

日々是Oracle APEX

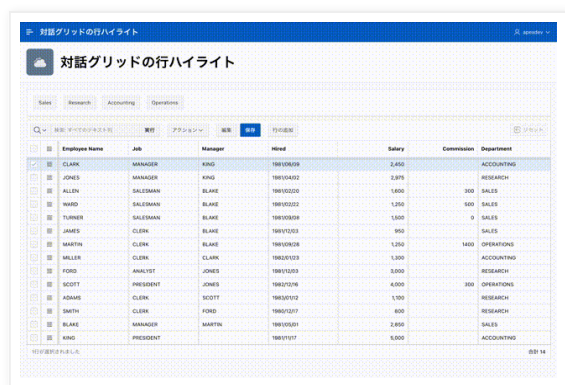
Oracle APEXを使った作業をしていて、気の付いたところを忘れないようにメモをとります。

2023年7月27日 木曜日

外部からの操作で対話グリッドの行をハイライトさせる

対話グリッド自体が持つハイライトの設定ではなく、外部からの操作で対話グリッドの行をハイライトさせる方法について調べました。

サンプルの実装では、ボタンにマウスポインタが乗ったときに、ボタンの静的IDに一致している部門の行をハイライトしています。ポインタが外れたときにハイライトを解除します。



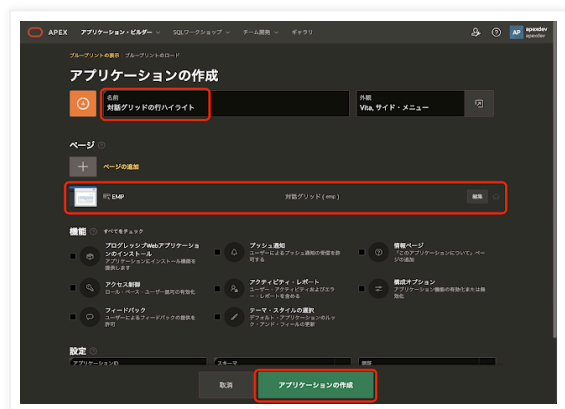
Employee Name	Job	Manager	Minid	Salary	Commission	Department
CLARK	MANAGER	KING	19810809	2,450		ACCOUNTING
JAMES	MANAGER	KING	19810809	2,875		RESEARCH
ALLEN	SALESMAN	BLAKE	19810809	1,600	300	SALES
WARD	SALESMAN	BLAKE	19810809	1,250	300	SALES
TURNER	SALESMAN	BLAKE	19810809	1,500	0	SALES
JAMES	CLERK	BLAKE	19810809	950		SALES
SMITH	CLERK	BLAKE	19810809	1,200	1400	OPERATIONS
MILLER	CLERK	CLARK	19810809	1,300		ACCOUNTING
FORD	ANALYST	JONES	19810809	3,000		RESEARCH
SCOTT	PRESIDENT	JONES	19810809	4,000	300	OPERATIONS
ADAMS	CLERK	SCOTT	19810809	1,100		RESEARCH
SMITH	CLERK	FORD	19810809	800		RESEARCH
BLAKE	MANAGER	MARTIN	19810809	2,800		SALES
KING	PRESIDENT		19810809	5,000		ACCOUNTING

以下、実装方法の紹介です。

サンプル・データセットのEMP/DEPTを英語でインストールして使用します。

アプリケーション作成ウィザードを起動します。名前は対話グリッドの行ハイライトとします。

デフォルトで作成されているホーム・ページを削除し、対話グリッドのページを追加します。



対話グリッドのページの追加では、編集を許可を選択し、表またはビューにEMPを指定します。

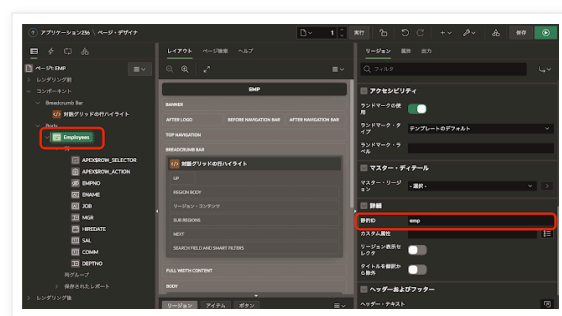


以上でアプリケーションを作成します。

アプリケーションが作成されたら、**ページ・デザイナー**で対話グリッドのページを開きます。

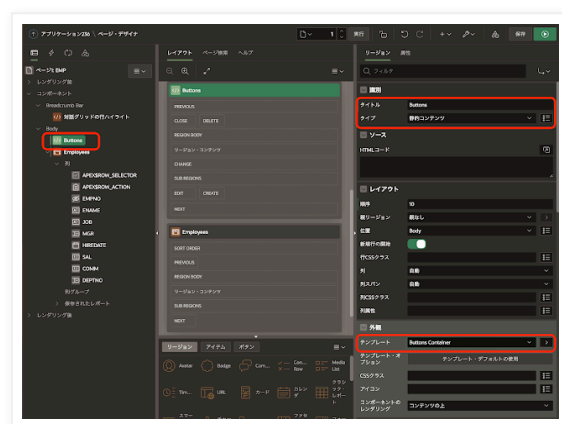
対話グリッドをJavaScriptから操作するために、対話グリッドに静的IDを設定します。

詳細の静的IDに**emp**を設定します。



ボタンを配置するリージョンを作成します。

識別のタイトルは**Buttons**、タイプは静的コンテンツです。外観のテンプレートとして、**Buttons Container**を選択します。



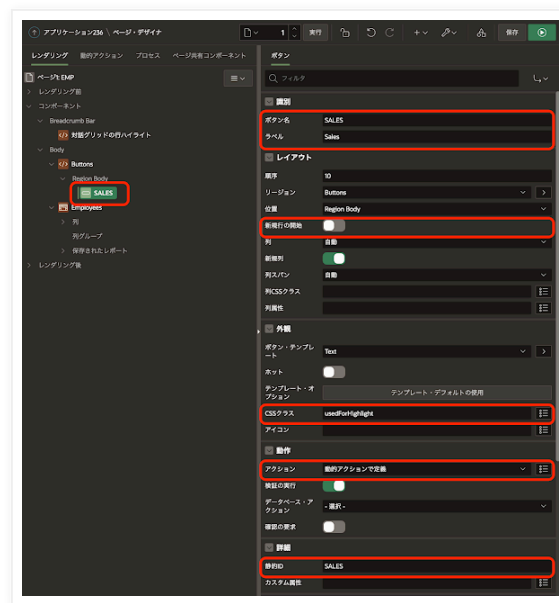
作成したリージョンにボタン**SALES**を作成します。識別のボタン名は**SALES**、ラベルは**Sales**とします。このボタンの上にマウス・ポインタが乗ったときに、列Departmentの値がSALESである行をハイライトさせます。

レイアウトの新規行の開始はオフにします。これから作成する他のボタンも同様にオフに設定し、それらのボタンを横並びに配置します。

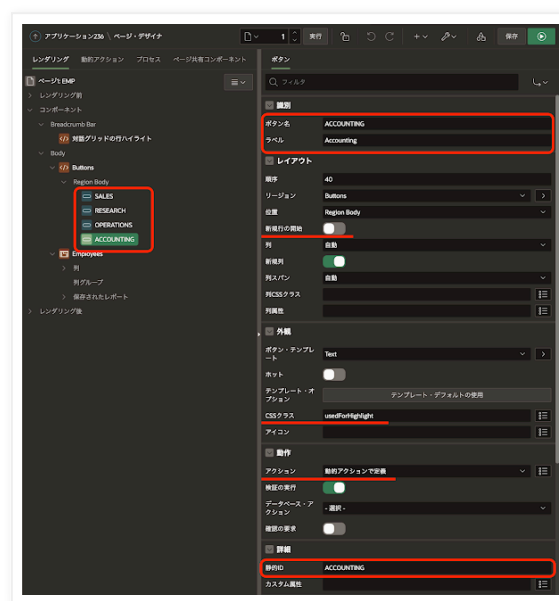
外観のCSSクラスとしてusedForHighlightを設定します。カスタム・イベントのターゲットに使用し、複数のボタンから呼び出される動的アクションを1カ所で定義します。

動作のアクションとして動的アクションで定義を選択します。ボタンを押しても何も実行されないようにします。

詳細の静的IDとしてSALESを指定します。この値と列Departmentの値の一致を確認します。



同様の設定でボタンRESEARCH、OPERATIONS、ACCOUNTINGを作成します。SALESの部分を置き換えます。



画面上に配置するコンポーネントの作成は以上です。

これから行ハイライトの実装を始めます。

ページ・プロパティのJavaScriptのファンクションおよびグローバル変数の宣言に以下を記述します。行ハイライトを行なうファンクション`applyHighlightRows`と、行ハイライトを解除する`removeHighlightRows`を含みます。

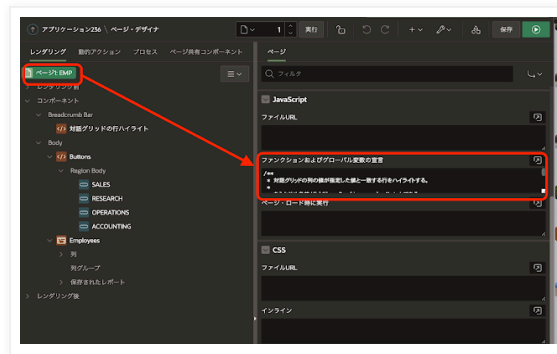
```
/**
 * 対話グリッドの列の値が指定した値と一致する行をハイライトする。
 *
 * あらかじめ名前がhlClassPrefix + valueMatchである
 * CSSクラスが定義されていること。
 *
 * @param igID 対話グリッドの静的ID
 * @param valueMatch この値と一致する行がハイライトされる
 * @param columnName 一致を確認する列
 * @param hlClassPrefix 列に追加するクラスの接頭辞
 */
function applyHighlightRows(igId, valueMatch, columnName, hlClassPrefix) {
    let grid = apex.region(igId).call("getViews","grid");
    let model = grid.model;
    model.forEach( (record, index, id) => {
        let value = model.getValue(record, columnName);
        if (value !== null && typeof value === "object") {
            // 列のタイプが選択リストやポップアップLOVの場合、表示値で一致を確認する。
            value = value.d;
        };
        if (value === valueMatch) {
            let meta = model.getRecordMetadata(id);
            meta.highlight = hlClassPrefix + valueMatch;
            model.metadataChanged(id);
        }
    });
}

/**
 * ハイライトを解除する。
 *
 * @param igId 対話グリッドの静的ID
 */
function removeHighlightRows(igId) {
    let grid = apex.region(igId).call("getViews","grid");
    let model = grid.model;
    model.forEach( (record, index, id) => {
        let meta = model.getRecordMetadata(id);
        if (meta.highlight !== null) {
            meta.highlight = null;
            model.metadataChanged(id);
        }
    });
}
```

```
}
```

highlight-row.js hosted with ❤ by GitHub

[view raw](#)



RecordMetadataのhighlightに設定するクラスは、実際にはTR要素のclass属性に追加されます。背景色の変更に使用するCSSクラスをページ・プロパティのCSSのインラインで定義します。

```
.a-GV-row.hlrSALES {
  --a-gv-background-color: var(--u-color-6);
}

.a-GV-row.hlrRESEARCH {
  --a-gv-background-color: var(--u-color-8);
}

.a-GV-row.hlrACCOUNTING {
  --a-gv-background-color: var(--u-color-10);
}

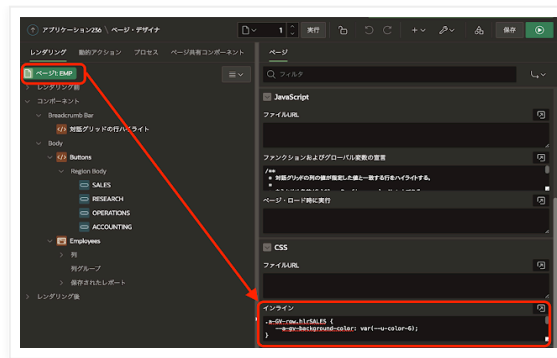
.a-GV-row.hlrOPERATIONS {
  --a-gv-background-color: var(--u-color-12);
}
```

highlight-row-background-color.css hosted with ❤ by GitHub

[view raw](#)

TR要素の子要素であるTDの背景色（background-colorの指定）は、CSS変数の--a-gv-background-colorが設定されています。そのため、TR要素でbackground-colorを指定しても、TD要素の--a-gv-background-color（デフォルトは白）が優先されます。

選択した行の背景色を変更するため、TD要素の背景色である--a-gv-background-colorの定義を変更しています。



左ペインで**動的アクション・ビュー**を開きます。カスタム・イベントで実行される動的アクションを作成します。

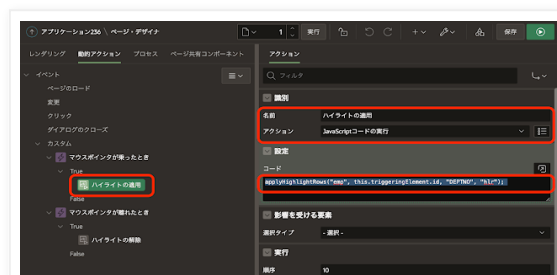
識別の名前は**マウスポインタが乗ったとき**とします。**イベントのタイミング**に**カスタム**を選択します。

カスタム・イベントは**mouseenter**（mouseenter自体は標準のブラウザ・イベントです）、**選択タイプ**は**jQueryセクタ**、**jQueryセクタ**として**usedForHighlight**を指定します。

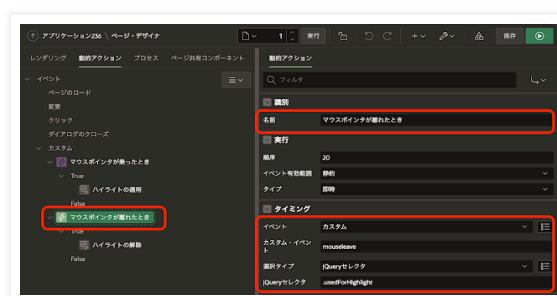


TRUEアクションの名前は**ハイライトの適用**、**アクション**として**JavaScriptコードの実行**を選択します。設定のコードに**applyHighlightRows**の呼び出しを記述します。

`applyHighlightRows("emp", this.triggeringElement.id, "DEPTNO", "h1r");`

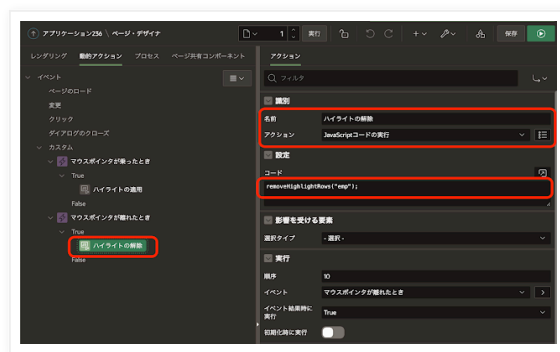


同様に、**動的アクション**として**マウスポインタが離れたとき**を作成します。**タイミング**の**カスタム・イベント**に**mouseenter**の代わりに**mouseleave**を指定します。



TRUEアクションとしてハイライトの解除を作成します。設定のコードにremoveHighlightRowsの呼び出しを記述します。

```
removeHighlightRows("emp");
```



以上でアプリケーションは完成です。アプリケーションを実行すると、記事の先頭のGIF動画のように動作します。

今回作成したAPEXアプリケーションのエクスポートを以下に置きました。
<https://github.com/ujnak/apexapps/blob/master/exports/ig-highlight-rows.zip>

Oracle APEXのアプリケーション作成の参考になれば幸いです。

完

Yuji N. 時刻: 12:34

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.