

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6月4日 金曜日

ワクチン接種状況ダッシュボードに機能を追加する - 追記

先日に作成したワクチン接種状況ダッシュボードを、もう少し改変した際に使った設定などを追記します。

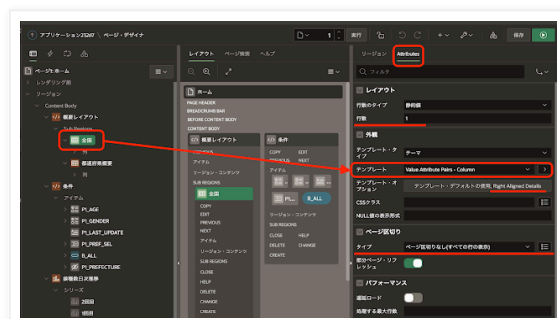
クラシック・レポートのテンプレート変更

Oracle APEXが提供しているレポート向けのリージョン・タイプは3つあります。クラシック・レポート、対話モード・レポート、対話グリッドです。この中で一番古くから提供されているのがクラシック・レポートです。対話的にレポートの操作をするといった高度な機能は含まれていませんが、テンプレートを変更することによって、レポートではない表示形式に変えることができます。

概要を表示しているレポートの表示を以下に変更しました。

全国	
人口	127,128,905
接種数	6,823,545
1回目接種数	6,266,076
1回目接種率	4.93%
2回目接種数	557,469
2回目接種率	0.44%

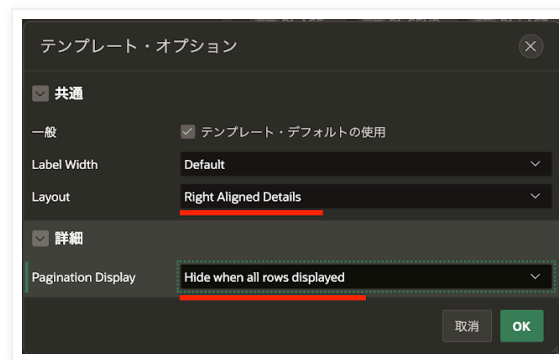
クラシック・レポートのリージョンのAttributesを開き、外観のテンプレートとしてValue Attribute Pair - Columnを選択します。テンプレート・オプションのLayoutにRight Aligned Detailsを選択することで、数値を右に揃えています。



その他、レイアウトの行数のタイプを静的値とし、行数に1を指定しています。今回のレポートの表示結果はつねに1行なので、1回のSELECT文の実行でフェッチする行数を1行に制限することは、パフォーマンス上のメリットがあります。

ページ区切りのタイプはページ区切りなし(すべての行の表示)にしています。テンプレート・オプションにも同様の設定があります。

詳細の**Pagination Display**を、**Hide when all rows displayed**に設定すると、必要な時だけページ区切りが表示されます。今回は最初からページ区切りは表示していないので、**Default**から変更していません。



都道府県のセレクトの追加

都道府県のセレクトを追加しました。セレクトを使って都道府県を切り替えたときに、動的アクションとしてJavaScriptを実行し、ページ・アイテムP1_PREFECTUREへの番号の設定と、概要を表示するリージョンのタイトルの設定を行なっています。



Trueアクションとして記載したJavaScriptは以下です。

```
let sel = $f_SelectedOptions("P1_PREF_SEL");
$("#P1_PREFECTURE", sel.value);
document.getElementById("pref_specific_heading").textContent = sel.text;
```

選択されている要素を取得するために**\$f_SelectedOptions**というOracle APEXが提供しているJavaScript APIを使用しています。

例えば、DOMを使って都道府県名を取得するには
`document.getElementById("P1_PREF_SEL").options[document.getElementById("P1_PREF_SEL").selectedIndex].text`

jQueryを使って選択された都道府県名を取得するには
`$f_SelectedOptions("P1_PREF_SEL").text()`

といった呼び出しになります。Oracle APEXが前提のコーディングですので、Oracle APEXが提供しているAPIの使用が推奨です。jQueryは便利ですが、一般的なトレンドとして使われない方向になっているので、これからの使用は避けた方がよいでしょう。

Oracle APEXがというよりjQuery自体の話なので、それぞれで判断する必要があります。

動的アクション内での値の設定

上記で都道府県のセレクトを追加しました。都道府県の選択は、新たに追加したセレクトと、対話グリッドの2つになっています。この2つの選択結果を同期させるため、対話グリッドで都道府県を選択したときは、都道府県のセレクトに選択結果を設定するだけにします。

元々のコードは以下でした。(最初の記事ではPREFCODEだった列名はPREFECTUREに変更されています。)

```
let model = this.data.model;
let record = this.data.selectedRecords[0];
let prefecture = model.getValue(record, "PREFECTURE");
$('P1_PREFECTURE', prefecture);
document.getElementById("pref_specific_heading").textContent
    = model.getValue(record, "PREFECTURE_NAME");
```

次の様に変更しています。

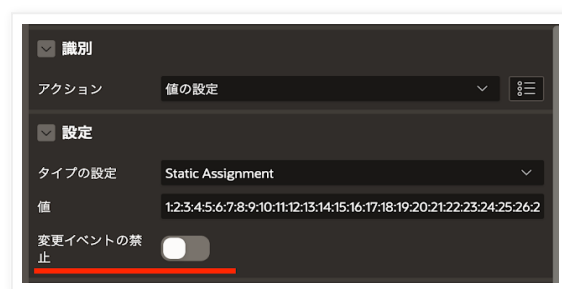
```
let model = this.data.model;
let record = this.data.selectedRecords[0];
let prefecture = model.getValue(record, "PREFECTURE");
$('P1_PREF_SEL', prefecture);
```

ページ・アイテムP1_PREF_SELの値をOracle APEXのJavaScript API **\$s**を使って変更すると、**変更のイベント**で実行される動的アクションが起動されます。ページ・アイテムP1_PREFECTUREへの値の設定と、リージョンのタイトルpref_sepecific_headingへの文字列の設定は、ページ・アイテムP1_PREF_SELに設定された動的アクションによって実行されます。

デフォルトでは\$sによって発生する変更イベントは許可されます。禁止する場合は第4引数にtrueを渡します。

```
$('P1_PREF_SEL', prefecture, null, true);
```

これは、動的アクションの**アクション**として**値の設定**を選んだときの、**設定の変更イベントの禁止**と同じ指定になります。



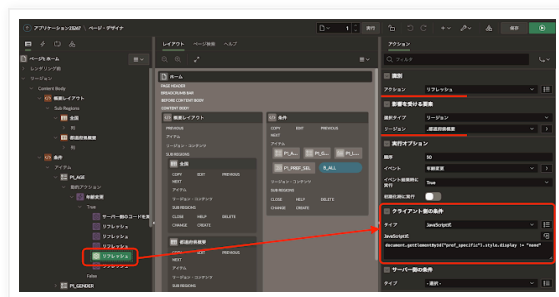
リフレッシュの抑制

非表示になっているレポートをリフレッシュする必要はありません。動的アクションのTrueアクションに**クライアント側の条件**を設定することにより、リフレッシュの実行を抑制することが可能です。

リフレッシュを行うアクションの**クライアント側の条件のタイプ**に**JavaScript式**を選択し、以下のような**JavaScript式**を記述します。

```
document.getElementById("pref_specific").style.display != "none"
```

pref_specificはリフレッシュの対象としているリージョンの静的IDです。属性のstyleに含まれているdisplayがnoneでないとき、つまり、リージョンが表示されているときのみリフレッシュを実行します。



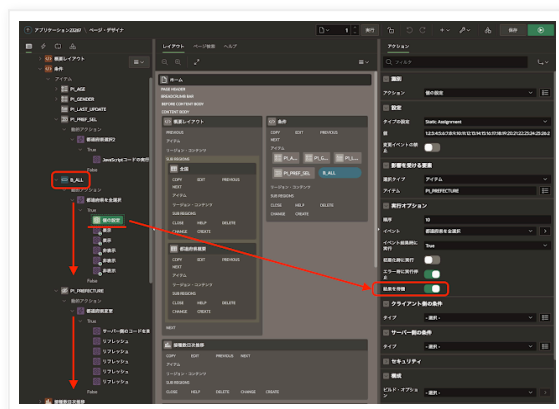
アクションにクライアント側の条件が設定できるようになったのは、Oracle APEX 21.1からです。

ページ・アイテムP1_GENDERおよびP1_AGEが変更されたときに実行されるリフレッシュに指定します。P1_PREFECTUREには指定しません。

動的アクションは非同期で実行される

全国から都道府県が表示が変わるとき、または、都道府県から全国に表示が変わるときには、いくつかのリージョンが表示から非表示、非表示から表示に切り替わります。

ページ・アイテムP1_PREFECTUREの変更によって呼び出されるリージョンのリフレッシュと、表示の切り替えを行う動的アクションは、それぞれ非同期で実行されるため、どちらが先に実行されるかは判断できません。そのため、P1_PREFECTUREの変更で呼び出されるリフレッシュには**クライアント側の条件**は付けません。



ボタンB_ALLをクリックすると最初にページ・アイテムP1_PREFECTUREに値を設定します。そのアクションの**実行オプション**として**結果を待機**がONになっています。

これはP1_PREFECTUREへ値が設定されるまで待機する、という動作です。P1_PREFECTUREに設定された動的アクションの実行が完了するまで待機する、という意味ではありません。そのため、リージョンの表示/非表示の切り替えとリージョンのリフレッシュのどちらが先に実行が完了するかは判断できない、ということになります。

処理の順番を決めたい場合は別々のアクションではなく、アクションをひとつのJavaScriptコードの実行として、その中で順番にリージョンの表示/非表示を切り替えるコードおよびリフレッシュを呼び出すコードを記載する必要があります。

リージョン表示セレクト

Oracle APEXは標準で、**リージョン表示セクタ**というリージョンを提供しています。

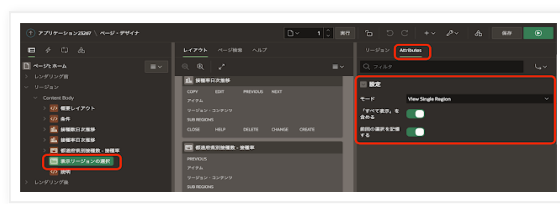


ページに存在する**リージョン**のプロパティに含まれる**詳細のリージョン表示セクタ**をONにすると、リージョン表示セクタでの操作対象になります。



すべて表示、または、リージョン表示セクタのプロパティがONのリージョンを選択して表示することができます。

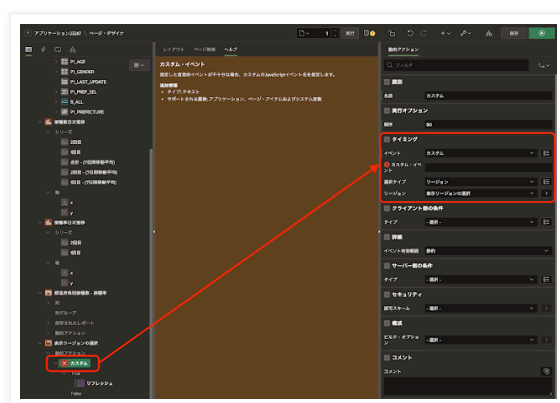
リージョン表示セクタの**Attributes**より、いくつかのオプションを設定できます。**モード**や「**すべての表示**」を含めるなどの切り替えが含まれます。



今回の設定ではリージョンが非表示のときは、リフレッシュを行わない設定にしています。そのため、個別のリージョン、例えば、**接種数日次推移**を選択している状態から**すべて表示**に切り替えたときは、**接種率日次推移**と**都道府県別接種数 - 接種率**は(年代と性別の選択をリージョンが非表示の間に切り替えていると)、非表示なる直前の結果が表示されます。

リフレッシュへのクライアント側の条件を外せば、リージョンが非表示の間もリフレッシュが行われるため、このようなことは発生しません。

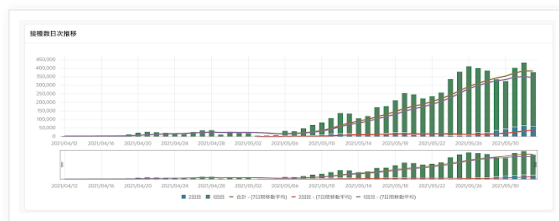
リージョン表示セクタの**変更**となる動的アクションの**タイミング**は定義されていません。**タイミング**には**カスタム・イベント**も定義できます。



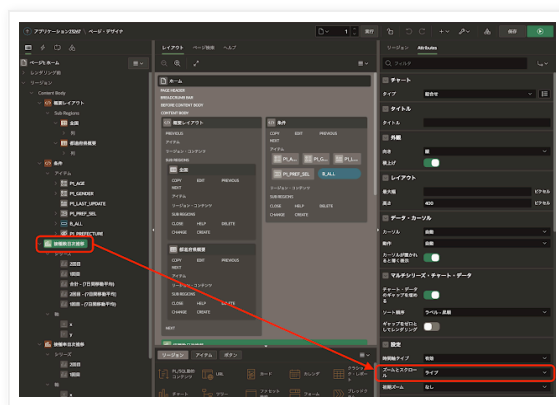
そこまでの設定は行なっていません。また、このような用途であれば、リージョン表示セレクトではなく、ボタンやチェックボックス・グループを使って、表示するリージョンを選択する機能を実装した方が簡単でしょう。

チャートのズーム

接種数日次推移のチャートにズームとスクロールの機能を追加しました。



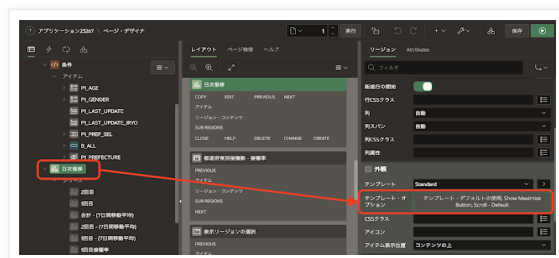
チャートの**Attributes**の**設定**に、**ズームとスクロール**というプロパティがあります。これに**ライブ**を指定しています。



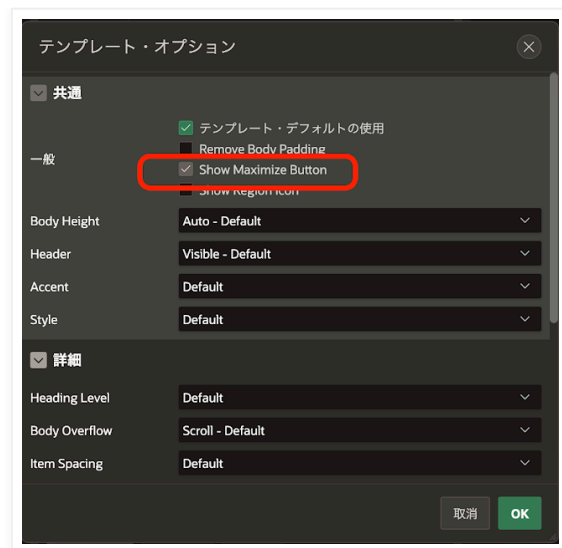
チャートの全画面表示

チャートに限りませんが、**テンプレート・オプション**の**Show Maximize Button**にチェックを入れると、チャートを全画面で表示できるようになります。

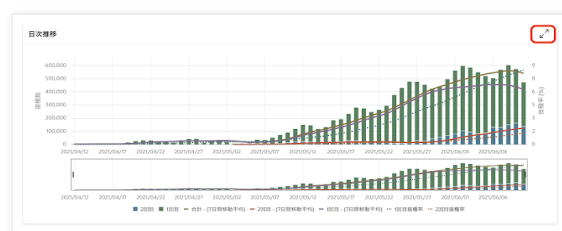
外観のテンプレート・オプションを開きます。



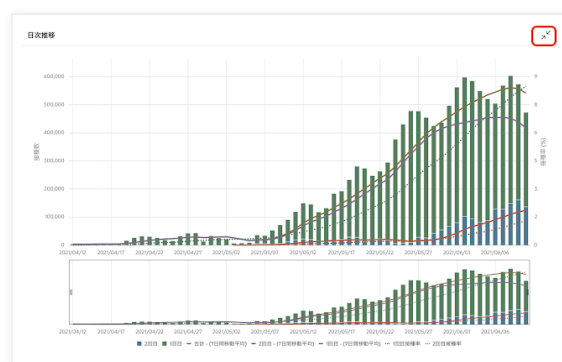
共通の一般に含まれます。



リージョンの最大化ボタンは、右上端に表示されます。

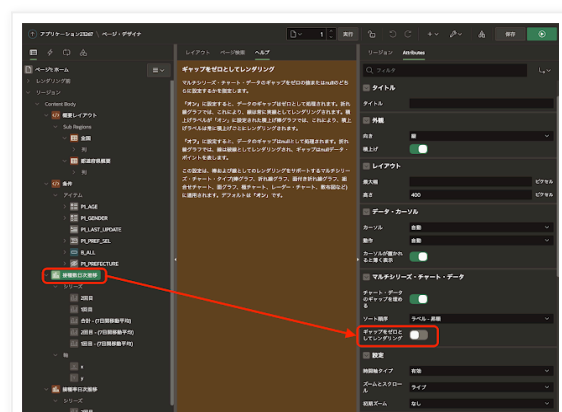


最大化すると、元に戻るボタンに切り替わります。

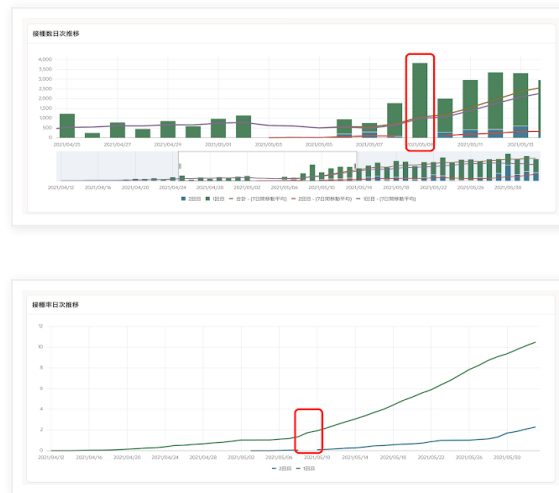


データに抜けがあるときの対応

チャートのAttributesに含まれる、マルチシリーズ・チャート・データのギャップをゼロとしてレンダリングをOFFにしました。



結果として、チャートが途中で消えることがあります。以下は和歌山県のデータです。



原因は5月9日の2回目の接種が0回で、その日のデータ自体が存在しないためです。今回はチャートに複数のシリーズを作成しているので、マルチシリーズ・チャートになります。

選択肢としては、その日のデータを0として扱うか、nullとして扱うかになります。nullとして扱う様にしたので、チャートが途切れることになってしまいました。しかし、0として扱うよりは良いです。

接種率についてはチャートがつながれば良いだけですが、移動平均を正確に算出するには、データ自体にその日の件数が0であることを示す行が必要です。

今後、2回目の接種が行われない日はそれほどないと予想されるため、特に対応はしていません。

完

Yuji N. 時刻: 14:11

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.