

日々是Oracle APEX

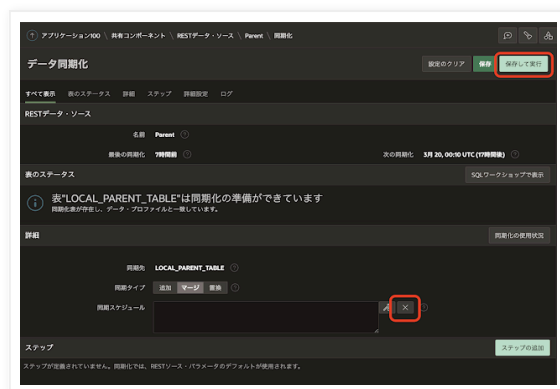
Oracle APEXを使った作業をしていて、気の付いたところを忘れないようにメモをとります。

2021年3月19日 金曜日

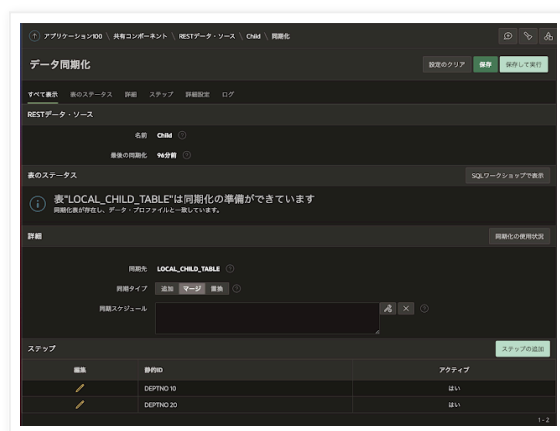
RESTデータ・ソースを使った高度な同期化(4) - 自動化による同期

[こちらの記事](#)の継続です。前回では、ボタンのクリックで表LOCAL_PARENT_TABLEおよびLOCAL_CHILD_TABLEの更新を一度に行うプロセスを作成しました。今回は、その処理を自動化を作成することにより、定期的に行います。

最初にRESTデータ・ソースの同期化の管理をそれぞれ開き、不要な設定を除きます。Parentについては、**同期スケジュール**を削除します。



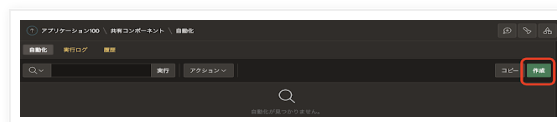
RESTデータ・ソースChildについても同様に、同期スケジュールが設定されていたら削除します。ステップの定義も不要なのですが、これはアプリケーションをエクスポートしたときの設定サンプルとするため、残しておきます。記事を読んでいる方は削除して構いません。



共有コンポーネントの自動化を開きます。



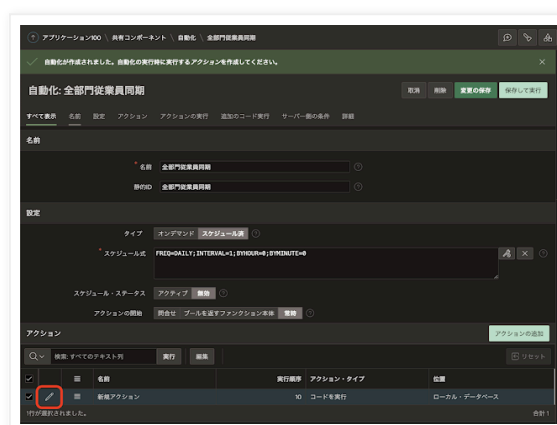
作成をクリックします。



自動化を作成します。名前を**全部門従業員同期**、タイプを**スケジュール済**、アクションの開始を**常時**、**実行スケジュール**を**毎日午前0時**とします。作成をクリックします。



自動化が作成されます。**アクション**が、あらかじめひとつ作成されます。このアクションに、実際に行う処理を定義します。**新規アクション**の鉛筆アイコンをクリックします。



同期処理を行うコードには以下を利用します。**APEX_AUTOMATION.LOG_ERROR**、**LOG_INFO**、**LOG_WARN**を使うことで自動化のログを記録することができます。**コード**を設定して、**変更の適用**をクリックします。

```
declare
  l_parameters apex_exec.t_parameters;
begin
  --
```

```

-- Step 1: Perform "Parent" Synchronization. No Parameters (empty array) are passed in.
--
apex_rest_source_sync.dynamic_synchronize_data(
    p_module_static_id => 'Parent',
    p_sync_static_id   => 'Dynamic Execution',
    p_sync_parameters  => l_parameters );

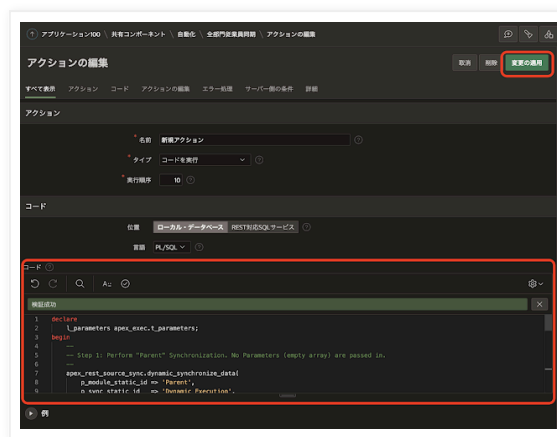
--

-- Step 2: Synchronize "Child" for all changed or new rows of the parent table.
--
for d in (
    select deptno
    from local_parent_table )
loop
    l_parameters.delete;
    apex_exec.add_parameter(
        p_parameters    => l_parameters,
        p_name          => 'deptno',
        p_value         => to_char( d.deptno ) );

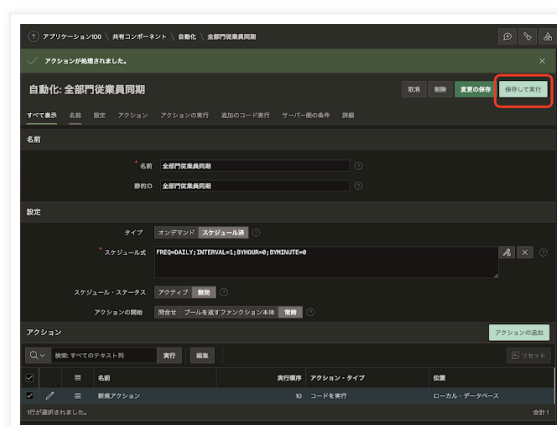
    apex_rest_source_sync.dynamic_synchronize_data(
        p_module_static_id => 'Child',
        p_sync_static_id   => 'Dynamic Deptno',
        p_sync_parameters  => l_parameters );

    -- this writes an informational message to the Automation Execution log.
    apex_automation.log_info(
        'Child REST API synchronized for DEPTNO=' || d.deptno || ' ');
end loop;
end;

```



自動化の画面で、**保存して実行**をクリックします。アクションに設定したコードが実行されます。



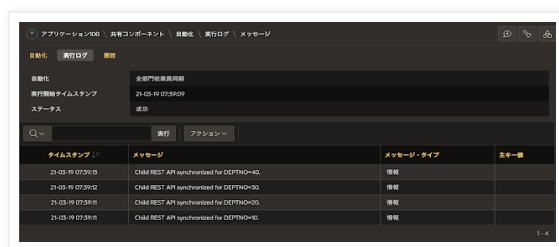
作成済みの自動化を一覧するページに戻り、**実行ログ**を開きます。



実行ログより**メッセージ**を確認します。数値をクリックします。



同期処理を行うコードに含まれているAPEX_AUTOMATION.LOG_INFOでの書き込みが、実行ログより参照できます。4つの部門、10、20、30、40について、表LOCAL_CHILD_TABLEへの同期処理がそれぞれ実施されていることが分かります。



自動化が正しく動作することが確認できたので、**スケジュール・ステータス**を無効から**アクティブ**へ変更します。実行はしないので、**変更の保存**をクリックします。



自動化のリストから、**スケジュール・ステータス**が**アクティブ**になっていることを確認します。



以上で作業は完了です。

注意として、リストに表示されている**次回の実行**ではタイムゾーンが表示されていません。SQLコマンドで、以下のようなSQLにてビューAPEX_APPL_AUTOMATIONSを検索することにより、ローカルの時刻での次回の実行時刻を確認することができます。

```
select name, polling_next_run_timestamp at time zone 'Asia/Tokyo' from apex_appl_automations;
```

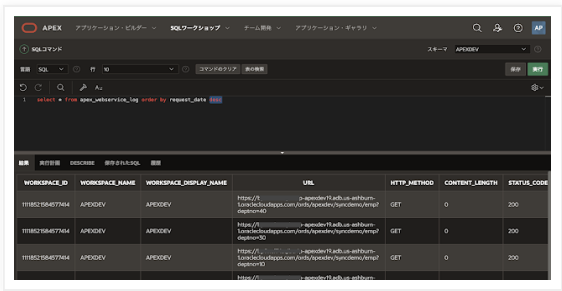
特にAutonomous Databaseなど、サーバーがUTCで動作している場合は要注意です。



RESTデータ・ソースによる同期化の設定でも自動化による同期化でも、最終的にはリモートのデータベースへREST APIによる呼び出しが行われています。どのような呼び出しが行われているかは、ビューAPEX_WEBSERVICE_LOGを検索することにより確認できます。

```
select * from apex_webservice_log order by request_date desc
```

REST API呼び出しのステータス・コードや、URLとして引数が適切に渡されているか、処理時間など、基本的な情報が記録されています。そのため、これらについてコード中で、必ずしもログとして記録する必要はありません。



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/rest-synch.sql>

Oracle APEXのアプリケーション開発の一助になれば幸いです。

完

Yuji N. 時刻: 17:20

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