

# 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。

2022年5月26日 木曜日

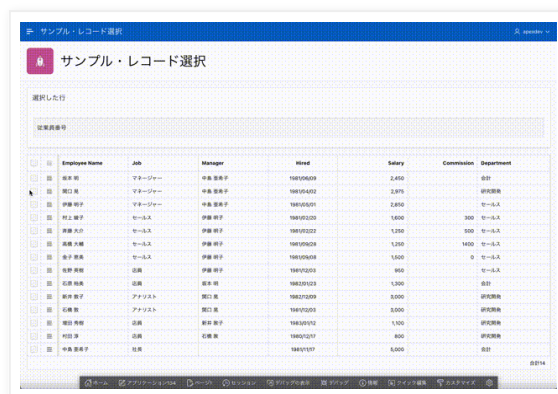
## レポート上で選択した行のIDをページ・アイテムに設定する

レポート上で行を選択して、その情報を使いたいというのはよくある要件です。簡単に実装方法をまとめてみました。

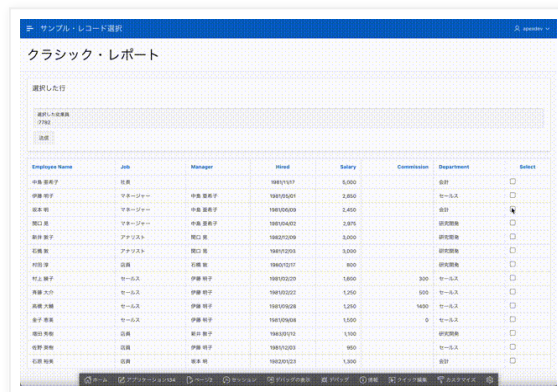
サンプルの実装には、**サンプル・データセット**に含まれる**EMP/DEPT**の表**EMP**のデータを使用しています。表EMPに含まれる従業員を選択し、選択された従業員の従業員番号をページ・アイテムに設定します。

対話グリッドでは動的アクションとして実装します。行セレクトや選択行の変更イベントなどが標準で実装されているため、追加で記述するコードが少なくて済みます。また、動的アクションなので、選択を確定するためにページを送信する必要がありません。

推奨は、対話グリッドを使った実装です。



クラシック・レポートでも実装できます。**APEX\_ITEM.CHECKBOX2**を使って、チェックボックスの列をレポートに追加します。

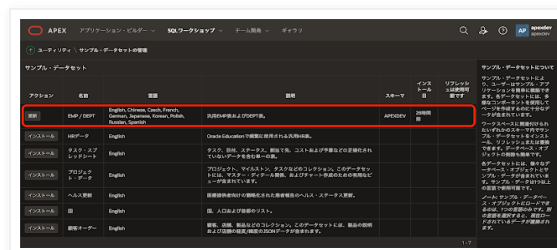


以下より、両方の実装について紹介します。

# アプリケーションの作成

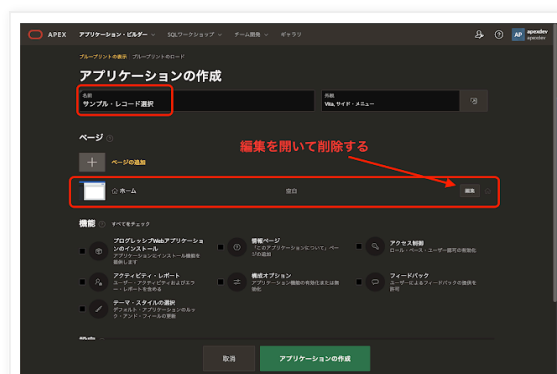
あらかじめ**サンプル・データセット**のEMP/DEPTをインストールします。

**SQLワークショップ**のユーティリティの**サンプル・データセット**からインストールします。

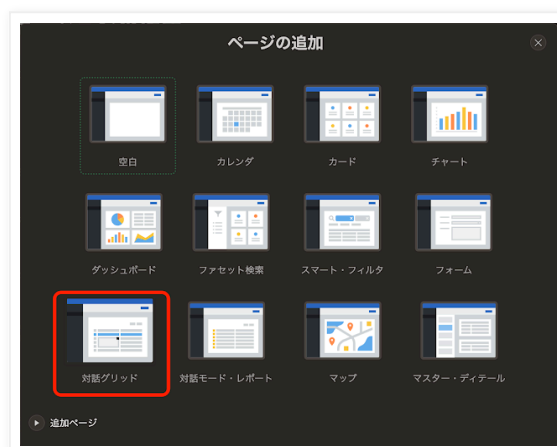


**アプリケーション作成ウィザード**を起動します。作成するアプリケーションの名前は**サンプル・レコード選択**とします。アプリケーションには、対話グリッドとクラシック・レポートのページを作成します。

最初に**ホーム・ページ**の**編集**を開いて、ページを削除します。



**ページ**の追加をクリックし、**対話グリッド**を選択します。

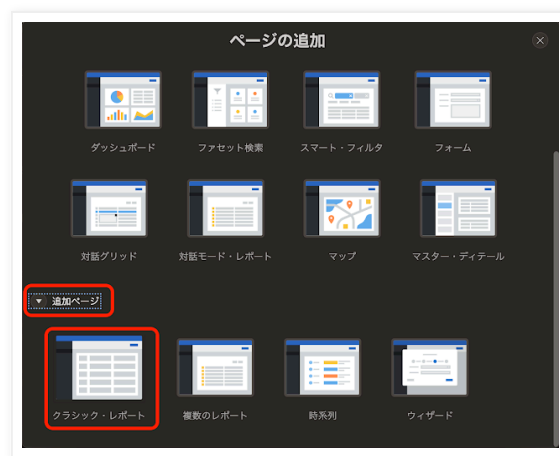


ページ名を**対話グリッド**とし、**表またはビュー**、**編集を許可**を選択します。表またはビューとして表**EMP**を選択します。

以上で、**ページ**の追加をクリックします。



再度、**ページの追加**をクリックします。今度はクラシック・レポートを追加します。**追加ページ**を開いて、**クラシック・レポート**を選択します。

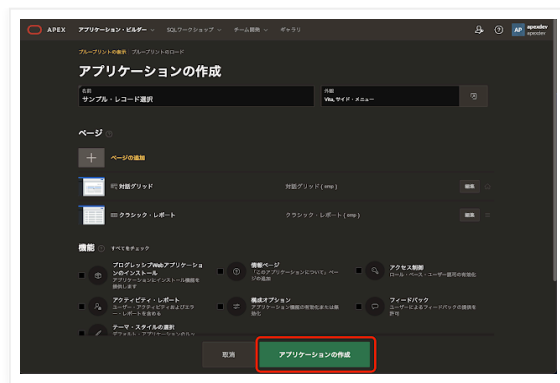


ページ名を**クラシック・レポート**とし、**表またはビュー**、**クラシック・レポート**を選択します。**表またはビュー**として**表EMP**を選択します。

以上で、**ページの追加**をクリックします。



アプリケーションの作成を実行します。

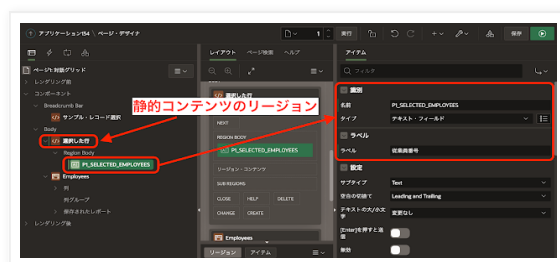


以上で、アプリケーションの準備は完了です。

## 対話グリッドでの実装

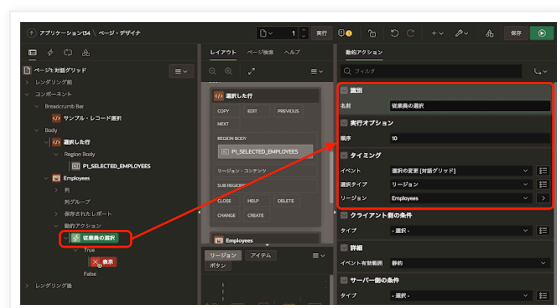
対話グリッドの行セレクトを使って、選択した行をページ・アイテムに設定します。

対話グリッドのページを開き、静的コンテンツのリージョンを作成します。作成したリージョンにページ・アイテムP1\_SELECTED\_EMPLOYEESを作成します。このページ・アイテムに選択した行、つまり複数の従業員の従業員番号を','区切りで設定します。



対話グリッドのリージョンEmployeesに動的アクションを作成します。

識別の名前は従業員の選択とします。タイミングのイベントとして選択の変更[対話グリッド]を指定します。選択タイプはリージョン、リージョンはEmployeesになります。



TRUEアクションとして、JavaScriptコードの実行を選択し、設定のコードに以下を記述します。

```
// Fetch model.getRecord in Row Select mode of the IG
// https://community.oracle.com/tech/developers/discussion/4478408/fetch-model-getrecord-in-row
let selectedRecords = this.data.selectedRecords,
    recordIds = [];
```

```

if (selectedRecords.length > 0) {
    let model = this.data.model;
    // got some selected records
    // turn the records into record ids
    recordIds = selectedRecords.map(r => { return model.getRecordId(r);});
} else {
    // maybe there is nothing selected or maybe this is grid view and it is in cell range select
    let region = apex.region(this.triggeringElement.id),
        view = region.call("getCurrentView");

    if (view.internalIdentifier === "grid" && view.view$.grid("option", "selectCells")) {
        // yup it is grid view in cell range mode
        let range = view.view$.grid("getSelectedRange");
        if (range) {
            recordIds = range.recordIds;
        }
    }
}

apex.items.P1_SELECTED_EMPLOYEES.setValue(recordIds);

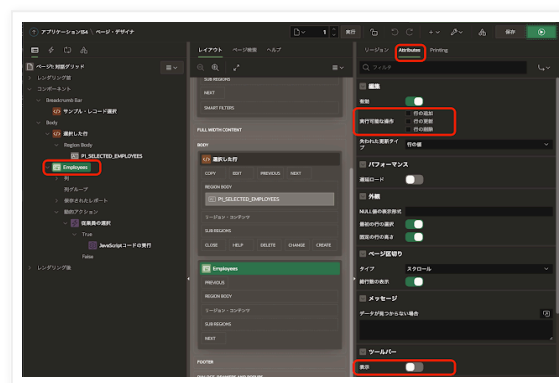
```

grid\_select\_rows.js hosted with ❤ by GitHub

[view raw](#)

対話グリッドでの実装は以上で完成です。

対話グリッドのAttributesの設定を変更し、行の追加、行の更新、行の削除をできないようにしたり、ツール・バーの表示をOFFにして、行の選択以外できないようにすると、より使いやすくなるでしょう。



## クラシック・レポートでの実装

クラシック・レポートでも同様に、選択した従業員を設定するページ・アイテム **P2\_SELECTED\_EMPLOYEES** を作成します。さらに、送信ボタン **B\_SUBMIT** も作成します。ボタンの動作のアクションはページの送信とします。

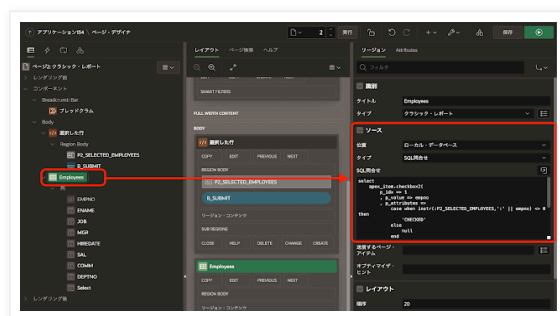


クラシック・レポートのリージョン**Employees**を選択し、**ソースのタイプ**を**SQL問合せ**に変更します。レポートにチェックボックスが含まれるよう、**SQL問合せ**に**APEX\_ITEM.CHECKBOX2**を含んだ**SELECT**文を記述します。

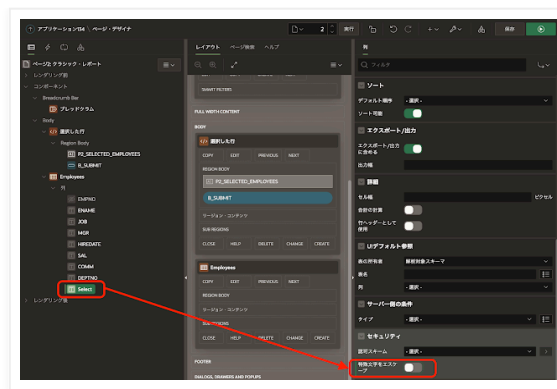
```
select
  apex_item.checkbox2(
    p_idx => 1
    , p_value => empno
    , p_attributes =>
      case when instr(:P2_SELECTED_EMPLOYEES,':' || empno) <> 0 then
        'CHECKED'
      else
        null
      end
  ) "Select"
, empno
, ENAME
, JOB
, MGR
, HIREDATE
, SAL
, COMM
, DEPTNO
from EMP
```

select\_with\_checkbox.sql hosted with ❤ by GitHub

[view raw](#)



追加した列**Select**はHTMLが生成されるため、**セキュリティ**の**特殊文字をエスケープ**を**OFF**にします。



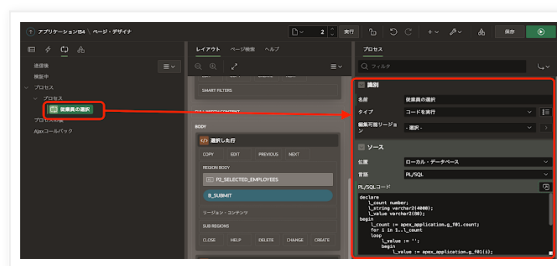
レポート上での選択結果は、ページの送信時にAPEX\_APPLICATION.G\_F01としてサーバーに渡されます。それを受け取って、ページ・アイテムP2\_SELECTED\_EMPLOYEESに設定するプロセスを作成します。

作成したプロセスの**名前**を**従業員の選択**とし、**ソース**の**PL/SQLコード**として以下を記述します。

```
declare
    l_count number;
    l_string varchar2(4000);
    l_value varchar2(80);
begin
    l_count := apex_application.g_f01.count;
    for i in 1..l_count
    loop
        l_value := '';
        if apex_application.g_f01.exists(i) then
            l_value := apex_application.g_f01(i);
        end if;
        l_string := l_string || ':' || l_value;
        apex_debug.info(l_string);
    end loop;
    :P2_SELECTED_EMPLOYEES := l_string;
end;
```

set\_selected\_employees\_to\_page\_items.pls hosted with ❤ by GitHub

[view raw](#)



以上でクラシック・レポートでの実装も完了です。

今回作成した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/sample-record-selection.sql>

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: 15:25

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.

---