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5月15日土曜日

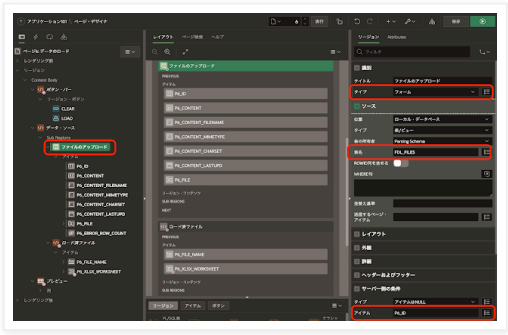
データをロードするページにユーザー表を使用する

こちらの記事の続きになります。すでにアプリケーションが作成済み、ページ作成ウィザードにより、ページ番号6としてデータのロードを行うページが存在していることを前提とします。

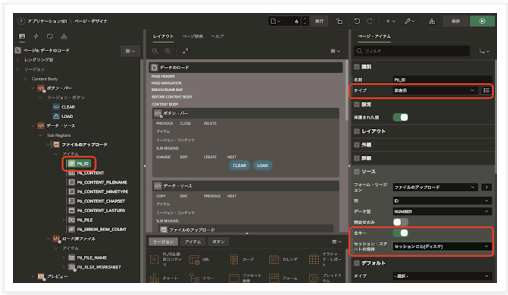
以下より、変更作業について記述します。

表FLD_FILESの列CONTENTにファイルをアップロードするよう、リージョン**ファイルのアップロード**を静的コンテンツから**フォーム**へ変更し、BLOBへの保存を行うプロセスを追加します。

タイプを**フォーム**に変更し、**ソース**の**表名**に**FDL_FILES**を選択します。**サーバー側の条件**の**アイテム**はF6_FILEから**F6_ID**へ変更します。

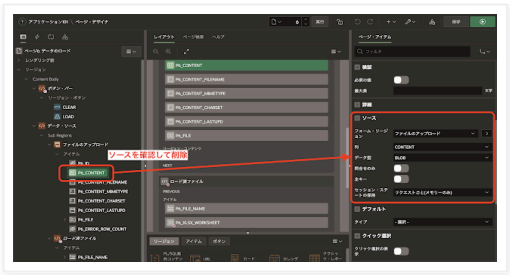


新規に認識されたページ・アイテム**P6_ID**を選択し、**タイプ**を**非表示**に変更します。**主キー**を**ON**、**セッション・ステートの保持**を**セッションごと(ディスク)**に切り替えます。



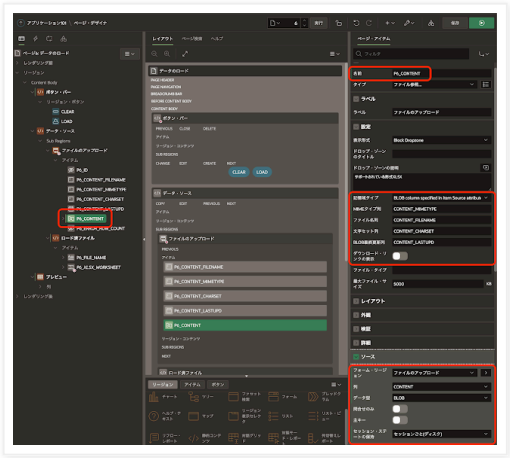
データのロードを行うページは、いくつかの処理で**ページの送信**(フォームのPOST処理)を行います。ページの送信後ページが再表示される際に、**セッション・ステートの保持**が**リクエストごと(メモリーのみ)**であると、変更されたページ・アイテムが保持されず初期状態に戻ります。モーダル・ダイアログなど、ページの送信後に別ページに遷移する場合は、送信時のページ・アイテムの値を保持する必要はありませんが、今回は送信時のページを再表示するため、ページ・アイテムの値を保持するように設定します。

ページ・アイテム**P6_CONTENT**のソース定義を確認した後、**削除**します。すでに存在するP6_FILEをP6_CONTENTに変更します。



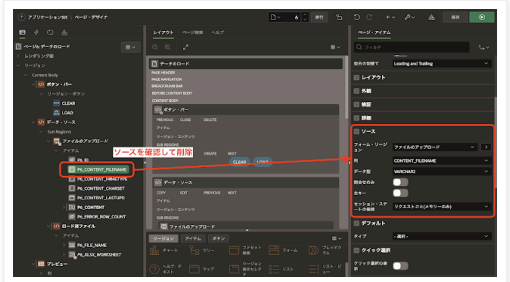
ページ・アイテム**P6_FILE**を選択し、**名前**を**P6_CONTENT**に変更します。**設定**の**記憶域タイプ**を**BLOB column specified in Item Source attribute**に変更し、ソースに定義されているBLOBカラムに保存するよう指定します。**MIMEタイプ列**は**CONTENT_MIMETYPE**、**ファイル名列**は**CONTENT_FILENAME**、**文字セット列**は**CONTENT_CHARSET**、**BLOB最終更新列**は**CONTENT_LASTUPD**を指定し、**ダウンロード・リンクの表示**は**OFF**にします。

ソースの定義として、**フォーム・リージョン**に**ファイルのアップロード**を選択します。**列**に**CONTENT**を選択し、**データ型**は**BLOB**、**セッション・ステートの保持**は**セッションごと(ディスク)**を選択します。

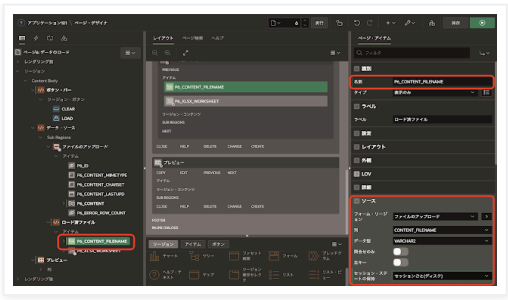


ページ・アイテム名は内部的にはIDにて保存されており(選択リストでページ・アイテムを設定するプロパティの場合)、P6_FILEとして指定されていたプロパティは改めて設定を変更しなくてもP6_CONTENTとして認識されるようになります。

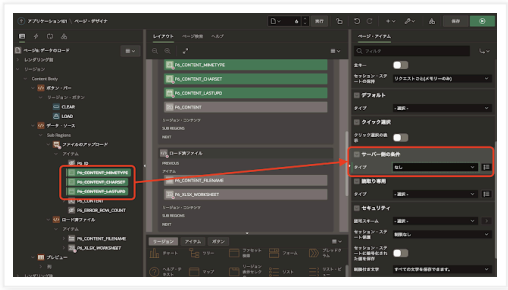
ページ・アイテム**P6_CONTNT_FILENAME**のソース定義を確認した後、**削除**します。すでに存在するP6_FILE_NAMEをP6_CONTENT_FILENAMEに変更します。



ページ・アイテム**P6_FILE_NAME**を選択し、**名前**を**P6_CONTENT_FILENAME**に変更します。**ソース**の定義として、**フォーム・リージョン**に**ファイルのアップロード**を選択します。**列**に**CONTENT_FILENAME**を選択し、**データ型**は**VARCHAR2**、**セッション・ステートの保持**は**セッションごと(ディスク)**を選択します。



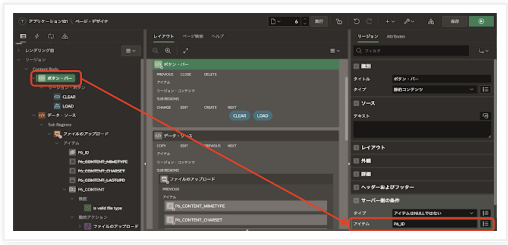
ページ・アイテム**P6_CONTENT_MIMETYPE**、**P6_CONTENT_CHARSET**、**P6_CONTENT_LASTUPD**を非表示とするため、これらのページ・アイテムをすべて選択し、**サーバー側の条件**の**タイプ**を**なし**とします。



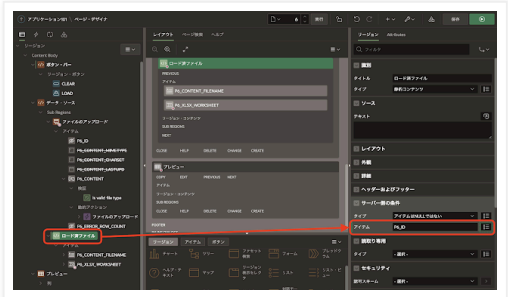
リージョンの表示条件としてP6_FILE(名称をP6_FILEからP6_CONTENTに変更済みなので、P6_CONTENTとして表示される)がNULLかどうかで判断している部分をページ・アイテムP6_IDの有無で判定するように変更します。

タイプが**ファイル参照...**のページ・アイテムで**記憶域タイプ**として**BLOB column specified in Item Source attribute**が指定されていると、ページの送信後に**セッション・ステートの保持**の設定に関わらず、値が初期化される模様です。ユーザー表に保存する場合は、主キーによってデータの有無を判断する方が適切です。

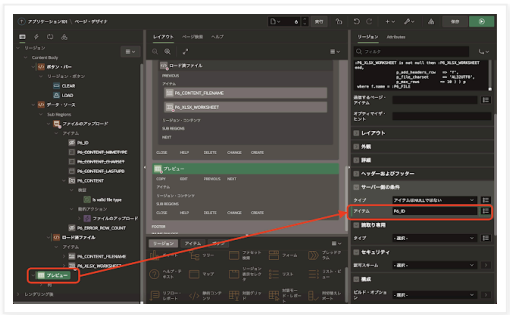
リージョンの**ボタン・バー**を選択し、**サーバー側の条件**の**アイテム**をP6_CONTENTから**P6_ID**へ変更します。



サブ・リージョンの**ロード済みファイル**に、同様の変更を行います。



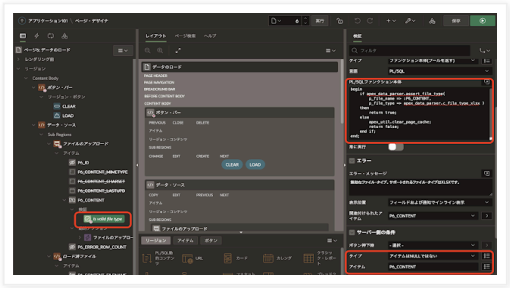
リージョンの**プレビュー**にも、同様の変更を行います。



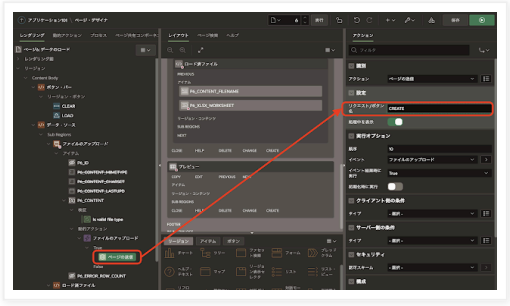
ファイル・タイプの検証に使用されるコードを更新します。以下のコードでファイルがXLSXであることを検証します。

```
begin
    if apex_data_parser.assert_file_type(
        p_file_name => :P6_CONTENT,
        p_file_type => apex_data_parser.c_file_type_xlsx )
    then
        return true;
    else
        apex_util.clear_page_cache;
        return false;
    end if;
end;
```

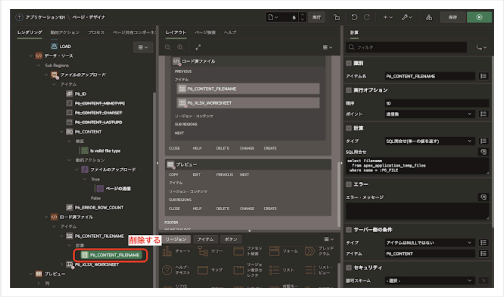
ページ・アイテムP6_CONTENTに紐づく検証**Is valid file type**を選択し、**PL/SQLファンクション本体**を上記のコードに置き換えます。また**サーバー側の条件**として、**タイプ**が**アイテムはNULLではない**、**アイテム**として**P6_CONTENT**を指定し、P6_CONTENTが指定されているときのみ検証を実施します。



動的アクション**ファイルのアップロード**のTrueアクション**ページの送信**を選択し、**リクエスト/ボタン名**に**CREATE**を設定します。プロセス**フォームの行の自動処理(DML)**は、このリクエスト/ボタン名を認識し行の挿入処理(つまりBLOBへの保存)を行います。

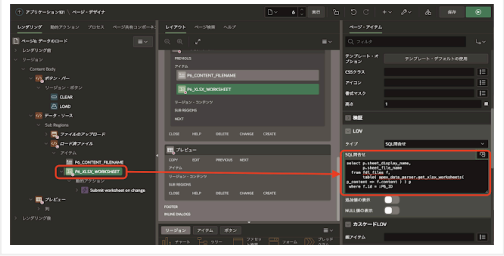


ページ・アイテム**P6_CONTENT_FILENAME**に紐づく**計算**は不要なので**削除**します。(これから追加する)プロセスによって、ファイル名は設定されます。



アップロードされたExcelのシートを選択するためのページ・アイテム**P6_XLSX_WORKSHEET**の、**LOV**を表示するための**SQL問合せ**を以下に変更します。

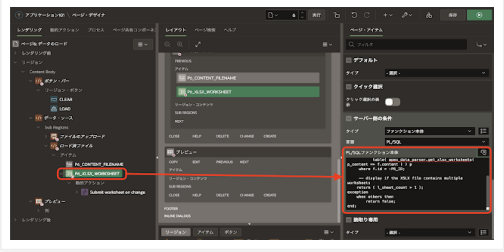
```
select p.sheet_display_name,
       p.sheet_file_name
  from fdl_files f,
       table( apex_data_parser.get_xlsx_worksheets( p_content => f.content ) ) p
 where f.id = :P6_ID
```



サーバー側の条件として設定されている**PL/SQLファンクション本体**も以下のコードに変更します。Excelに含まれているシート数が 1 より多いときのみ、シートが選択を行います。

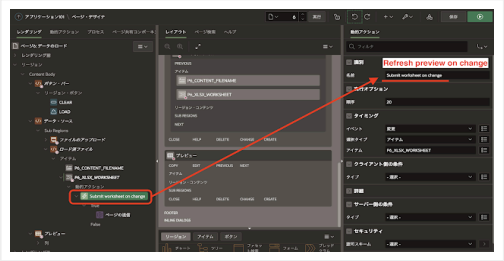
```
declare
    l_sheet_count number;
begin
    select count(*)
      into l_sheet_count
      from fdl_files f,
           table( apex_data_parser.get_xlsx_worksheets( p_content => f.content ) ) p
     where f.id = :P6_ID;

     -- display if the XSLX file contains multiple worksheets
    return ( l_sheet_count > 1 );
exception
    when others then
        return false;
end;
```

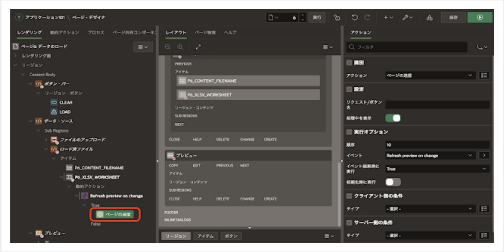


選択されたシートが変更される度にページの送信が行われるよう、動的アクションが設定されています。ファイル自体はすでにアップロード済みなので、プレビューのクラシック・レポートをリフレッシュするように変更します。

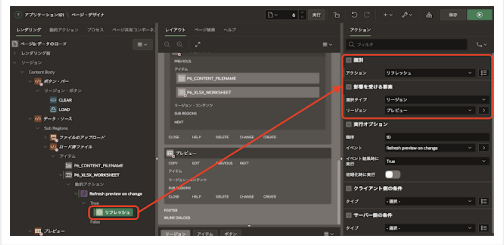
動的アクション**Submit worksheet on change**を選択し、**名前**を**Refresh preview on change**に変更します。



Trueアクションの**ページの送信**を選択します。



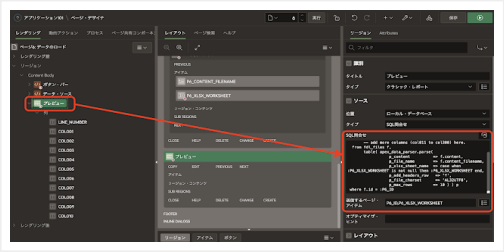
識別の**アクション**を**リフレッシュ**に変更し、**影響を受ける要素**の**選択タイプ**として**リージョン**を選択します。**リージョン**として**プレビュー**を指定します。



プレビューのソースであるSQL問合せを以下に変更します。

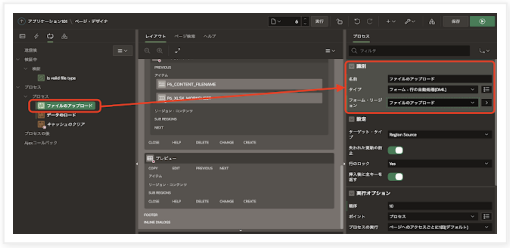
```
select p.line_number,
       p.col001, p.col002, p.col003, p.col004, p.col005, p.col006, p.col007, p.col008, p.col009, p.col010
       -- add more columns (col011 to col300) here.
  from fdl_files f,
       table( apex_data_parser.parse(
                 p_content          => f.content,
                 p_file_name        => f.content_filename,
                 p_xlsx_sheet_name  => case when :P6_XLSX_WORKSHEET is not null then :P6_XLSX_WORKSHEET end,
                 p_add_headers_row   => 'Y',
                 p_file_charset      => 'AL32UTF8',
                 p_max_rows         => 10 ) ) p
 where f.id = :P6_ID
```

リージョンの**プレビュー**を選択し、**ソース**の**SQL問合せ**を上記のコードに更新します。リフレッシュの際にSELECT文で参照しているページ・アイテムの値が送信されるよう、**送信するページ・アイテム**として**P6_ID**および**P6_XLSX_WORKSHEET**を設定します。



左ペインでプロセス・ビューを開き、新規にファイルを表FDL_FILESに保存するプロセスを作成します。

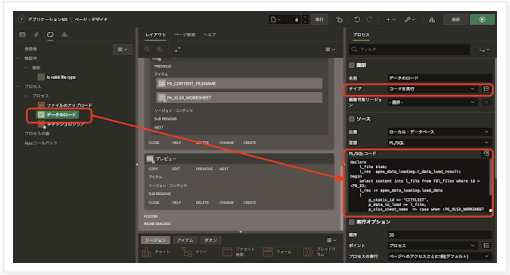
プロセスの作成を行い、**名前**を**ファイルのアップロード**とします。**タイプ**に**フォーム - 行の自動処理(DML)**を選択し、**フォーム・リージョン**として**ファイルのアップロード**を指定します。他の設定はデフォルトのままとします。



ユーザー表からのデータのロードについては、2021年5月15日の時点で不具合のため**ソース・データ型**として**SQL Query**を使えません。そのため以下のPL/SQLコードにてデータのロードを行います。

```
declare
    l_file blob;
    l_res  apex_data_loading.t_data_load_result;
begin
    select content into l_file from fdl_files where id = :P6_ID;
    l_res := apex_data_loading.load_data
    (
        p_static_id => 'CITYLIST',
        p_data_to_load => l_file,
        p_xlsx_sheet_name  => case when :P6_XLSX_WORKSHEET is not null then :P6_XLSX_WORKSHEET end
    );
    :P6_ERROR_ROW_COUNT :=  l_res.error_rows;
end;
```

プロセスの**データのロード**を選択し、**タイプ**を**コードを実行**に変更し、上記の**PL/SQLコード**を設定します。

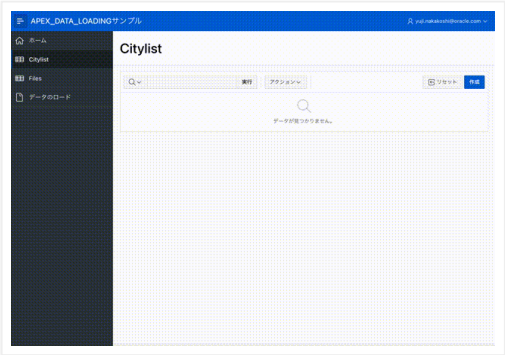


以上でアプリケーションの変更は完了です。

複数のシートを含むテスト用のExcelシートcitylist2.xlsxを以下に配置しました。

https://github.com/ujnak/apexapps/blob/master/exports/citylist2.xlsx

こちらを使用して、アプリケーションの動作を確認してみます。成功時のメッセージ出力などは調整の余地がありますが、概ねページ作成ウィザードによって作成されたページと同様な動作を行なっています。



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/apex-data-loading-sample.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: 14:17

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