

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。

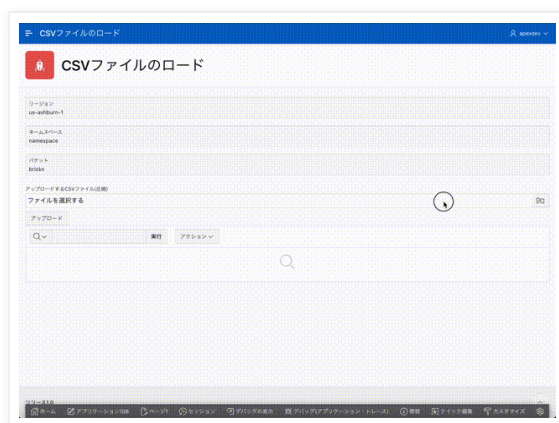
2022年4月4日月曜日

CSVファイルをオブジェクト・ストレージを介して表にインポートするアプリ

処理時間が短いDBMS_CLOUD.COPY_DATAを使って、CSVファイルをデータベースにインポートするAPEXアプリケーションを作成してみます。

[こちらの記事](#)（[とこちらの記事](#)）と同じ環境で作業を行います。

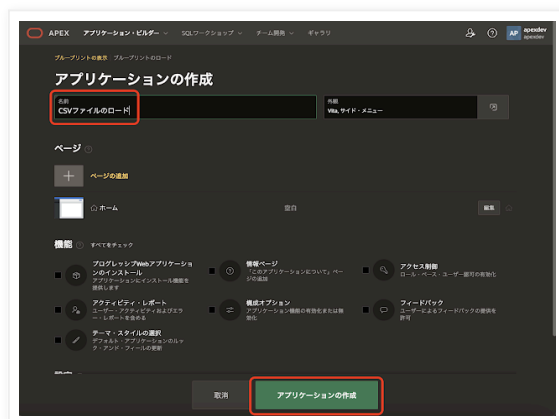
作成されるアプリケーションは以下のような動作をします。アップロードしているCSVファイルは1000万行、圧縮したサイズは41MB、非圧縮で834MBのサイズです。ファイルのアップロードから表へのインポートが完了するまで、処理時間は大体50秒です。



以下よりアプリケーションの作成手順を紹介します。

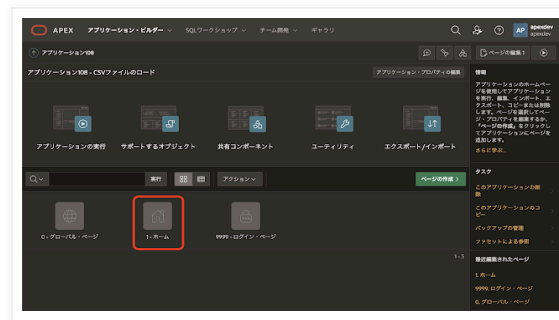
アプリケーション作成ウィザードを実行し、空のアプリケーションを作成します。

名前はCSVファイルのロードとし、アプリケーションの作成を実行します。



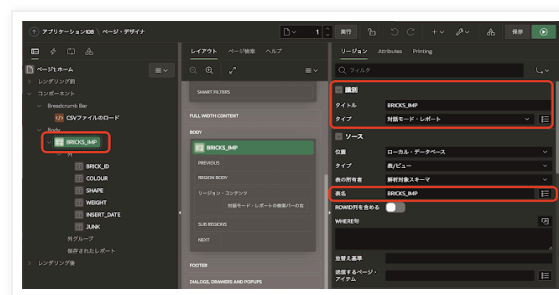
アプリケーションが作成されます。すべての機能をホーム・ページに実装します。

ページ・デザイナーにてホーム・ページを開きます。



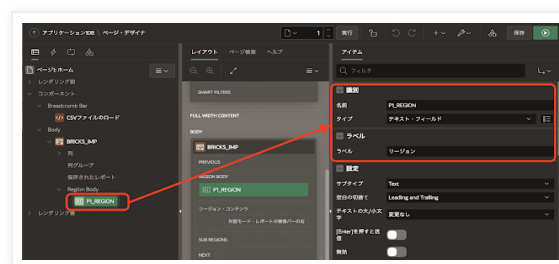
Bodyの位置に、新規にリージョンを作成します。

識別のタイトルをBRICKS_IMP、タイプとして対話モード・レポートを選択します。ソースの表名にBRICKS_IMPを指定します。

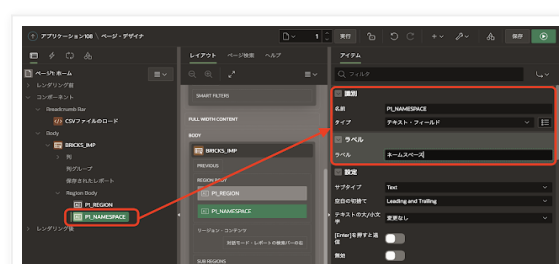


対話モード・レポートのリージョンに、オブジェクト・ストレージ上のアップロード先となる、リージョン、ネームスペースおよびバケットを指定するページ・アイテムP1_REGION、P1_NAMESPACEおよびP1_BUCKETを作成します。また、CSVファイルを選択するページ・アイテムP1_FILEも作成します。

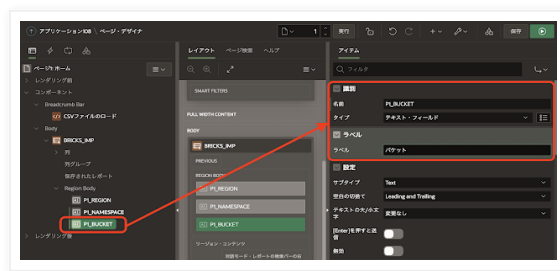
ページ・アイテムを作成し、識別の名前をP1_REGIONとします。タイプはテキスト・フィールド、ラベルはリージョンです。



同様にページ・アイテムP1_NAMESPACEを作成します。ラベルはネームスペースとします。



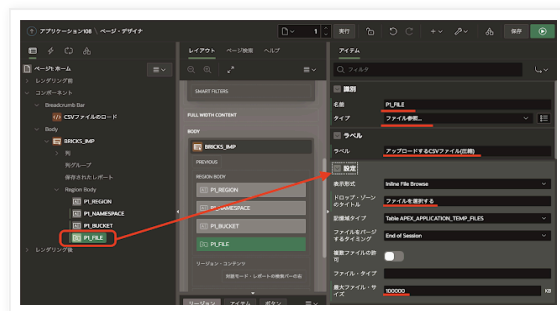
ページ・アイテムP1_BUCKETを作成します。ラベルはバケットとします。



アップロードするCSVファイルを選択するページ・アイテムP1_FILEを作成します。

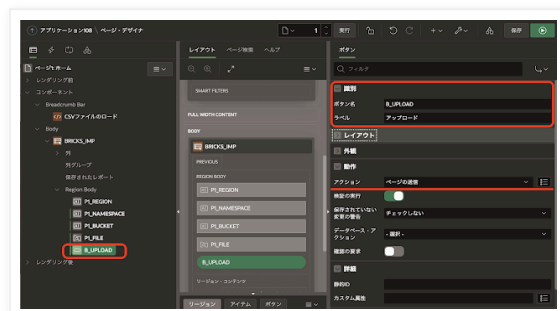
識別の名前はP1_FILE、タイプとしてファイル参照...を選択します。ラベルはアップロードするCSVファイル(圧縮)とします。

設定のドロップゾーンタイトルはファイルを選択する、記憶域タイプはデフォルトのTable APEX_APPLICATION_TEMP_FILESのままとします。APEXが提供している表を使うので、ファイルを保存するための表を用意する手間が省けます。最大ファイル・サイズは100000とします。(100MBが上限となる)



ファイルのアップロードを実行するボタンを作成します。

識別の名前はB_UPLOAD、ラベルはアップロードとします。動作のアクションはデフォルトのページの送信のままとします。



ボタンを押したときに実行されるプロセスを作成します。

左ペインでプロセス・ビューを表示させ、プロセスの作成を実行します。

識別の名前はCSVファイルのロード、タイプとしてコードの実行を選択します。ソースのPL/SQLコードには以下を記述します。

```
declare
```

```

l_blob blob;
l_filename varchar2(80);
l_path varchar2(400);
-- For UTL_FILE
l_file      utl_file.file_type;
l_buffer    raw(32767);
l_amount    binary_integer := 32767;
l_pos       integer := 1;
l_blob_len  integer;

begin
    -- APEXにアップロードされたファイルをBLOBに取り出す。
    select filename, blob_content into l_filename, l_blob
    from apex_application_temp_files
    where name = :P1_FILE;
    -- オブジェクト・ストレージ上の保存先を決める。
    l_path := 'https://objectstorage.' || :P1_REGION || '.oraclecloud.com/n/' || :P1_NAMESPACE
              || '/b/' || :P1_BUCKET || '/o/temp/' || l_filename;
    -- DATA_PUMP_DIR以下にBLOBをファイルとして書き出す。
    l_blob_len := dbms_lob.getlength(l_blob);
    l_file := utl_file.fopen('DATA_PUMP_DIR', l_filename, 'wb', 32767);
    l_pos := 1;
    l_amount := 32767;
    while l_pos <= l_blob_len
    loop
        dbms_lob.read(l_blob, l_amount, l_pos, l_buffer);
        utl_file.put_raw(l_file, l_buffer, TRUE);
        l_pos := l_pos + l_amount;
    end loop;
    utl_file.fclose(l_file);
    -- DATA_PUMP_DIR以下のファイルをオブジェクト・ストレージにアップロードする。
    dbms_cloud.put_object(
        credential_name => 'DEF_CRED'
        , object_uri => l_path
        , directory_name => 'DATA_PUMP_DIR'
        , file_name => l_filename
    );
    -- CSVをロードする表を初期化する。
    execute immediate 'truncate table bricks_imp';
    -- オブジェクト・ストレージから表にデータをロードする。
    dbms_cloud.copy_data
    (
        table_name => 'BRICKS_IMP'
        , credential_name => 'DEF_CRED'
        , file_uri_list => l_path
        , format => json_object(
            'type' value 'csv'
            , 'compression' value 'gzip'
        )
    );

```

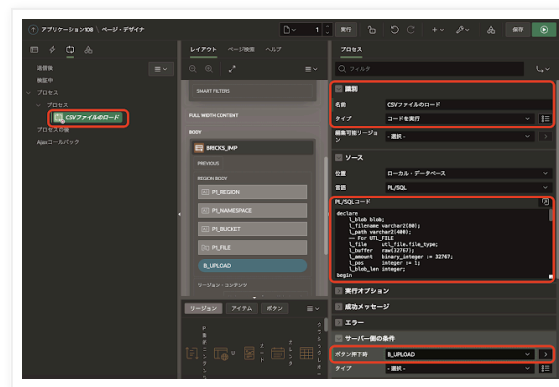
```

        , 'characterset' value 'AL32UTF8'
        , 'dateformat' value 'DD-MM-RR'
    )
);
-- DATA_PUMP_DIR以下のファイルを削除する。
dbms_cloud.delete_file(
    directory_name => 'DATA_PUMP_DIR'
    , file_name => l_filename
);
end;
```

upload_obs_and_load_csv.sql hosted with ❤ by GitHub

[view raw](#)

サーバー側の条件のボタン押下時にB_UPLOADを指定します。



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、最初のGIF動画のような動作になります。

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/csv-import-from-object-storage.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。

追記

オブジェクト・ストレージを操作するOCI PL/SQL SDKを使うと、以下のようなコードになります。

```

declare
    l_blob blob;
    l_filename varchar2(80);
    l_mime_type varchar2(200);
    l_path varchar2(400);
    -- For Object Storage PL/SQL SDK
    l_put_response dbms_cloud_oci_obs_object_storage_put_object_response_t;
    l_status_code integer;
begin
    -- APEXにアップロードされたファイルをBLOBに取り出す。
    select filename, mime_type, blob_content into l_filename, l_mime_type, l_blob
    from apex_application_temp_files
```

```

where name = :P1_FILE;
-- OCI PL/SQL SDKを使って、オブジェクト・ストレージにアップロードする。
l_put_response := dbms_cloud_oci_obs_object_storage.put_object
(
    namespace_name => :P1_NAMESPACE
    , bucket_name => :P1_BUCKET
    , object_name => 'temp/' || l_filename
    , content_type => l_mime_type
    , put_object_body => l_blob
    , region => :P1_REGION
    , credential_name => 'MY_OCI_CRED'
);
l_status_code := l_put_response.status_code;
-- OCI PL/SQL SDK呼び出し終了。
l_path := 'https://objectstorage.' || :P1_REGION || '.oraclecloud.com/n/' || :P1_NAMESPACE
        || '/b/' || :P1_BUCKET || '/o/temp/' || l_filename;
-- CSVをロードする表を初期化する。
execute immediate 'truncate table bricks_imp';
-- オブジェクト・ストレージから表にデータをロードする。
dbms_cloud.copy_data
(
    table_name => 'BRICKS_IMP'
    , credential_name => 'DEF_CRED'
    , file_uri_list => l_path
    , format => json_object(
        'type' value 'csv'
        , 'compression' value 'gzip'
        , 'characterset' value 'AL32UTF8'
        , 'dateformat' value 'DD-MM-RR'
    )
);
end;

```

upload_obs_and_load_plsq_sdk.sql hosted with ❤ by GitHub

[view raw](#)

処理時間にそれほど違いはないようです。APIキーの準備などが必要ですが、PL/SQL SDKも充実してきています。PL/SQL SDKのマニュアルは[こちら](#)になります。

完

Yuji N. 時刻: 22:27

共有

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.
