

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。

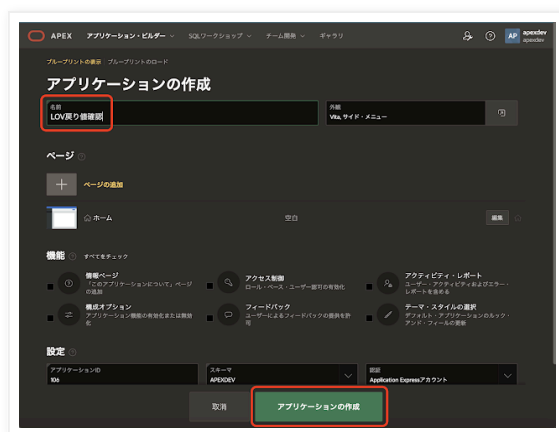
2022年2月4日 金曜日

LOVの戻り値を確認する

動的LOVの戻り値を確認するにはどうしたらよいかという相談があったので、少し考えてみました。

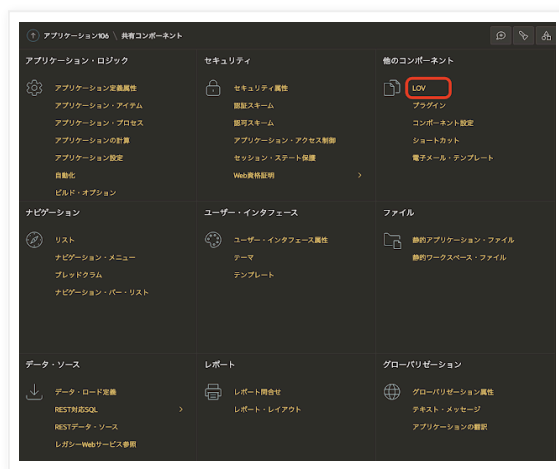
確認のために、最初に空のアプリケーションを作ります。

アプリケーション作成ウィザードを起動し、名前をLOV戻り値確認とします。追加指定はなしで、アプリケーションの作成を実行します。



LOVは共有コンポーネントとして作成します。

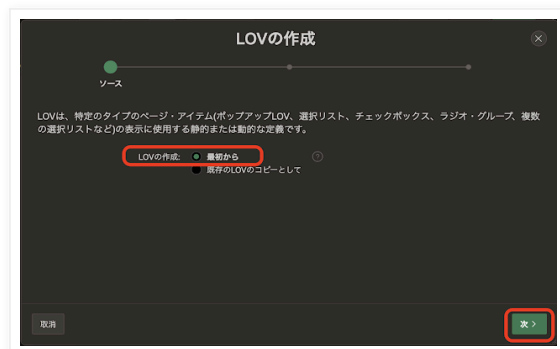
共有コンポーネントのLOVを開きます。



作成を実行します。



LOVの作成は最初から行います。次へ進みます。



標準のディクショナリ・ビューのALL_USERSを元にLOVを定義するので、名前はALL_USERSとします。タイプはDynamicです。

次へ進みます。



データ・ソースはローカル・データベース、ソース・タイプはSQL Queryを選択します。SQL SELECT文を入力には以下を記述します。

```
select * from all_users
```



ソース・タイプとして表を選んでいることと、SELECT * FROM 表/ビューをSQLとして記述していることに、実質的な違いはありません。

戻り値にUSER_ID、表示値にUSERNAMEを選択します。作成を実行します。

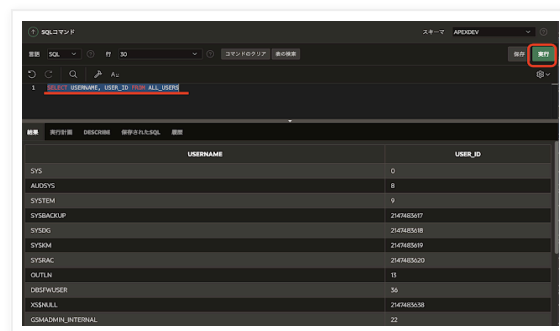


LOVとしてALL_USERSが作成されました。



表示値と戻り値の対応は、SQLワークショップのSQLコマンドにて、以下のSELECT文を実行することにより確認できます。

```
SELECT USERNAME, USER_ID FROM ALL_USERS
```



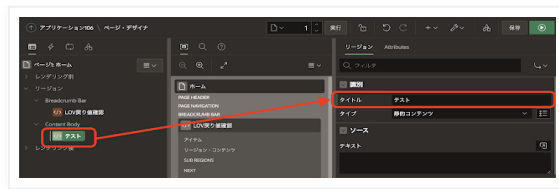
LOVとしてUSERNAMEのどれかが選択されているときの戻り値は、それに対応しているUSER_IDの値になります。

今回は画面にページ・アイテムを配置した上で、戻り値を確認します。確かにカスケード指定がある場合や、バインド変数を使っている場合など、簡単にSELECT文を取り出して実行できない場合があります。

作成したLOVのALL_USERSを扱うページ・アイテムを作成します。

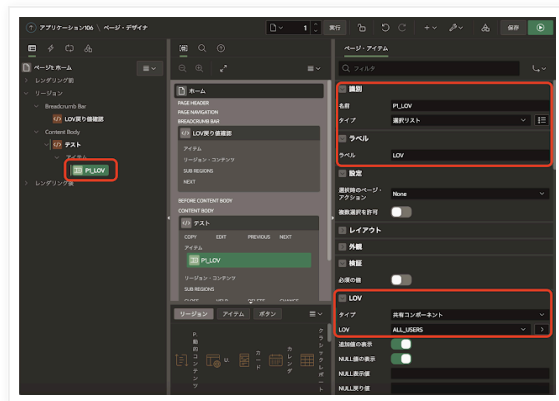
ページ・デザイナーにてホーム・ページ（ページ番号1）を開きます。Content Bodyの上でコンテキスト・メニューを表示させ、リージョンの作成を実行します。

作成したリージョンの識別の名前はテストとします。

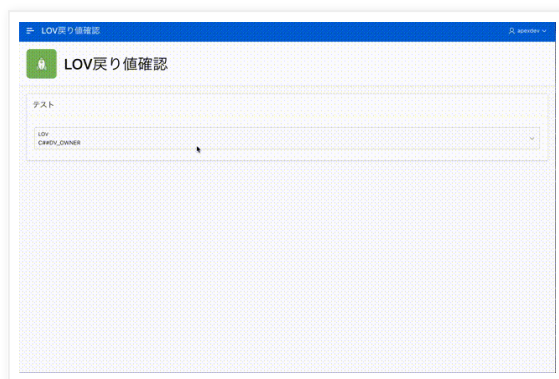


リージョン上でコンテキスト・メニューを表示させ、ページ・アイテムの作成を実行します。

識別の名前をP1_LOV、タイプを選択リストとします。ラベルはLOV、LOVの設定としてタイプを共有コンポーネント、LOVは先ほど作成したALL_USERSを選択します。

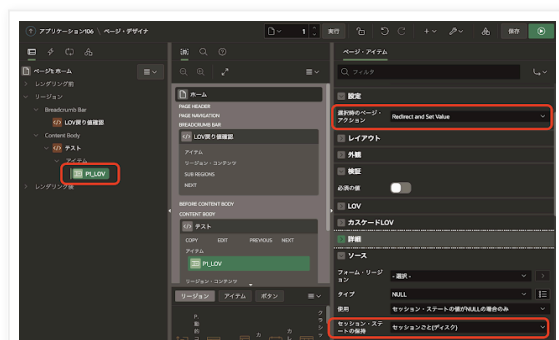


この状態でページを実行します。確かにこのままでは、LOVの戻り値（ページ・アイテムP1_LOVの値）を確認できません。

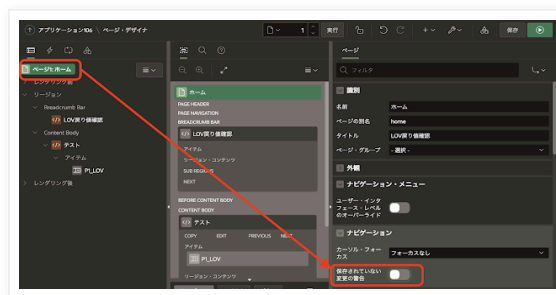


ページをリダイレクトさせて確認する

設定の選択時のページ・アクションとしてRedirect and Set Valueを選択します。ソースのセッション・ステートの保持がセッションごと(ディスク)になっていることを確認します。



このままでは、LOVで選択を切り変える度に警告が表示されます。そのため、ページ・プロパティのナビゲーションの保存されていない変更の警告をOFFに切り替えます。



この設定により、LOVの表示値を切り替えたときに以下のようなURLを宛先としたGETリクエストが発行されます。

`https://ホスト/ords/f?p=106:1:132892018777395::NO::P1_LOV:58`

現在開いているページがターゲットです。そのURLにページ・アイテム名と戻り値が追加されています。

ページを実行しLOVを切り替えてから、開発者ツール・バーよりセッションを呼び出します。



デバッグ用のウィンドウが新たに開いて、セッション・ステートとして保存されている（つまりデータベースに保存されている）ページ・アイテムとそのアイテム値を確認することができます。



この値はデータベース、つまりサーバー側に保存されている値であり、ブラウザ上の値とは同期していません。

ページを送信して確認する

ページ・アイテムP1_LOVの設定の選択時のページ・アクションをSubmit Pageに変更することで、LOVの戻り値を確認します。



ページを保存し、先ほどと同様の操作を行います。

LOVで選択した値は、サーバーにHTTPのPOSTリクエストによって送信されます。見かけ上は、Redirect and Set Valueと違いがありません。

動的アクションを設定して確認する

今までの設定はブラウザ上で選択した値をサーバーに送信して（Redirect and Set ValueはGETで送信、Submit PageではPOSTで送信）、サーバーにセッション・ステートとして保存された値を確認していました。

ページ・アイテムのソースのセッション・ステートの保持が、リクエストごと(メモリーのみ)が選択されていると、セッション・ステートとして保存されない（つまりデータベースに保存されない）ため、デバッグ画面のセッションにはページ・アイテムおよびアイテム値は表示されません。



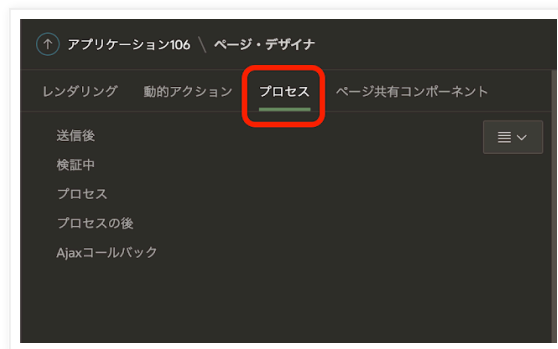
実際、ページ・アイテムのアイテム値をセッション・ステートとして保持する必要は、あまりありません。

APEXアプリケーションは、それぞれのページを単位として処理が行われます。

ページ・デザイナーの左ペインのレンダリング・ビューに定義されている内容が、HTTPのGETリクエストを受け付けて行われる処理になります。



プロセス・ビューに定義されている内容が、POSTリクエストを受け付けて行われる処理です。（今までのところ何も定義していないので、画面上は空です）。

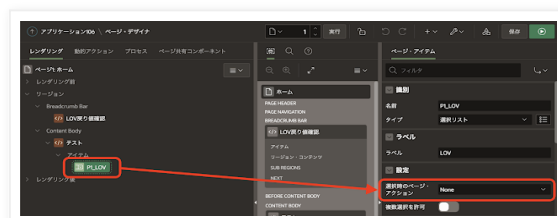


セッション・ステートの保持が、リクエストごと(メモリーのみ)の場合、ページ・アイテムの値は、1 ページ内で定義されたレンダリング・ビューでの処理、またはプロセス・ビューで定義された処理の間だけ、アイテム値が有効です。

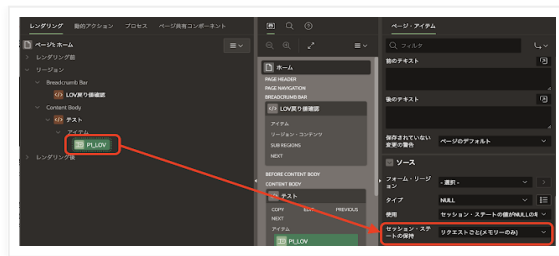
セッション・ステートに保持していると（データベースに保存することになる）、次の呼び出し時に前回のページ・アイテムの値を参照することができます。しかし、前回のページ・アイテムの値が必要になる場合はほとんどありません。

そのため、通常、ページ・アイテムのセッション・ステートの保持は、リクエストごと(メモリーのみ)に設定します。例外は、表を編集するために作成されるフォームの主キーとなるページ・アイテムくらいです。

選択リストの設定の選択時のページ・アクションをNoneに変更します。



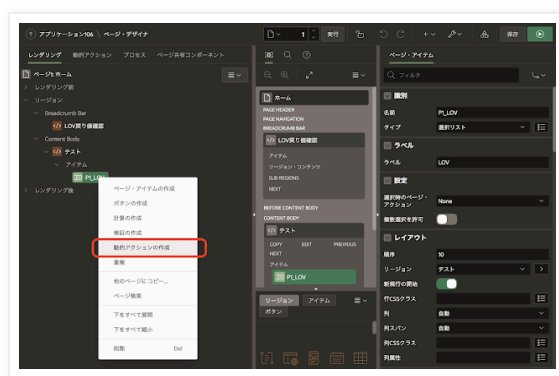
ソースのセッション・ステートの保持をリクエストごと(メモリーのみ)に設定します。



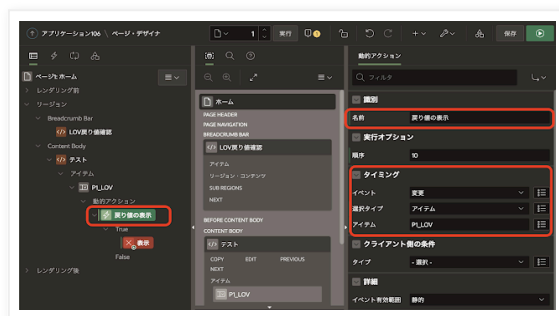
画面を実行して、選択リストの値を切り替えてもサーバー側に値は送信されない（GETリクエストもPOSTリクエストも発行されない）ため、**デバッグ画面のセッション**からページ・アイテム P1_LOVのアイテム値を確認することはできません。

変更された値を確認するため、JavaScriptコンソールにアイテム値に出力させてみます。

ページ・アイテム上でコンテキスト・メニューを表示させ、**動的アクション**の作成を実行します。



作成された**動的アクション**の名前を**戻り値の表示**とします。JavaScriptとして実行する処理の**タイミング**として、**イベント**が**変更**、**選択タイプ**が**アイテム**、**アイテム**として**P1_LOV**を選択します（これがページ・アイテムで動的アクションを作成したときのデフォルトです）。つまり、ページ・アイテムP1_LOVの値が変更されたときに、次に定義するアクションが実行されます。

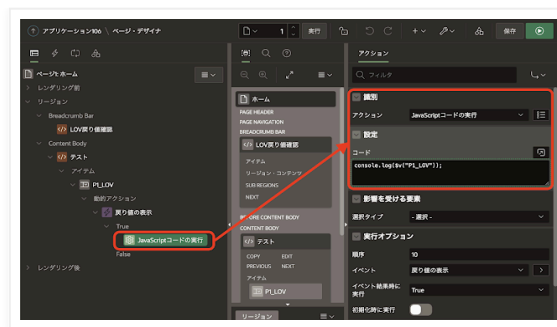


設定したタイミングで実行されるJavaScriptコードとして、以下を記述します。

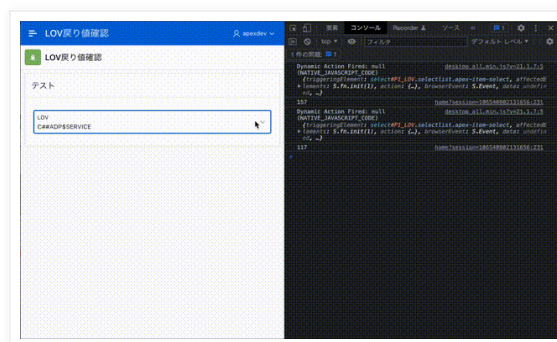
```
console.log($v("P1_LOV"));
```

ファンクション\$vhはOracle APEXが提供しているJavaScriptのライブラリに含まれています。APIのリファレンスは以下になります。APEXのバージョンごとに違いがあるので、使用しているバージョンのリファレンスを参照する必要があります。

<https://docs.oracle.com/en/database/oracle/application-express/21.2/aexjs/index.html>



以上を設定しページを実行し選択リストを切り替えると、切り替えるたびにJavaScriptコンソールに戻り値が表示されます。



そもそも戻り値を確認したくなる理由

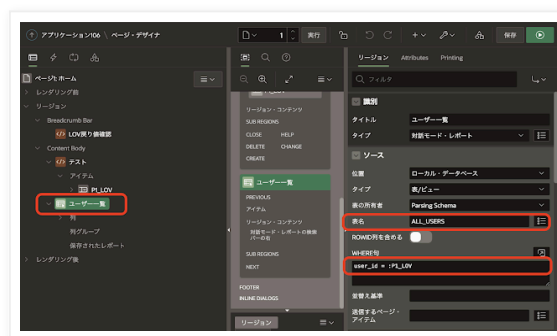
ページ・アイテムのアイテム値は、ブラウザ上に見えている値とサーバーにセッション・ステートとして保存されている値は一致しません。セッション・ステートから参照されるページ・アイテムのアイテム値は、**以前に保持していた値を参照する**ためのものだからです。

選択した値を使ってレポートやチャートを再表示したい、というのは多いです。

対話モード・レポートのリージョンを作成し、**表名にALL_USERS、WHERE句として**

user_id = :P1_LOV

を設定します。選択リストで選択したユーザーの行のみがレポートに表示されることを期待しています。

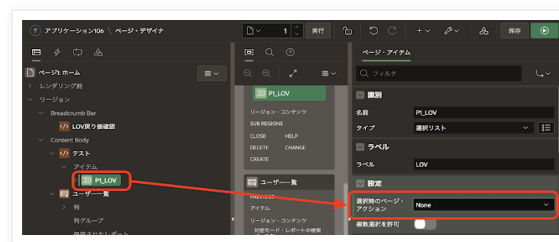


選択リストの設定の選択時のページ・アクションがRedirect and Set Valueのときは、期待通りの動作をします。

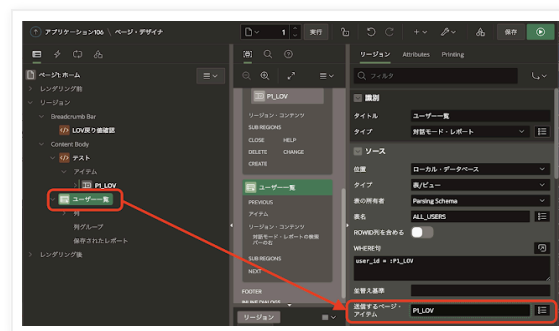
選択リストの設定の選択時のページ・アクションがSubmit Pageのときは、期待通りの動作はしません。送信されたP1_LOVの値はPOST処理の間のみ有効で、画面の再描画が行われるレンダリング処理には引き継がれずNULLになるためです。

最近GETやPOSTリクエストを発行して画面を再描画するよりは、Ajaxコールを発行し部分的に画面を再描画する実装が多くなっています。そのため、**選択時のページ・アクション**を**Redirect and Set Value**にする代わりに、動的アクション（つまりブラウザ上のJavaScriptの実行）によるレポートやチャートの再描画を実装してみます。

選択時のページ・アクションはNoneとします。

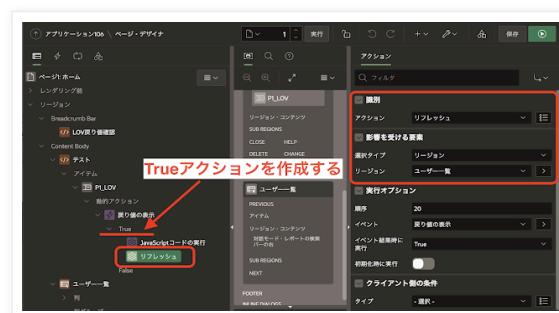


対話モード・レポートのリージョンのソースの送信するページ・アイテムとしてP1_LOVを指定します。リージョンが再描画される際に、ブラウザ上で保持されているP1_LOVの値がサーバーに（Ajaxコールの引数として）送信されます。



選択リストを切り替えた時に対話モード・レポートが再描画されるよう、動的アクションのTrueアクションを作成します。

識別のアクションとしてリフレッシュを選択します。影響を受ける要素の選択タイプをリージョン、リージョンに再描画の対象となるリージョンを選択します。



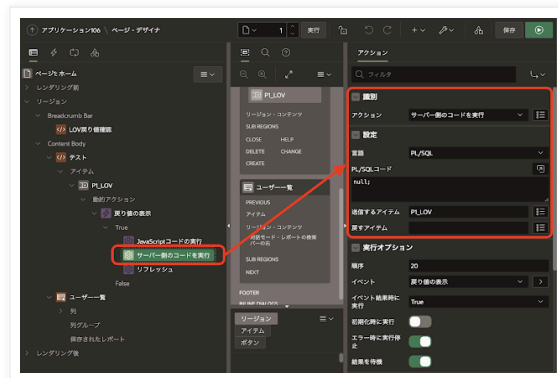
以上で、ページ全体ではなくリージョンの部分更新が実装できました。



選択リストの値がレポートやチャートの再描画時に反映されない場合、**送信するページ・アイテムの設定がされていない**ケースが多いです。

少し特殊な設定として、ブラウザ上の値を動的アクションで呼び出すPL/SQLコード内で使いたい、というケースがあります。

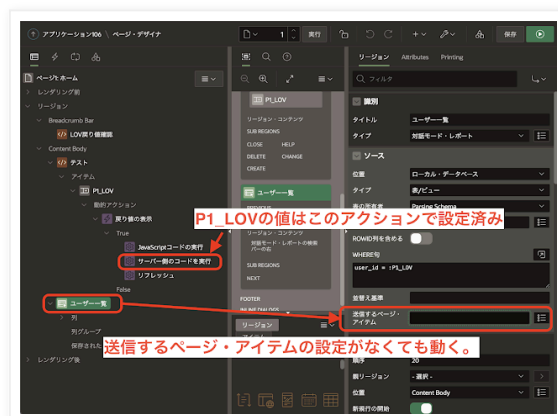
Trueアクションの識別のアクションとして、**アクションをサーバー側のコードを実行**を選択し、**PL/SQLコード**に何もしないという意味のnull;を記述します。**送信するアイテム**として**P1_LOV**を指定します。これでブラウザの画面上の値がサーバーに送信されます。



ページ・アイテム**P1_LOV**のソースの**セッション・ステートの保持がセッションごと(ディスク)**となっている場合は、このアクションで送信されたアイテム値がディスクに保存されます。

結果としてデバッグ画面のセッションから、ページ・アイテムのアイテム値を確認することができます。

また、後続の処理では、設定済みのページ・アイテムの値を参照することができます。



LOVの戻り値を確認する方法の説明は以上になります。

簡単なアプリケーションですが、エクスポートを以下に置きました。

<https://github.com/ujnak/apexapps/blob/master/exports/lovreturnvalue.sql>

LOVの戻り値だけの説明ではなくなりましたが、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: 11:40

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.
