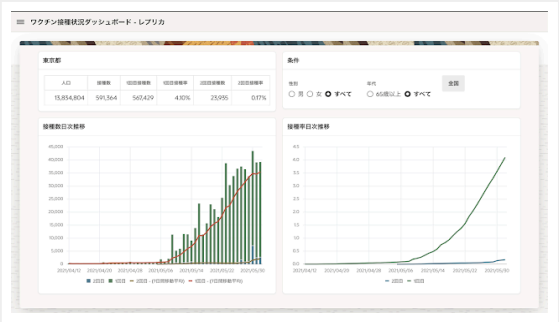
日日是Oracle APEX

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。

2021年6月3日木曜日

ワクチン接種状況ダッシュボードに機能を追加する

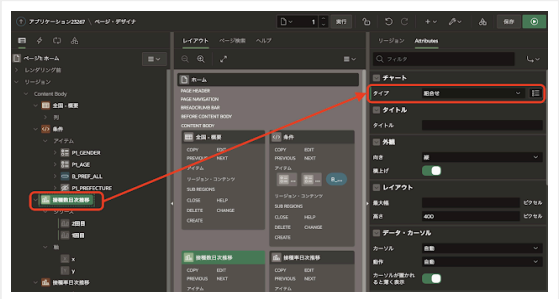
前回作成してみたワクチン接種状況ダッシュボードですが、同じものを作るだけであれば、あまり意味がありません。せっかく手元にデータを持ってきたのですから、少々活用してみました。



ワクチン接種数の７日間移動平均をグラフに追加する

接種数日次推移のグラフに７日間の接種数の移動平均を表示させてみます。

リージョンの**接種数日次推移**を選択し、**Attributes**の**チャート**の**タイプ**を棒から**組合せ**に変更します。移動平均は棒ではなく、折れ線で表示させます。



シリーズの作成を行います。**名前**を**2回目 - (7日間移動平均)**とします。**タイプ**は**折れ線**を選択します。**ソース**の**タイプ**は**SQL問合せ**、**SQL問合せ**として以下を記述します。

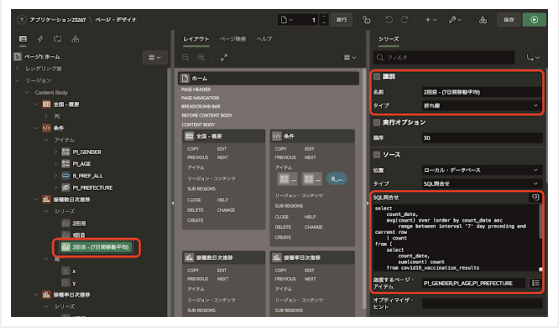
```
select
    count_date,
    avg(count) over (order by count_date asc
        range between interval '6' day preceding and current row
    ) count
from (
    select
        count_date,
        sum(count) count
    from covid19_vaccination_results
```

```
    where status = 2
        and gender in (select column_value from apex_string.split(:P1_GENDER, ':'))
        and age in (select column_value from apex_string.split(:P1_AGE, ':'))
        and prefecture in (select column_value from apex_string.split(:P1_PREFECTURE, ':'))
    group by count_date
)
```

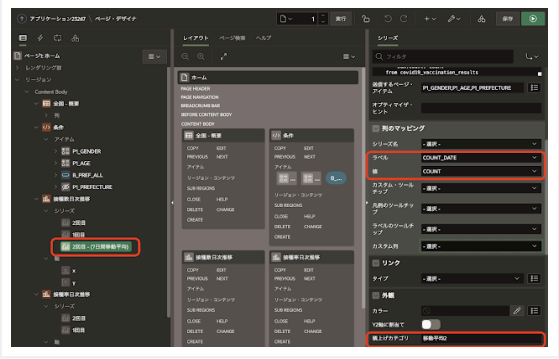
元々の接種数の検索結果にSQLのWINDOW関数を適用することで、移動平均を算出しています。

SQLのWINDOW関数の結果をすぐにチャートに表示できるのは便利、と感じるか、SQLで書かなければいけないのか？と感じるかで、Oracle APEXが使えるツールと感じるかどうかの分かれ目になると思います。SQL自体はOracle APEXを使うかどうか、さらにはOracle Databaseを使うかどうかにも関係なく役立てることができる知識なので、難しくても学ぶ価値があります。

送信するページ・アイテムとして**P1_GENDER、P1_AGE、P1_PREFECTURE**を指定します。



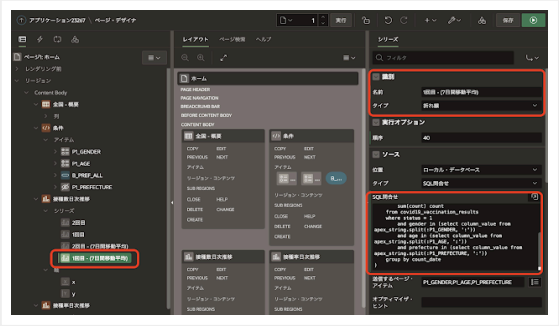
列のマッピングとして、**ラベル**に**COUNT_DATE**、**値**に**COUNT**を指定します。**外観**の**積上げカテゴリ**に**移動平均2**を入力します。移動平均は積み上げ対象から外したいので、単独となる積上げカテゴリにしています。



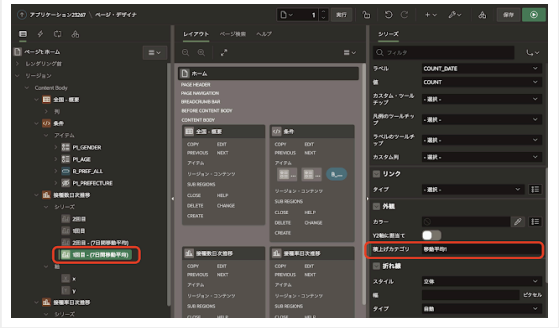
作成したシリーズ**2回目 - (7日間移動平均)**を**重複**させます。**名前**は**1回目 - (7日間移動平均)**に変更します。**SQL問合せ**を以下に変更します。where status = 2の部分をwhere status = 1に変更しています。

```
select
    count_date,
    avg(count) over (order by count_date asc
        range between interval '6' day preceding and current row
    ) count
from (
    select
        count_date,
        sum(count) count
    from covid19_vaccination_results
    where status = 1
```

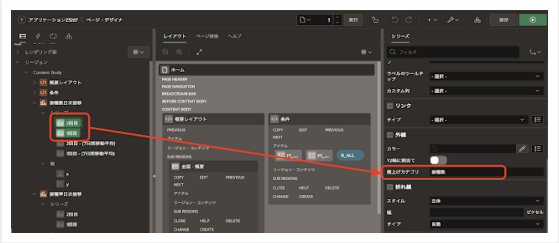
```
        and gender in (select column_value from apex_string.split(:P1_GENDER, ':'))
        and age in (select column_value from apex_string.split(:P1_AGE, ':'))
        and prefecture in (select column_value from apex_string.split(:P1_PREFECTURE, ':'))
    group by count_date
)
```



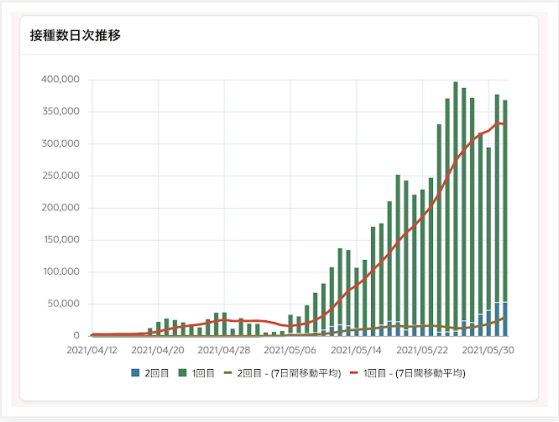
積上げカテゴリは**移動平均1**に変更します。



元々あったシリーズ**1回目**と**2回目**を選択し、**積上げカテゴリ**を**接種数**とします。日次の接種数は積み上げの対象になります。



ページの保存と実行を行うと、接種数日次推移のチャートに、7日間の移動平均を表す折れ線グラフが追加されていることが確認できます。

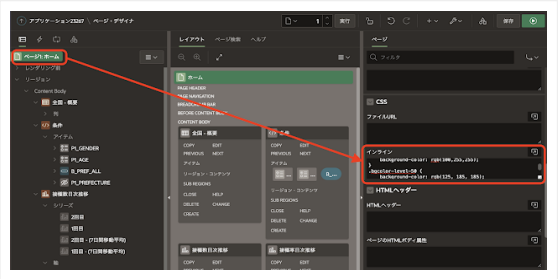


カラムの値に依存してセルの背景色を変更する

元のダッシュボードは接種率に応じて連続的に背景色を変更しています。Oracle APEXで、それを簡単に実装する方法は見つけることはできませんでした。

その代わりに、段階的に背景色を変更してみます。それぞれの背景色を設定するCSSクラスを定義します。これは**ページ**のプロパティの**CSS**、**インライン**に記載します。CSSクラスは全部で11あります。

```
.bgdolor-level-100 {
    background-color: rgb(0,255,255);
}
.bgcolor-level-90 {
    background-color: rgb(25,255,255);
}
.bgcolor-level-80 {
    background-color: rgb(50,255,255);
}
.bgcolor-level-70 {
    background-color: rgb(75,255,255);
}
.bgcolor-level-60 {
    background-color: rgb(100,255,255);
}
.bgcolor-level-50 {
    background-color: rgb(125, 185, 185);
}
.bgcolor-level-40 {
    background-color: rgb(150,255,255);
}
.bgcolor-level-30 {
    background-color: rgb(175,255,255);
}
.bgcolor-level-20 {
    background-color: rgb(200,255,255);
}
.bgcolor-level-10 {
    background-color: rgb(225,255,255);
}
.bgcolor-level-00 {
    background-color: rgb(255,255,255);
}
```



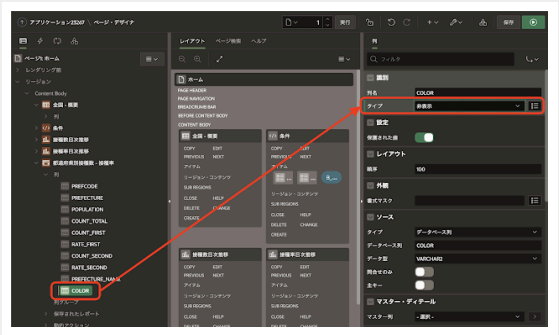
対話グリッドのリージョン**都道府県別接種数 - 接種率**を選択し、**SQL問合せ**を以下に変更します。検索結果の列として**COLOR**を追加し、接種率が0, 10, 20, ... 100までの11段階に分けてCSSクラスを返すようにします。

```
with v_first as (
    select
```

```
        prefecture,
        sum(count) count
    from covid19_vaccination_results
    where status = 1
        and gender in (select column_value from apex_string.split(:P1_GENDER, ':'))
        and age in (select column_value from apex_string.split(:P1_AGE, ':'))
        and prefecture in (select column_value from apex_string.split(:P1_PREFECTURE, ':'))
    group by prefecture
),
v_second as (
    select
        prefecture,
        sum(count) count
    from covid19_vaccination_results
    where status = 2
        and gender in (select column_value from apex_string.split(:P1_GENDER, ':'))
        and age in (select column_value from apex_string.split(:P1_AGE, ':'))
        and prefecture in (select column_value from apex_string.split(:P1_PREFECTURE, ':'))
    group by prefecture
),
v_total as (
    select
        prefecture,
        prefecture_name,
        sum(count) total
    from covid19_vaccination_targets
    where 1=1
        and gender in (select column_value from apex_string.split(:P1_GENDER, ':'))
        and age in (select column_value from apex_string.split(:P1_AGE, ':'))
        and prefecture in (select column_value from apex_string.split(:P1_PREFECTURE, ':'))
    group by prefecture, prefecture_name
)
select
    f.prefecture prefecture,
    t.prefecture_name prefecture_name,
    t.total population,
    (f.count + s.count) count_total,
    f.count count_first,
    (f.count / t.total) * 100 rate_first,
    s.count count_second,
    (s.count / t.total) * 100 rate_second,
    case
    when (f.count / t.total) * 100 >= 100 then
        'bgcolor-level-100'
    when (f.count / t.total) * 100 >= 90 then
        'bgcolor-level-90'
    when (f.count / t.total) * 100 >= 80 then
        'bgcolor-level-80'
    when (f.count / t.total) * 100 >= 70 then
        'bgcolor-level-70'
    when (f.count / t.total) * 100 >= 60 then
        'bgcolor-level-60'
    when (f.count / t.total) * 100 >= 50 then
        'bgcolor-level-50'
    when (f.count / t.total) * 100 >= 40 then
        'bgcolor-level-40'
    when (f.count / t.total) * 100 >= 30 then
        'bgcolor-level-30'
```

```
    when (f.count / t.total) * 100 >= 20 then
        'bgcolor-level-20'
    when (f.count / t.total) * 100 >= 10 then
        'bgcolor-level-10'
    else
        'bgcolor-level-00'
    end color
from v_first f
    join v_second s on f.prefecture = s.prefecture
    join v_total t  on f.prefecture = t.prefecture
```

列**COLOR**が対話グリッドに認識されるので、この列の**タイプ**を**非表示**に変更します。

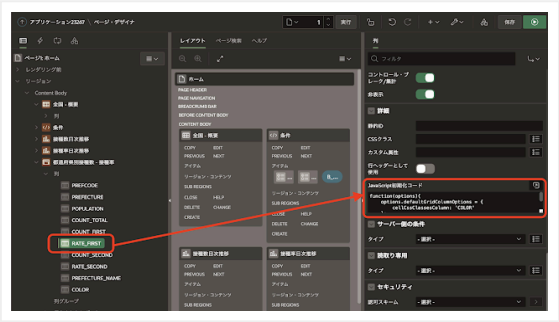


列COLORとして返される値を、列RATE_FIRSTにCSSクラスとして適用します。

列**RATE_FIRST**を選択し、**詳細**の**JavaScript初期化コード**として以下を記述します。

```
function(options){
    options.defaultGridColumnOptions = {
        cellCssClassesColumn: 'COLOR'
    };
    return options;
}
```

対話グリッドの列のプロパティcellCssClassesColumnとして**COLOR**を設定しています。これで列COLORの値が、CSSクラスとして列RATE_FIRSTに適用されます。

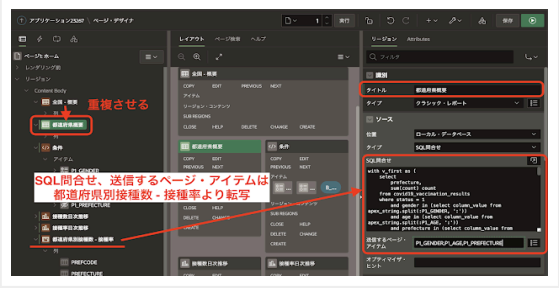


ページの保存と実行を行うと、1回目接種率のセルの色が異なって表示される都道府県があることが確認できます。

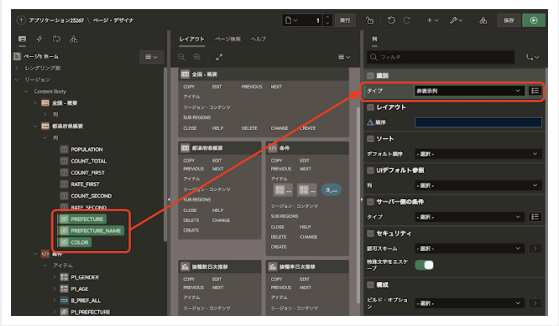
都道府県の概要を表示する

都道府県の選択時に全国の概要ではなく、都道府県の概要を代わりに表示させます。

リージョン**全国 - 概要**を**重複**させ、**名前**を**都道府県概要**に変更します。**ソース**の**SQL問合せ**、**送信するページ・アイテム**はリージョン**都道府県別接種数 - 接種率**より**コピー**します。

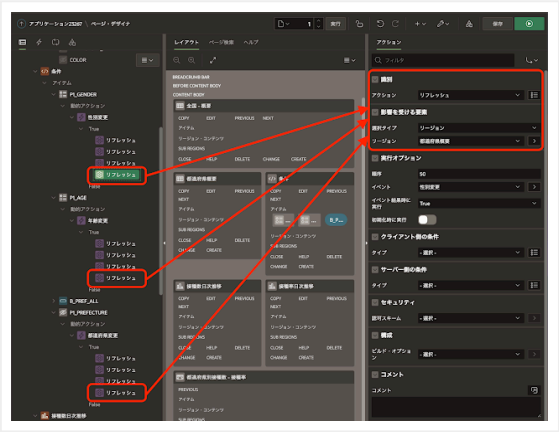


追加でレポートに認識された列**PREFECTURE**、**PREFECTURE_NAME**、**COLOR**の**タイプ**を**非表示列**にします。



追加したリージョン**都道府県概要**が、検索条件の変更時にリフレッシュされるよう**Trueアクションの作成**をします。

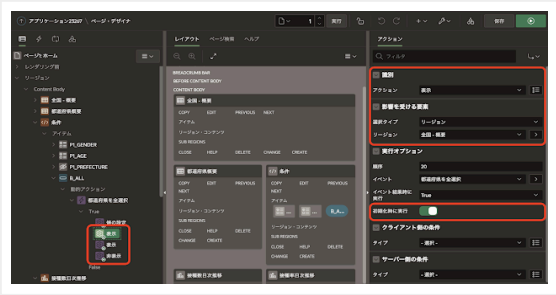
ページ・アイテム**P1_GENDER**、**P1_AGE**、**P1_PREFECTURE**それぞれに、**アクション**が**リフレッシュ**のTrueアクションを作成します。**影響を受ける要素**の**選択タイプ**として**リージョン**、**リージョン**として**都道府県概要**を指定します。



全国が表示対象のときは**全国 - 概要**のリージョンだけを表示し、都道府県が選択されているときは**都道府県概要**のリージョンだけを表示するように設定します。

全国ボタンをクリックしたときに実行される動的アクション**都道府県を全選択**に**Trueアクションを３つ追加**します。

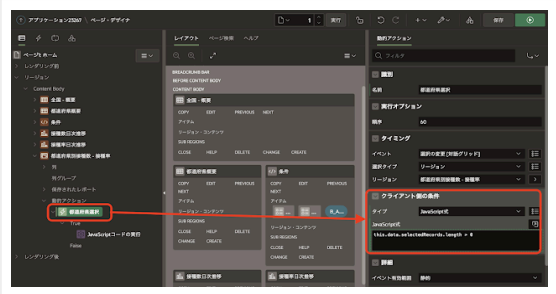
アクションは**表示**、**リージョン**は**全国 - 概要**、**初期化時に実行**は**ON**。
アクションは**表示**、**リージョン**は**都道府県別接種数 - 接種率**、**初期化時に実行**は**ON**。
アクションは**非表示**、**リージョン**は**都道府県概要**、**初期化時に実行**は**ON**。



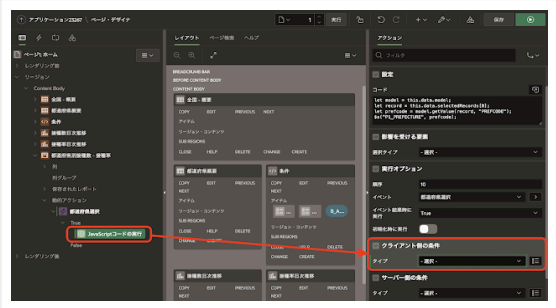
次に対話グリッドで都道府県の選択を行ったときに実行される動的アクションを追加します。

最初に、動的アクション**都道府県選択**の**クライアント側の条件**の**タイプ**を**JavaScript式**として、以下の**JavaScript式を**設定します。

```
this.data.selectedRecords.length > 0
```

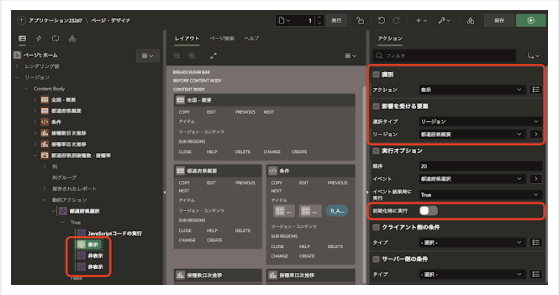


Trueアクションの**JavaScriptコードの実行**から、**クライアント側の条件**を外します。動的アクション自体に**クライアント側の条件**がついているので、Trueアクションには不要になりました。

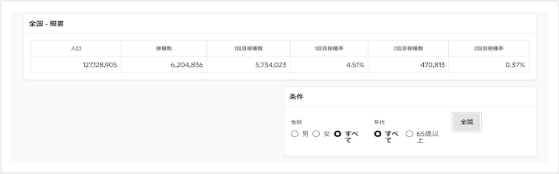


全国のボタンをクリックしたときとは反対に、**都道府県概要**のリージョンだけが表示されるよう、**Trueアクションを3つ作成**します。

アクションは**表示**、**リージョン**は**都道府県概要**、**初期化時に実行**は**OFF**。
アクションは**非表示**、**リージョン**は**全国 - 概要**、**初期化時に実行**は**OFF**。
アクションは**非表示**、**リージョン**は**都道府県別接種数 - 接種率**、**初期化時に実行**は**OFF**。



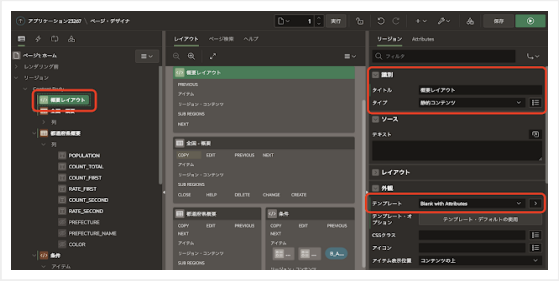
この状態でページを実行してみます。全国の表示は以下のようになります。



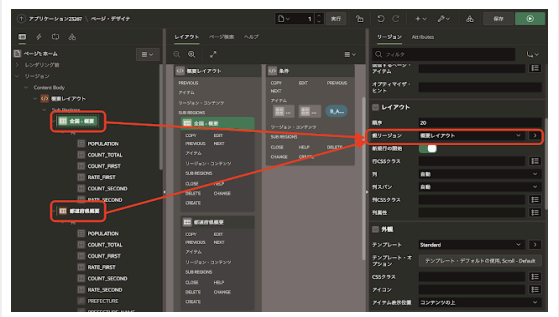
都道府県を選択すると、以下のようになります。

レイアウトが崩れない様に、リージョンを追加し、**全国 - 概要**と**都道府県概要**のリージョンを作成したリージョンのサブ・リージョンにします。

リージョンの作成を行い、Content Bodyの先頭に配置します。**識別**の**タイトル**を**概要レイアウト**とし、**タイプ**に**静的コンテンツ**を選択します。**外観**の**テンプレート**として**Blank with Attributes**を選択します。



リージョン**全国 - 概要**および**都道府県概要**の**レイアウト**の**親リージョン**を**概要レイアウト**に変更します。

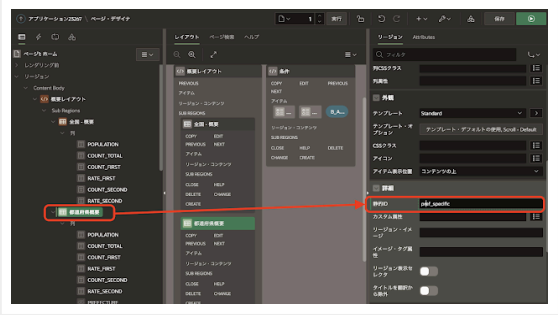


以上で全国と都道府県で選択を切り替えても、ページのレイアウトが保持されます。

都道府県概要のタイトル表示を都道府県名にする

都道府県を選択したときのレポートのタイトルが都道府県概要となっており、どの都道府県かわかりません。ここに都道府県名を表示させます。

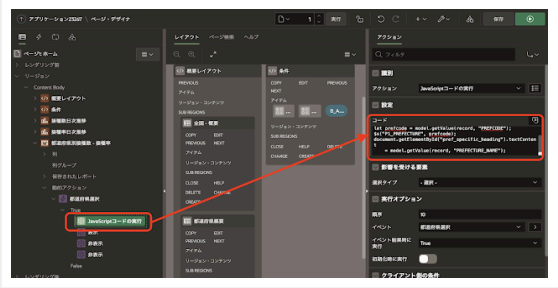
都道府県概要のリージョンに静的IDを設定します。**静的ID**は**pref_specific**とします。



都道府県選択を行なったときに実行されるTrueアクションである、**JavaScriptコードの実行**の**コード**を以下に変更します。末尾に1行追記しリージョンのタイトルに都道府県名を設定しています。

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

idが**pref_sepecific_heading**であるHTML要素のテキスト部分に、**都道府県名**を代入しています。



今回の作業は以上になります。今までで作業は、公開済みのアプリケーションのエクスポートに含まれています。

作成したアプリケーションは以下よりアクセスできます。
https://apex.oracle.com/pls/apex/japancommunity/r/vaccine-dashboard/home

Oracle APEXのアプリケーションの作成の参考になれば幸いです。

完

Yuji N. 時刻: 16:22

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