

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5月30日 日曜日

RESAS APIを呼び出してデータを取得する

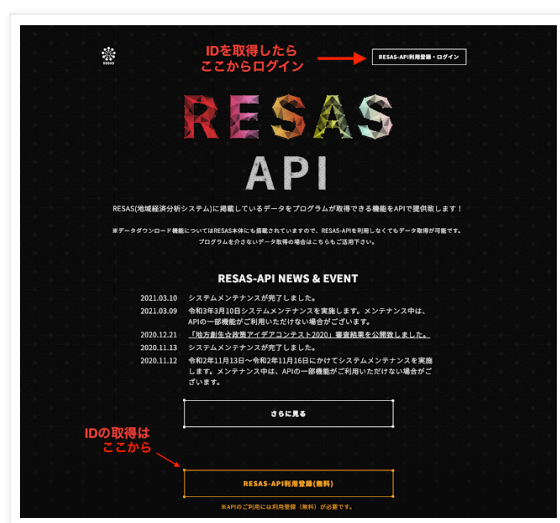
日本政府が公開しているデータを手入力する手間を省くために **RESAS API**を呼び出すことにしました。RESASには、[こちら](#)からアクセスします。ヘルプにある紹介を抜粋します。

地域経済分析システム（RESAS：リーサス）は、地方創生の様々な取り組みを情報面から支援するために、経済産業省と内閣官房（まち・ひと・しごと創生本部事務局）が提供しています。

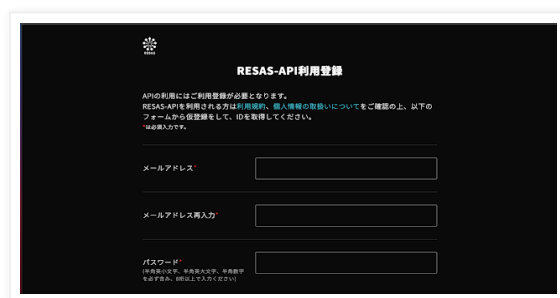
自治体職員の方や、地域の活性化に関心を持つ様々な分野の方によって、効果的な施策の立案・実行・検証のためなどに広く利用されています。

以下、Oracle APEXのアプリケーションから、RESAS APIを使って都道府県一覧を取り出し、データベースに保存するまでの作業ログです。

最初にRESAS APIの利用者登録を行ないます。RESAS APIの[トップ・ページ](#)を開きます。



RESAS-API利用者登録のフォームが開くので、必要事項を入力してIDを取得します。



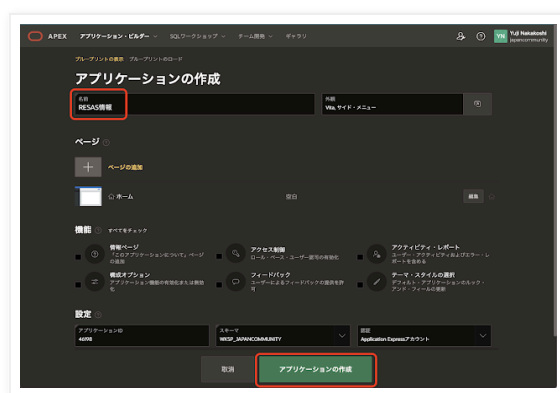
取得したIDでログインをすると、[マイページ](#)にアクセスできるようになります。



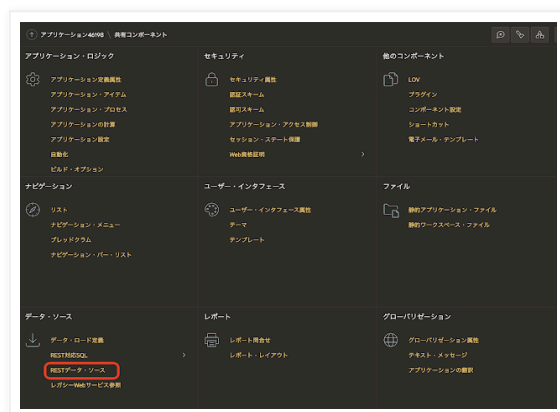
マイページからAPIキーの取得/確認ができます。



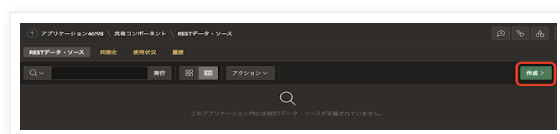
Oracle APEXのRESTデータ・ソースを使って、RESAS APIを呼び出すことにします。受け皿となるアプリケーションを作成します。**アプリケーションの作成を開始し、名前をRESAS情報とします。**他の設定はせず、そのまま**アプリケーションの作成**をクリックします。



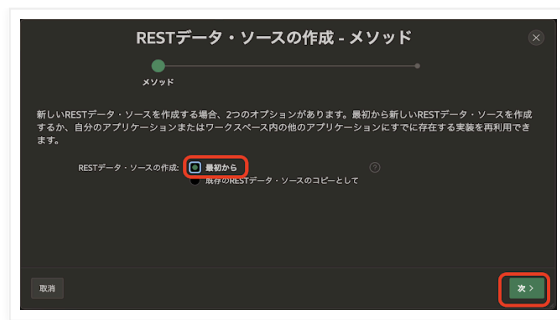
アプリケーションが作成されたら、RESTデータ・ソースを作成します。**共有コンポーネントのRESTデータ・ソース**を開きます。



RESTデータ・ソースの一覧より、**作成**をクリックします。



RESTデータ・ソースの作成として最初からを選んで、次へ進みます。

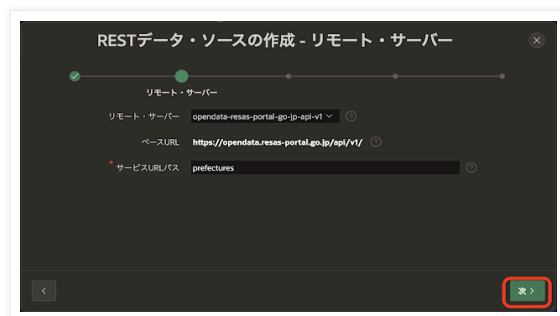


今回はRESAS APIを呼び出して都道府県一覧を取得します。参照したAPIの仕様は[こちら](#)です。

RESTデータ・ソース・タイプに簡易HTTPを選択します。名前は**RESAS - 都道府県一覧**としました。URLエンドポイントに<https://opendata.resas-portal.go.jp/api/v1/prefectures>を入力し、次へ進みます。



リモート・サーバー、ベースURLおよびサービスURLパスは変更不要です。確認だけを行い、次へ進みます。



認証が必要です。をONにします。APIキーの設定方法はRESAS [API詳細仕様](#)に記載されている通り、HTTPのリクエスト・ヘッダーとしてX-API-KEYを追加し、値としてAPIキーを指定します。

資格証明は- 新しい資格証明の入力 -を選択します。**認証タイプ**は**HTTPヘッダー**、**名前**は**X-API-KEY**です。**値**に取得済みのAPIキーを入力します。**検出**をクリックします。



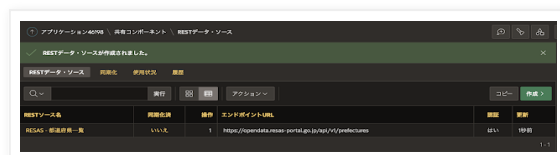
REST APIが発行され、取得されたデータが表示されます。データ・プロファイルも確認しておきます。



PREFCODEがNUMBER型、PREFNAMEがVARCHAR2(4000)として認識されていることが確認できます。RESTデータ・ソースの作成を実行します。



RESTデータ・ソースが作成されました。



作成されたRESTデータ・ソースの内容を確認するため、レポートのページを作成します。ページ作成ウィザードを開始し、レポートをクリックします。



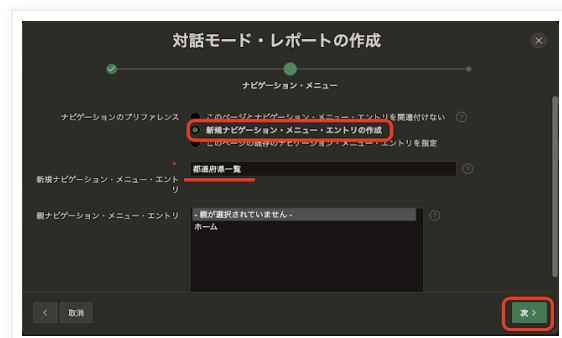
対話モード・レポートをクリックします。



ページ名は都道府県一覧とします。次へ進みます。



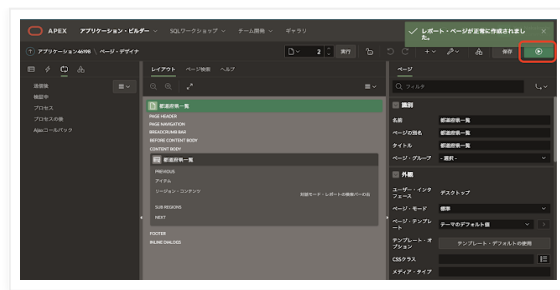
ナビゲーションのプリファレンスとして、新規ナビゲーション・メニュー・エントリの作成を選択し、次へ進みます。



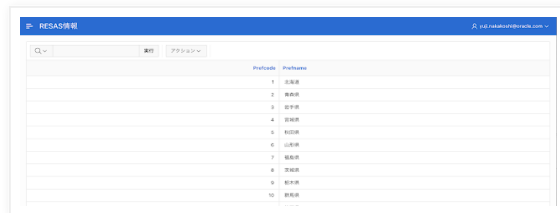
データ・ソースとしてRESTデータ・ソースを選択し、RESTデータ・ソースに先程作成したRESAS - 都道府県一覧を指定します。作成をクリックします。



ページが作成されたら、それを実行して結果を確認します。



47都道府県の情報がリストされます。



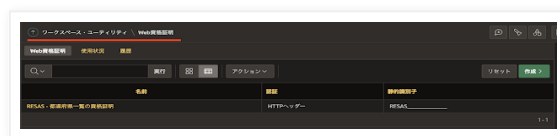
RESAS APIを呼び出すことで、都道府県一覧を取得できることが確認できました。

今まで作成した設定を再利用しやすくするために、設定内容を調整します。

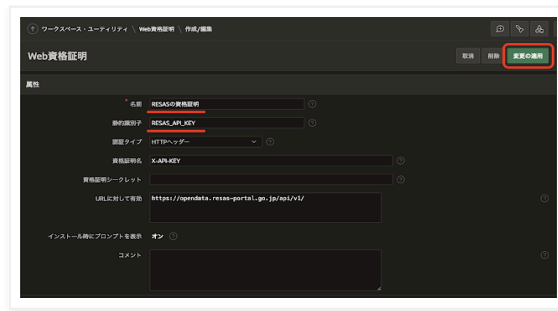
最初に共有コンポーネントのWeb資格証明を開きます。



作成済みのWeb資格証明が一覧されます。作成されたWeb資格証明はワークスペースで共有されるため、実際には**ワークスペース・ユーティリティのWeb資格証明**が開いています。また、このアプリケーションだけではなく、**ワークスペース内で作成済みのWeb資格証明が一覧されます**。作成したWeb資格証明を開きます。



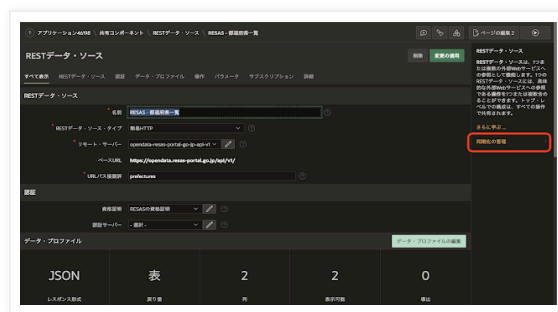
作成されている資格証明は都道府県一覧だけではなく、RESAS API全般で有効であるため、**名前をRESASの資格証明**に変更します。また、**静的識別子はRESAS_API_KEY**とします。この資格証明は**URLに対して有効**に設定されている<https://opendata.resas-portal.go.jp/api/v1/>以下へのアクセスに使うことができます。**変更の適用**をクリックします。



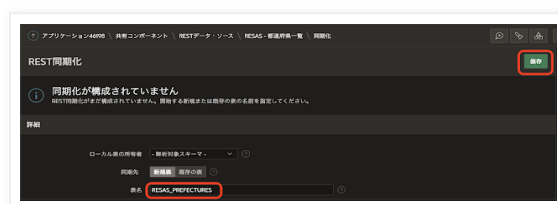
Web資格証明の参照は内部IDに基づいているため、名前や静的識別子の変更の影響はありません。PL/SQLコード内でWeb資格証明の静的識別子を参照している場合 (APEX_WEB_SERVICE.MAKE_REST_REQUESTなどの呼び出しの引数として)は、参照しているコードの変更が必要になります。

今回は都道府県のコードと名前をAPI経由で取得していますが、これは変更が発生しない情報です。ですので、ローカルの表にコピーし、そちらを参照することによりREST API呼び出しを行わない様にします。

共有コンポーネントのRESTデータ・ソースを開き、作成したRESTデータ・ソースの同期化を設定します。同期化の管理を実行します。



同期先を新規表とし、表名としてRESAS_PREFECTURESを指定します。保存をクリックします。



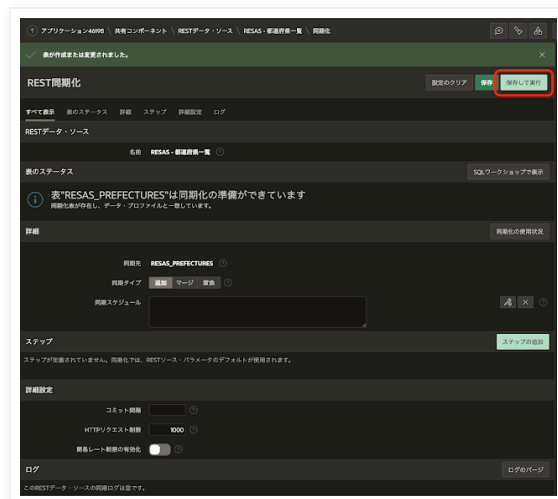
ここで指定した同期化表を新たに作成します。表の作成をクリックします。

表はRESTデータ・ソースとして検出されたデータ・プロファイルをもとに定義されます。SQLの表示をクリックすると、作成される表のDDLを確認できます。主キー制約やNOT NULL制約の追加など、あらかじめ設定した表を先に作成して、同期化表として指定することもできます。RESTデータ・ソースの同期に影響のない範囲であれば、ここで新規に作成された表の定義を後から変更することもできます。

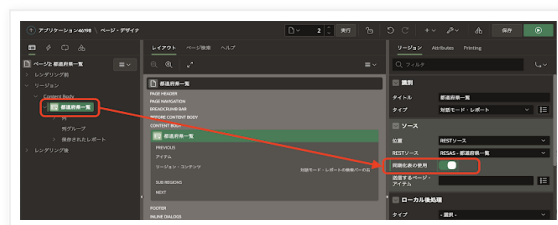
```
create table "RESAS_PREFECTURES" (
  "PREFCODE"          NUMBER
, "PREFNAME"          VARCHAR2(4000)
, "APEX$SYNC_STEP_STATIC_ID"  VARCHAR2(255)
, "APEX$ROW_SYNC_TIMESTAMP"  TIMESTAMP WITH TIME ZONE
)
```



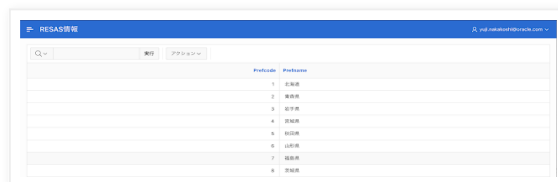
表が作成されたら、**保存して実行**をクリックします。都道府県一覧が変わることはないため、定期的な同期処理は不要です。保存と実行で行われる一回だけの処理でデータのコピーは完了します。



同期化された内容を確認するため、先程作成した都道府県一覧のレポートのページを開きます。対話モード・レポートの**ソースの同期化表の使用**を**ON**に変更します。



ページを実行し、レポートの内容に変化が無いことを確認します。



RESAS APIを呼び出してデータを取得する方法の紹介は以上になります。

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/resas-api.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。

ちなみに登録したWeb資格証明の資格証明シークレットはエクスポートの対象からは除外されます。ですので、アプリケーションをインポートした後に自分自身で取得したRESAS APIのAPIキーを登録する必要があります。

完

Yuji N. 時刻: 12:35

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.
