

# 日日は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1月21日 木曜日

## リモート・データへのアクセス (1) - アクセス先の準備

AshburnにあるAutonomous Databaseのインスタンスに東京のAutonomous Databaseからアクセスする必要があったので、REST APIを使ってデータの同期をとるように構成してみました。その作業ログです。

両方ともAlways FreeのAutonomous Transaction Processingですが、AshburnのものはDBのバージョンが21c、東京は19cです。両方ともにAPEXのワークスペースとしてAPEXDEVを構成しています。

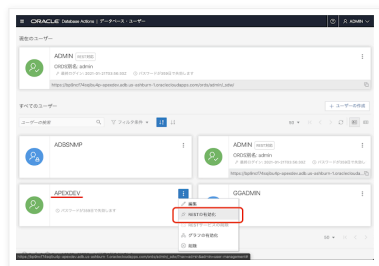
色々な方法はあるのですが、できるだけGUIを使うように試んでいます。

### Ashburn側のスキーマAPEXDEVのREST API有効化

SQL Developer WebにADMINでサインインし、**管理**に含まれる**データベース・ユーザー**を開きます。



データベース・ユーザーの一覧より、Oracle APEXのワークスペースに紐づいているスキーマ、今回はAPEXDEVですが、その**RESTの有効化**を開きます。



**承認が必要**をONにし、**REST対応ユーザー**をクリックします。



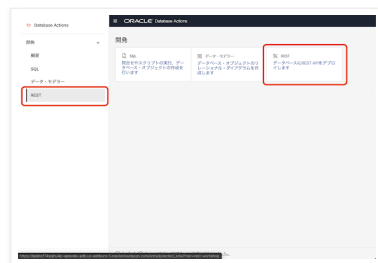
これでスキーマAPEXDEVのREST操作が有効化されます。

**URLマッピング・タイプ**は**BASE\_PATH**がデフォルトで、通常は変更の必要はありません。詳しくはマニュアルの[こちら](#)に説明があります。**スキーマ別名**の指定によってURLに現れる文字列を変更することができます。安全性を考慮すると、実際のスキーマとは異なる名前にすることが推奨です。今回は検証なので、変更は小文字**apexdev**にしているだけです。

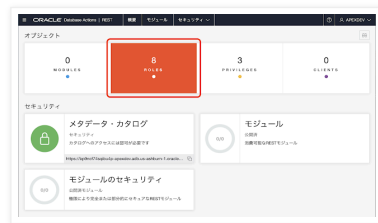
**データベース・ユーザーAPEXDEV**に上記の変更を行うことで、**APEXDEV**にて**SQL Developer Web**にサインインできるようになります。逆に言うと、この変更を行わないと、どのようなデータベース・ユーザーでもSQL Developer Webにサインインできません。

### OAuthクライアントの作成

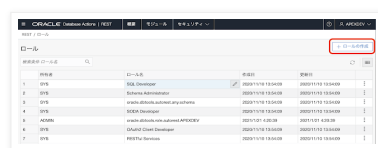
ADMINからサインアウトし、APEXDEVでSQL Developer Webにサインインし直します。トップ・ページより**REST**を開きます。



概要のページより、**ROLES**のタイルをクリックします。本来必要ないのですが、SQL Developer WebからOAuthクライアントを作成する際に必ずロールのアサインが必要なので、ダミーのロールを作成します。



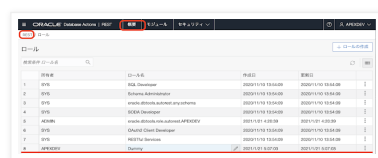
ロールの作成をクリックします。



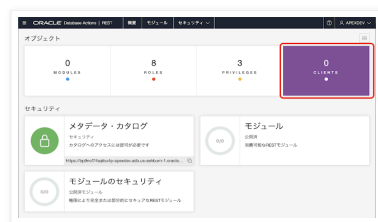
ロール名をとりあえずDummyとして、ロールを作成します。



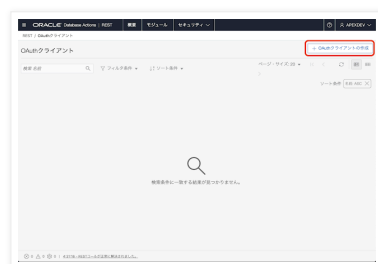
Dummyという名前のロールが作成されていることを確認します。左上のREST、もしくはメニューから概要をクリックし、概要のページに戻ります。



CLIENTSのタイルをクリックします。



OAuthクライアントの作成をクリックします。



クライアント定義として、名前を設定します。ここではapexdevとしています。それ以外の説明、サポートURI、サポート電子メールは必須項目になっていますが、OAuthの動作に影響がある設定ではないので、適当と思われる値を設定します。(クライアントを作

成するAPIのOAUTH.CREATE\_CLIENTでは、これらの値の指定は任意です）。

ロールのタブをクリックして開き、先ほど作成したDummyを選択します。その後、作成をクリックします。

OAuthクライアントが作成されたことを確認します。作成されたOAuthクライアントapexdevのクライアントID、および、クライアント・シークレットは目のアイコンをクリックすることで表示できます。

## SQL Developerロールの割り当て

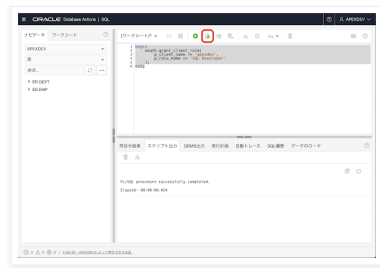
REST対応SQLを利用可能にするため、SQL DeveloperロールをOAuthクライアントapexdevに割り当てます。この作業は今のところ、GUIから行うことができません。

サイド・メニューを開き、SQLを実行するページを開きます。

以下のコードを実行し、OAuthクライアントapexdevにロールSQL Developerを割り当てます。

```
begin
  oauth.grant_client_role(
    p_client_name => 'apexdev',
    p_role_name => 'SQL Developer'
  );
end;
```

スクリプトの実行アイコンをクリックし、上記のコードを実行します。



スクリプト出力にPL/SQL procedure successfully completedと表示され、プロシージャの実行が成功していることを確認します。

以上でREST対応SQLが利用できるようになりました。

## 動作確認

REST対応SQLを実行し、動作の確認を行います。OAuthクライアントのページを開き、OAuthクライアントapexdevのハンバーガー・メニューを開き、Bearerトークンの取得を実行します。



新規トークンの取得をクリックし、現在のトークンを更新します。現在のトークンが有効期限内であり、権限などに変更がなければ、そのまま利用することもできます。現在のトークンをクリップボードなどにコピーしておきます。



または、端末のトークンを取得するには、次のコードを使用します。として記載されているcurlコマンドを実行することで、コマンドラインよりトークンを取得することも可能です。どちらかの情報をコピーし、OKをクリックしてダイアログを閉じます。

Autonomous DatabaseへアクセスするURLを確認し、以下のcurlコマンドを実行します。

```
curl -i -X POST -H"Authorization: Bearer 現在のトークンの値" --data-binary "select * from emp" -H"Content-Type: application/sql" -k https://ADBのURL.oraclecloudapps.com/or
```

select \* from empの結果が正しく返ってきたら、確認完了です。スキーマAPEXDEVに表EMPを用意していないときは、select sysdate from dualを代わりに実行してもよいでしょう。その場合の実行結果は以下のようになります。

```
HTTP/1.1 200 OK
Date: Thu, 21 Jan 2021 06:13:44 GMT
Content-Type: application/json
Transfer-Encoding: chunked
Connection: keep-alive
X-Frame-Options: SAMEORIGIN
```

```
{"env":{"defaultTimeZone":"UTC"},"items":
[{"statementId":1,"statementType":"query","statementPos":
{"startLine":1,"endLine":2},"statementText":"select sysdate from dual","resultSet":{"metadata":
[{"columnName":"SYSDATE","jsonColumnName":"sysdate","columnTypeName":"DATE","precision":0,"scale":
:0,"isNullable":1}],"items":[{"sysdate":"2021-01-
21T06:13:44Z"}],"hasMore":false,"limit":10000,"offset":0,"count":1},"response":[],"result":0}]}
```

ソース側のデータベースの設定ができたので、次からは、それにアクセスするOracle APEXのアプリケーションを定義します。

続く

Yuji N. 時刻: 15:32

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](#)