名	受注日	受注額	受注ステータス
1000000001	東京支店 受注	2023/01/01	100,000	完了
1000000002	大阪支店 受注	2023/01/02	200,000	完了
1000000003	名古屋支店 受注	2023/01/03	300,000	完了
1000000004	福岡支店 受注	2023/01/04	400,000	完了
1000000005	札幌支店 受注	2023/01/05	500,000	完了
1000000006	仙台支店 受注	2023/01/06	600,000	完了
1000000007	東京支店 受注	2023/01/07	700,000	完了
1000000008	大阪支店 受注	2023/01/08	800,000	完了
1000000009	名古屋支店 受注	2023/01/09	900,000	完了
1000000010	福岡支店 受注	2023/01/10	1,000,000	完了

以下はデバッグ画面です。



ちなみに、Oracle APEXのアプリケーションは、セッションに紐づくデータをセッション・ステートとしてデータベースに保存（つまりディスクに保存）しているか、一時的な値であればブラウザ側で保持した状態になっていて、それ以外には一時的であってもデータは保持していません。そのため、**データベースを再起動しても、Oracle APEXのセッションは継続します。**

サーバー側の条件は一番融通のきく設定なので、ビルド・オプションや認可スキームが適切であっても、つついサーバー側の条件を使いがちです。サーバー側の条件を設定する前に、ビルド・オプションや認可スキームが向いていないか、検討してみましょう。

動的アクション

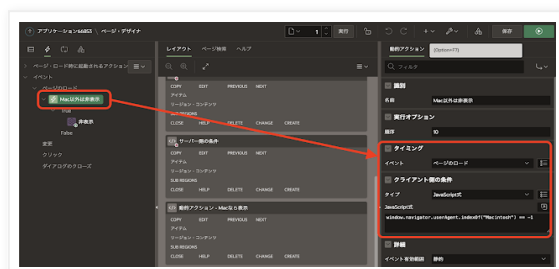
動的アクションによっても、コンポーネントの表示/非表示を制御できます。今までの設定は、評価でfalseが返ってくると、(表示に関連するコンポーネントであれば) **HTML自体が生成されません**でした。

動的アクションの場合は、**HTML自体は生成されている**ことが前提です。この違いはセキュリティの観点では重要です。ブラウザの開発者ツールを使って参照されると困るのであればサーバー側の設定などを使用し、動的アクションによる制御は使わないようにしましょう。

例えば、ページのロード時に、ブラウザがMac以外で動作している場合はリージョンを非表示にする動的アクションは以下のように定義します。

タイミングのイベントをページのロードとしています。**クライアント側の条件**として、**タイプ**に**JavaScript式**を選択し、以下の式を設定しています。

```
window.navigator.userAgent.indexOf("Macintosh") == -1
```



HTTPヘッダーのUser-AgentにMacintoshが含まれていなければ、Trueアクションとして登録されたアクションを実行します。上記では、**非表示**がTrueアクションです。

アクションとして非表示を選択し、**選択タイプ**を**リージョン**、非表示にするリージョンは**動的アクション - Macなら表示**を設定しています。



リージョンの表示・非表示に関していうと、非表示のアクションは対象となるHTML要素に `style="display: none;"`を追加しています。表示のアクションは、その設定を取り除いています。

最初の画面にあるボタンのクリックにより、ビルド・オプション、認可スキーム、サーバー側の条件、動的アクションそれぞれでリージョンの表示・非表示の動きを確認するアプリケーションのエクスポートを以下に置きました。(説明に使っている例はアプリケーションには含んでいません)。

<https://github.com/ujnak/apexapps/blob/master/exports/component-controls.sql>

Oracle APEXのアプリケーション開発の一助になれば幸いです。

完

Yuji N. 時刻: 16:51

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](#)

Powered by Blogger.