

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。

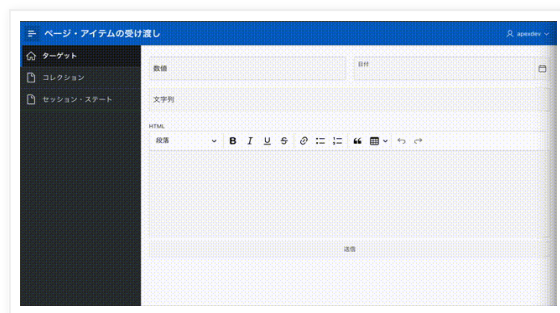
2022年7月4日月曜日

ページ・アイテムの値を次を開くページに引き継ぐ方法について

フォームを開いているページから別のページのフォームを開く際に、ページ・アイテムの値を引き継ぎたいケースがあります。この機能を実装するにあたって、以下の3つの方法について紹介します。

1. ターゲットに指定する。
2. コレクション（APEX_COLLECTION）を使う。
3. セッション・ステートに保存する。

サンプルとなるアプリケーションを作成し、実行した結果です。

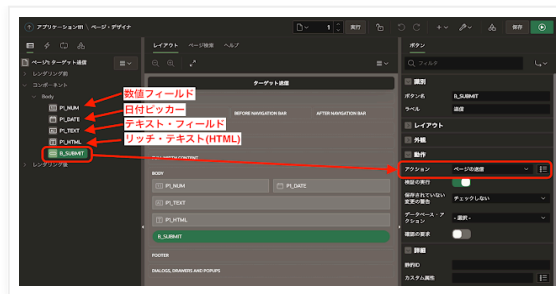


おおむね、同じ動作をするように実装しています。

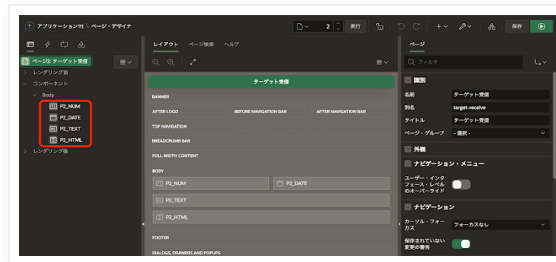
サンプル・アプリケーションについて

送信元となるページには、以下のように4つのページ・アイテムを作成しています。P1_NUM（数値フィールド）、P1_DATE（日付ピッカー）、P1_TEXT（テキスト・フィールド）、P1_HTML（リッチ・テキスト・エディタ、書式はHTML）です。

ページを送信するボタンとしてB_SUBMITを作成しています。動作のアクションはページの送信になります。



ボタンB_SUBMITをクリックして送信したページ・アイテムの値は、移動先のページのページ・アイテムに設定します。ページ・アイテム名のページ番号部分は変わりますが、それ以外は同じ設定のページ・アイテムです。



コレクションとセッション・ステートは、それぞれ別のページに実装しています。ページ・アイテムの名前はコレクションの場合はP3とP4、セッション・ステートの場合はP5とP6で始まります。

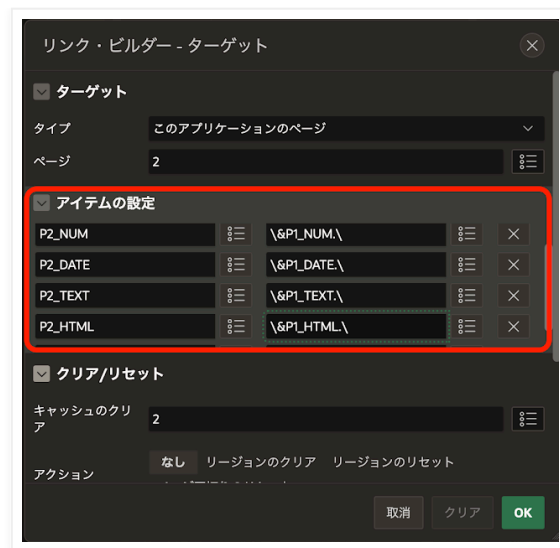
ターゲットに指定する

送信元のページのプロセス・ビューを開き、**ブランチ**を作成します。**動作のタイプ**はページまたはURL(リダイレクト)を選択し、**ターゲット**として移動先となるページを指定します。

サーバー側の条件として、ボタン押下時にB_SUBMITを選択します。



ターゲットのアイテムの設定として、移動先のページ・アイテムに渡す値を指定します。



実際のブランチの処理は、以下のURLへのリダイレクトになります。

```
f?p=
&APP_ID.:2:&SESSION.::&DEBUG.:2:P2_NUM,P2_DATE,P2_TEXT,P2_HTML:\&P1_NUM.\,\&P1_DATE.
.\,\&P1_TEXT.\,\&P1_HTML.\&success_msg=#SUCCESS_MSG#
```

ページ・アイテムの値を設定している部分に注目すると、以下のようにカンマでそれぞれのページ・アイテムの値が区切られています。

\&P1_NUM.\,\&P1_DATE.\,\&P1_TEXT.\,\&P1_HTML.

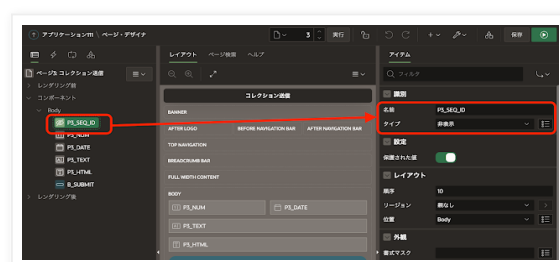
カンマ（およびコロン:）がページ・アイテムの値に含まれると、ページ・アイテムの値はそこで終了と認識されます。バックスラッシュで囲むことにより、その間にあるカンマは区切り文字と認識されないようになります。

また、データが大きいとURLとしても長くなり、上限値（ブラウザ依存）を超える場合も起こり得ます。結果として意図した通りに値が渡されません。

ターゲットに指定して渡すページ・アイテムは、表の主キーなどに限定した方が良いでしょう。任意の文字列などを含めるのは避けるべきです。

コレクション（APEX_COLLECTION）を使う

送信するページ・アイテムとして、**非表示**のページ・アイテムP3_SEQ_IDを追加します。その値を受信するページ・アイテムとしてP4_SEQ_IDも追加します。

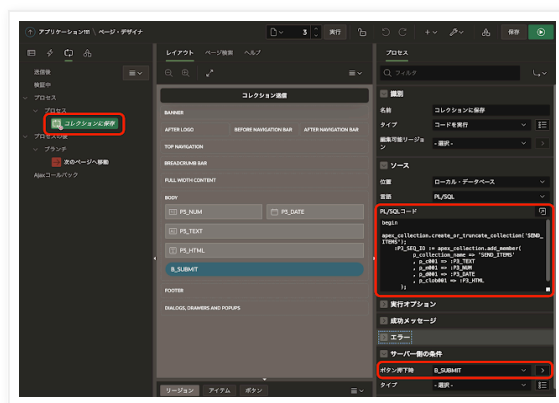


ボタンB_SUBMITを押し時に受け取ったページ・アイテムを、コレクションに保存するプロセスを作成します。

ソースのPL/SQLコードとして、以下を記述します。

```
begin
  apex_collection.create_or_truncate_collection('SEND_ITEMS');
  :P3_SEQ_ID := apex_collection.add_member(
    p_collection_name => 'SEND_ITEMS'
    , p_c001 => :P3_TEXT
    , p_n001 => :P3_NUM
    , p_d001 => :P3_DATE
    , p_clob001 => :P3_HTML
  );
end;
```

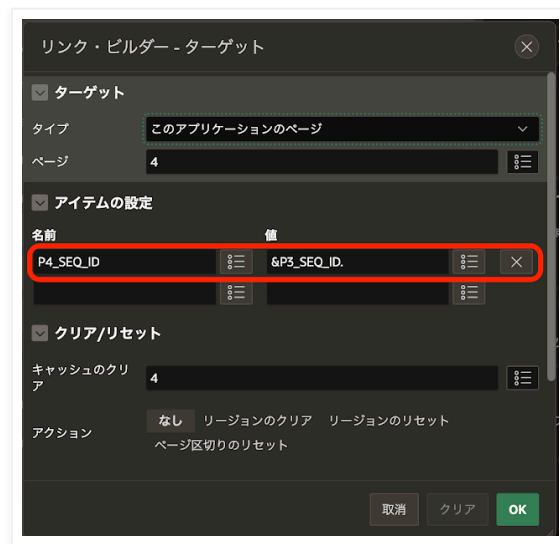
コレクション**SEND_ITEMS**に、ページ・アイテムの値を保存します。コレクションに保存された値を取り出す際に使用するシーケンス番号を、ページ・アイテム**P3_SEQ_ID**に保存します。



ページの移動は、ターゲットで指定した場合と同様に**ブランチ**を使います。



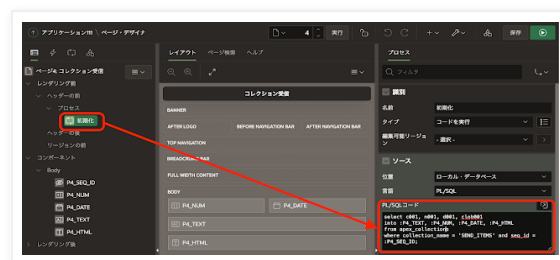
ターゲットの設定では、シーケンス番号**P3_SEQ_ID**のみを移動先のページのページ・アイテム**P4_SEQ_ID**に渡します。



移動先のページに、コレクションに保存された値をページ・アイテムに設定するプロセスを作成します。

ソースのPL/SQLコードは以下になります。

```
select c001, n001, d001, clob001
into :P4_TEXT, :P4_NUM, :P4_DATE, :P4_HTML
from apex_collections
where collection_name = 'SEND_ITEMS' and seq_id = :P4_SEQ_ID;
```

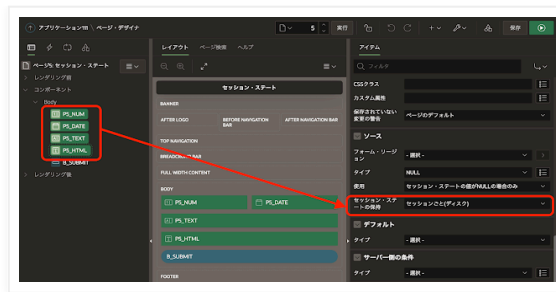


シーケンス番号を保持するページ・アイテムP4_SEQ_IDのセキュリティのセッション・ステート保護は、**チェックサムが必要 - セッション・レベル**を選択します。

コレクションの利用は一番手間がかかりますが、その分クリーンな実装になります。

セッション・ステートに保存する

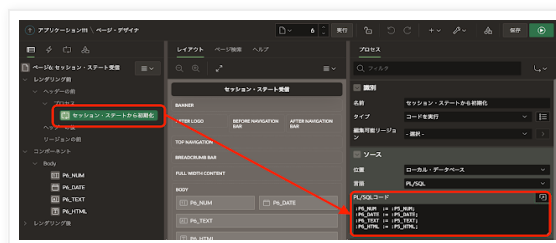
送信元となるページ・アイテムのソースのセッション・ステートの保持をセッションごと(ディスク)にすると、ページの送信の実行によってサーバーに送信されたページ・アイテムの値は、セッション・ステートとして保存されます。簡単にいうと、**データベースにコミットされます**。



移動先のページでは、送信されたページ・アイテムの値を直接参照することができます。ブランチでのページ・アイテムの設定は不要です。

初期化を行うプロセスの、ソースのPL/SQLコードは以下になります。

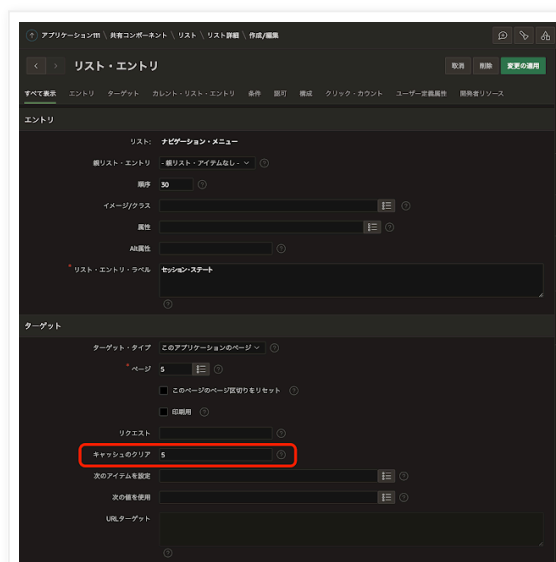
```
:P6_NUM := :P5_NUM;
:P6_DATE := :P5_DATE;
:P6_TEXT := :P5_TEXT;
:P6_HTML := :P5_HTML;
```



ただし、このように設定すると送信元のページ・アイテムをどこからでも設定および参照できるようになります。送信元のページを編集するような場合、その影響を調べるのが難しくなります。

セッション・ステートの保持をセッションごと(ディスク)に設定したページ・アイテムに、値を設定したり値を参照したりする処理は、できるだけ限定します。その方が、メンテナンスしやすいアプリケーションになります。

ページ・アイテムの値がセッション・ステートに保存されていると、ページを開いた時に以前の値がページ・アイテムに初期値として設定されます。ナビゲーション・メニューからページを開いたときに必ず空白にする場合は、**ナビゲーション・メニューのリスト・エントリのターゲットのキャッシュのクリアにページ番号を指定します。**



以上で、ページ・アイテムの値を引き継ぐ方法の紹介は終了です。

今回のサンプル・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/send-page-items.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: 18:11

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.
