

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29日 木曜日

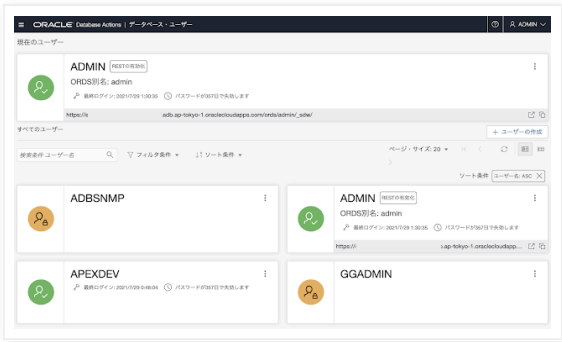
Oracle REST Data ServicesでのSQLの実行

以前に[Oracle APEXによるSQLの実行](#)という題で、Oracle APEXではSQLがどのように実行されているか紹介しています。[Oracle REST Data ServicesでのSQLの実行はAPEXとは異なり、Oracle Databaseのプロキシ接続を使っています。](#)

以下、確認作業のログになります。

最初にAutonomous Databaseで確認します。Always FreeのAutonomous Transaction Processingのインスタンスにユーザー**APEXDEV**が作成されています。

データベース・アクションのデータベース・ユーザーを開いて確認します。



RESTの有効化がされているのはユーザー**ADMIN**のみです。

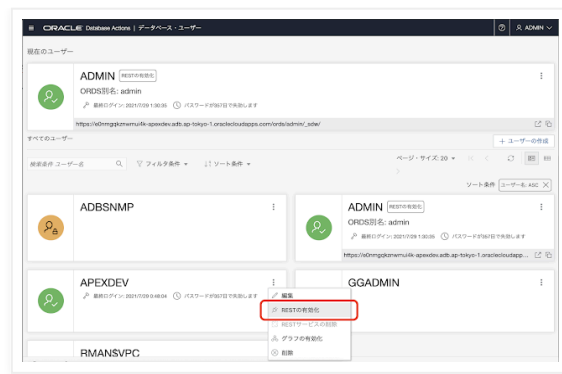
ビュー**PROXY_USERS**を確認します。

```
select * from proxy_users
```

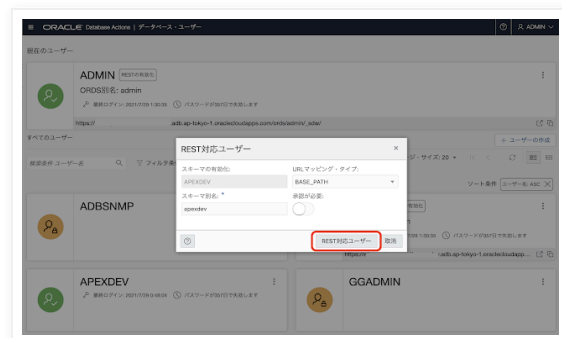
PROXY	CLIENT	AUTHENTICATION	FLAGS
C##CLOUD\$SERVICE	ADMIN	NO	PROXY MAY ACTIVATE ALL CLIENT ROLES
ORDS_PUBLIC_USER	ADMIN	NO	PROXY MAY ACTIVATE ALL CLIENT ROLES
OMLMOD\$PROXY	OML\$MODELS	NO	PROXY MAY ACTIVATE ALL CLIENT ROLES
ORDS_PUBLIC_USER	ORDS_PLSQL_GATEWAY	NO	PROXY MAY ACTIVATE ALL CLIENT ROLES
ADMIN	RMAN\$VPC	NO	PROXY MAY ACTIVATE ALL CLIENT ROLES

Autonomous Databaseでの**プロキシ・ユーザー**の初期状態です。

ユーザー**APEXDEV**の**RESTの有効化**を実行します。



ポップアップされるダイアログのREST対応ユーザーをクリックします。



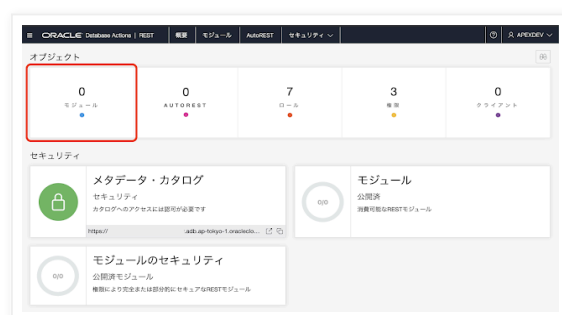
再度ビューPROXY_USERSを確認します。ORDS_PUBLIC_USERをプロキシとして、ユーザーAPEXDEVにて接続できるようになっています。

PROXY	CLIENT	AUTHENTICATION	FLAGS
C##CLOUD\$SERVICE	ADMIN	NO	PROXY MAY ACTIVATE ALL CLIENT ROLES
ORDS_PUBLIC_USER	ADMIN	NO	PROXY MAY ACTIVATE ALL CLIENT ROLES
OMLMOD\$PROXY	OML\$MODELS	NO	PROXY MAY ACTIVATE ALL CLIENT ROLES
ORDS_PUBLIC_USER	ORDS_PLSQL_GATEWAY	NO	PROXY MAY ACTIVATE ALL CLIENT ROLES
ADMIN	RMAN\$VPC	NO	PROXY MAY ACTIVATE ALL CLIENT ROLES
ORDS_PUBLIC_USER	APEXDEV	NO	PROXY MAY ACTIVATE ALL CLIENT ROLES

RESTの有効化はプロシージャ[ORDS_ADMIN.ENABLE_SCHEMA](#)を呼び出しています。この中で行われている処理のひとつとして、プロキシ接続の有効化が行われています。

簡単なRESTサービスを実装して、RESTサービスを実行しているセッションの情報を確認してみます。

データベース・アクションにユーザーAPEXDEVでサインインし、RESTを開きます。モジュールから作成します。

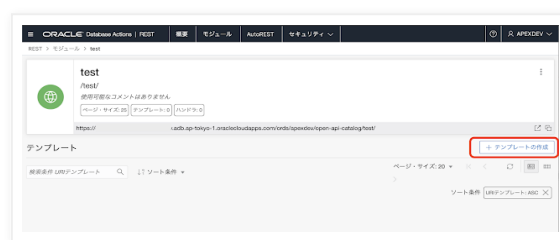


左上にあるモジュールの作成をクリックします。



モジュール名をtest、ベース・パスを/test/として、モジュールを作成します。公開はONにします。モジュール名、ベース・パスはtestでなくてもかまいません。

モジュールが作成されたら、続いてテンプレートの作成を実行します。



URIテンプレートをsessionとし、作成をクリックします。URIテンプレートについても、sessionでなければいけない、ということはありません。

テンプレートの作成

モジュール名
test

ベース・パス
/test/

URIテンプレート *
session

プレビューURL
https://
/adb.ap-tokyo-1.oraclecloudapps.com/ords/apexdev/test/s

優先度
0

HTTPエンティティ・タグ・タイプ
セキュア・ハッシュ

コメント

☒ 作成後ハンドラに移動

作成 取消

テンプレートが作成されたら、**ハンドラの作成**を実行します。

session

ハンドラ

ハンドラを作成

ハンドラの**ソース**として、以下のSELECT文を指定します。

```
select  
sys_context('userenv','session_user') as session_user,  
sys_context('userenv','session_schema') as session_schema,  
sys_context('userenv','current_schema') as current_schema,  
sys_context('userenv','proxy_user') as proxy_user  
from dual
```

メソッドはGET、ソース・タイプには**収集問合せ**を選択します。

ハンドラの作成

ハンドラ定義 使用可能なMIME

モジュール名
test

完全なURL
https://v.adb.ap-tokyo-1.oraclecloudapps.com/ords/apexdev/test/s

メソッド * ページ当たりのアイテム数
GET 25

ソース・タイプ
収集問合せ

ソース *

```

1 select
2   sys_context('userenv','session_user') as session_user,
3   sys_context('userenv','session_schema') as session_schema
4   sys_context('userenv','current_schema') as current_schema
5   sys_context('userenv','proxy_user') as proxy_user
6 from dual

```

コメント

作成 取消

以上でOracle REST Data Sources側の設定は完了です。実際に呼び出して結果を確認してみます。

ORACLE Database Actions | REST | 検索 | モジュール | AutoREST | セキュリティ

REST | モジュール | test | session | GET

session
最終更新: 1秒前
変更可能なコメントはありません
ソース・タイプ (jsoncollection) ページ・サイズ (25)

https://v.adb.ap-tokyo-1.oraclecloudapps.com/ords/apexdev/test/session

ソース RESTサービス呼び出す

```

1 select
2   sys_context('userenv','session_user') as session_user,
3   sys_context('userenv','session_schema') as session_schema,
4   sys_context('userenv','current_schema') as current_schema,
5   sys_context('userenv','proxy_user') as proxy_user
6 from dual

```

ダウンロード

RESTサービス呼び出した結果です。

```

{
  "items": [
    {
      "session_user": "APEXDEV",
      "session_schema": "APEXDEV",
      "current_schema": "APEXDEV",
      "proxy_user": "ORDS_PUBLIC_USER"
    }
  ],
  "hasMore": false,
  "limit": 25,
  "offset": 0,
  "count": 1,
  "links": [ ]
}

```

proxy_userがORDS_PUBLIC_USERになっていることが確認できます。

Autonomous Databaseでは、データベースの利用者はORDS_PUBLIC_USERを使ってデータベースには接続できません。プロキシ・ユーザーによる接続方法の参考として、オンプレミスの環境でsqlplusを使って接続してみます。

```
SQL> connect ords_public_user[apexdev]/*****@localhost/xepdb1.world
Connected.
SQL> select
sys_context('userenv','session_user') as session_user,
sys_context('userenv','session_schema') as session_schema,
sys_context('userenv','current_schema') as current_schema,
sys_context('userenv','proxy_user') as proxy_user
from dual 2 3 4 5 6
7 /
```

SESSION_USER	SESSION_SCHEMA	CURRENT_SCHEMA	PROXY_USER
APEXDEV	APEXDEV	APEXDEV	ORDS_PUBLIC_USER

SQL>

ユーザーの指定のords_public_user[apexdev]/*****の部分プロキシ接続の指定方法です。パスワードはユーザーORDS_PUBLIC_USERのものでAPEXDEVではありません。ユーザーAPEXDEVの代わりにORDS_PUBLIC_USERを使っているため、プロキシ(代理という意味) 接続になります。

プロキシ接続を許可するコマンドは以下になります。

```
ALTER USER APEXDEV GRANT CONNECT THROUGH ORDS_PUBLIC_USER;
```

GRANTの代わりにREVOKEを使うと、プロキシ接続の許可が解除されます。

Oracle APEXとデータベースへの接続方法が異なるため、Oracle REST Data Servicesでは使用するコネクション・プールを分けています。APEX向けのコネクション・プールはapex.xml、Oracle REST Data ServicesのRESTサービス向けはapex_pu.xmlが、コネクション・プールの構成ファイルになります。

APEXでもORDSでも、アプリケーションを開発している時点で接続方法の違いを意識することは無いかと思います。どちらも指定したユーザー(今回の場合ではAPEXDEV)の権限でSQLは実行されます。とはいえ、頭の片隅にでも入れていただき、障害が発生したときなどに役立ていただけると幸いです。

完

Yuji N. 時刻: 14:52

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](#)

