

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。

2021年7月13日 火曜日

RESTデータ・ソースを定期的に同期する

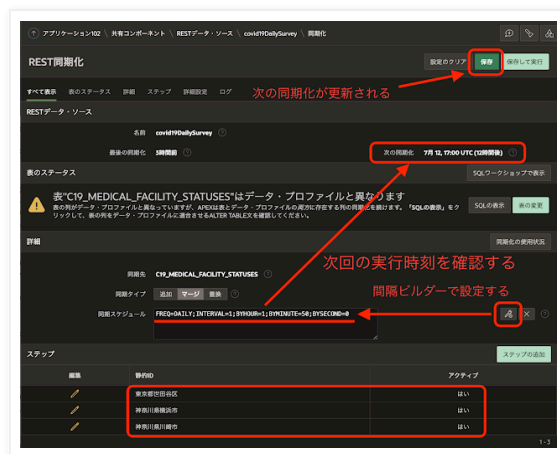
[こちらの記事](#)で作成したアプリケーションで参照している[全国医療機関の医療体制の状況](#)のオープンデータを、ローカル表に定期的に同期してみます。

RESTデータ・ソースの定期的な実装については、以前に[こちらの記事](#)で紹介しています。今回はステップバイステップの設定手順ではなく、同期化するための異なるアプローチをいくつか紹介します。

同期化は、東京都世田谷区 - 131121、神奈川県横浜市 - 141003、神奈川県川崎市 - 141305の3つの自治体を対象とします。

RESTデータ・ソースの同期化を使う

RESTデータ・ソースのREST同期化を開き、同期スケジュールとステップを定義します。



全国医療機関の医療提供体制の状況のオープンデータは午前1時頃に更新されます（[こちらの記載](#)を参照）。ですので、同期処理を午前2時に実施するよう設定します。

間隔ビルダーを開いて、頻度を日、間隔を1（頻度が日なので1日毎）、実行時間に2（午前2時）、実行分を0（0分）として実行間隔の設定を行います。その後、保存をクリックすると次の同期化が更新されます。

間隔ビルダー

このダイアログは、単純なスケジューラ・カレンダー式を作成して、同期化ジョブ実行の頻度を指定するために役立ちます。

頻度: 週 **日** 毎時 毎分 ⓘ

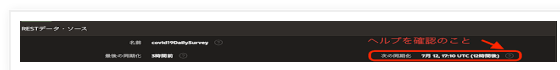
間隔: 1 ⓘ

実行時間: 2 ⓘ

実行分: 0 ⓘ

実行間隔の設定

以下は次の同期化の表示例です。7月12日の14:30頃の画面表示です。表示は7月12、17:10 UTCとなっています。日本時間としては9時間進めて、**7月13日 2:10 になります。2時ちょうどになっていません。**



次の同期化のヘルプには、以下のように記載されています。

次回スケジュールされている同期化が発生する時点を示します。同期化は、正時から開始して10分ごとに発生します。つまり、同期化は0、10、20、30、40、50の各分に開始しますが、その間には開始しません。

0分に実行される同期化は、50分(50分含む)から0分(0分含まず)の間になります。そのため、実行時間を1、実行分を50から59の間に設定すると、**次の同期化は午前2時ちょうどになります。**

同期スケジュールはサーバー側のタイムゾーンで設定します。**ヘルプの情報**から、サーバーのタイムゾーンを確認できます。



現在の時刻(サーバー上)を参照します。

Application Expressについて	
詳細	
製品のバージョン情報	21.11
スキーマ互換性	2021.04.15
パッチ・バージョン	1
最終パッチ時間	21-07-05 09:52:26
最終DDL時間	21-05-13 11:15:31
ホスト・スキーマ	APEX_PUBLIC_USER
アプリケーション所有者	APEX_210100
ワークスペースID	2600572927015410
ワークスペース名	APEXDEV
現行ユーザー	ADMIN
言語プリファレンス	ja
現在の時間(サーバー上)	21-07-12 14:43:23 +09:00

RESTデータ・ソースの同期化は、それ自体はDBMS_SCHEDULERのジョブではありません。あらかじめ10分ごとに実行されるようにスケジュールされているジョブより（1度きりのDBMS_SCHEDULERのジョブとして）実行されます。

同期化の対象とする3つの自治体はそれぞれ**ステップ**として定義します。



パラメータ名	値タイプ	式
localGovCode	数値	13121
since	PL/SQL式	to_char(sysdate-1,'YYYYMMDD')
prefName	NULL	

RESTデータ・ソースに指定できるパラメータlocalGovCodeに自治体コードを指定します。sinceは前日の日付を指定します。prefNameはlocalGovCodeが指定済みなので、NULLとします。



パラメータ名	値タイプ	式
localGovCode	数値	13121
since	PL/SQL式	to_char(sysdate-1,'YYYYMMDD')
prefName	NULL	

東京都世田谷区、神奈川県横浜市、神奈川県川崎市のステップをそれぞれ追加します。

以上で毎日午前2時に対象としている自治体の情報が更新されます。

自動化を使う

プロシージャアAPEX_REST_SOURCE_SYNC.DYNAMIC_SYNCHRONIZE_DATAを呼び出すことで、RESTデータ・ソースのパラメータを動的に指定することができます。

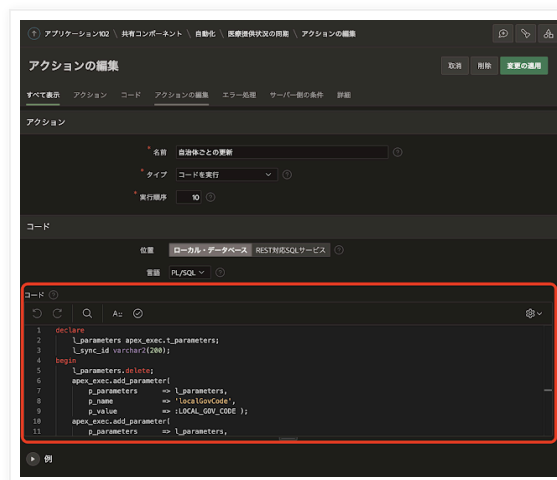
自動化のアクションとして指定するコードは、以下のようになります。バインド変数:LOCAL_GOV_CODEに自治体コードを割り当てます。

```
declare
    l_parameters apex_exec.t_parameters;
    l_sync_id varchar2(200);
begin
    l_parameters.delete;
    apex_exec.add_parameter(
        p_parameters => l_parameters,
        p_name       => 'localGovCode',
        p_value      => :LOCAL_GOV_CODE );
    apex_exec.add_parameter(
        p_parameters => l_parameters,
        p_name       => 'since',
```

```

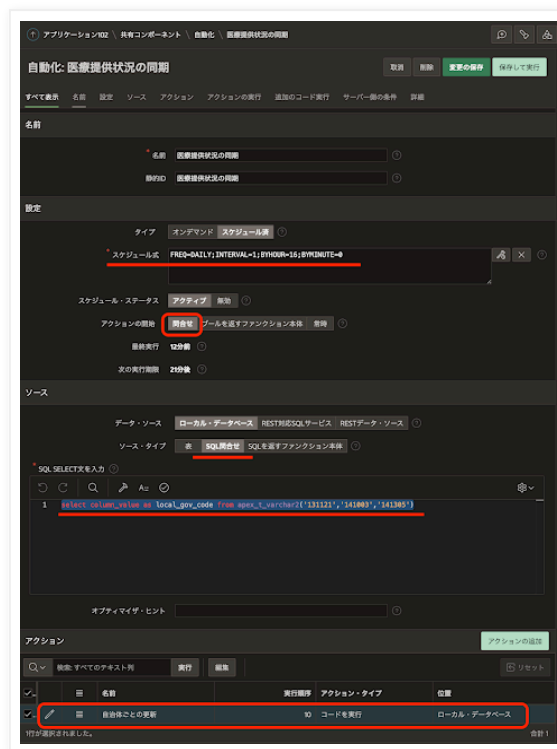
p_value          => to_char(sysdate-1,'YYYYMMDD') );
l_sync_id := '同期 - 自治体コード: ' || :LOCAL_GOV_CODE;
apex_rest_source_sync.dynamic_synchronize_data(
    p_module_static_id => 'covid19DailySurvey',
    p_sync_static_id   => l_sync_id,
    p_sync_parameters  => l_parameters );
end;

```



自動化のソースのソース・タイプをSQL問合せとし、同期化の対象とする自治体コードをSELECT文の検索結果として返します。以下が自治体コードを返すSELECT文の1例です。local_gov_codeとして3件検索されるので、自動化として登録されているアクションは3回実行されます。

```
select column_value as local_gov_code from apex_t_varchar2('131121','141003','141305')
```



自動化をコードから呼び出す

自動化をコードから呼び出すことも可能です。ソースのSQL問合せを動的に変更できます。ただし、これはソース・タイプをSQLを返すファンクション本体としても可能でしょう。

以下が自動化をコードから呼び出す一例です。ファンクション [APEX_EXEC.OPEN_QUERY_CONTEXT](#) として定義した **SQL問合せ** を自動化の呼び出しに指定します。

```
declare
    l_context apex_exec.t_context;
    l_columns apex_exec.t_columns;
    l_sql varchar2(4000);
begin
    l_sql := q'~
select column_value as local_gov_code from apex_t_varchar2('131121','141003','141305')
~';
    dbms_output.put_line(l_sql);
    apex_exec.add_column(
        p_columns => l_columns,
        p_column_name => 'LOCAL_GOV_CODE'
    );
    l_context := apex_exec.open_query_context(
        p_location => apex_exec.c_location_local_db,
        p_columns => l_columns,
        p_sql_query => l_sql
    );
    apex_automation.execute(
        p_static_id => 'syncdailysurvey',
        p_query_context => l_context
    );
end;
```

プロシージャ `APEX_AUTOMATION.EXECUTE` に自動化の静的IDを与えて呼び出すことにより、コード中より自動化を起動します。残念なことに **Oracle APEX 21.1** では指定したバインド変数に値が渡らないという不具合が見つかっています。そのため、自動化の呼び出しで `p_query_context` に `SELECT` 文を指定する使い方は避けるべきです。

以上になります。

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: 10:43

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](#)