

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。

2022年8月3日 水曜日

CKEditor5による画像アップロードを実装する(2) - REST APIの作成

CKEditor5にて画像の貼り付けを行なったときに発行される、画像のアップロード要求を受け付けて表CKE_IMAGESに保存する処理と、画像のダウンロード要求を受け付けるREST APIを作成します。

作成するREST APIは、以下の記事で紹介しているREST APIとほとんど同じです。APEXセッションによる保護の設定についても同じです。

Oracle REST Data ServicesのマニュアルにあるImage Galleryのサンプル・アプリを作成する

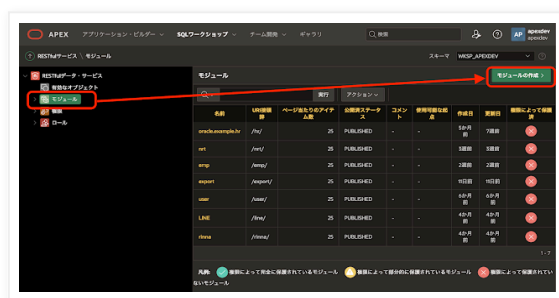
以下より、今回作成しているアプリケーションのためのREST APIを作成します。

記述する作業は、画像のアップロードとダウンロードに関するものに限定します。それ以外の作業は、上記の記事が参考になります。

画像のアップロードを受け付けるAPIを作成する

SQLワークショップのRESTfulサービスを開きます。

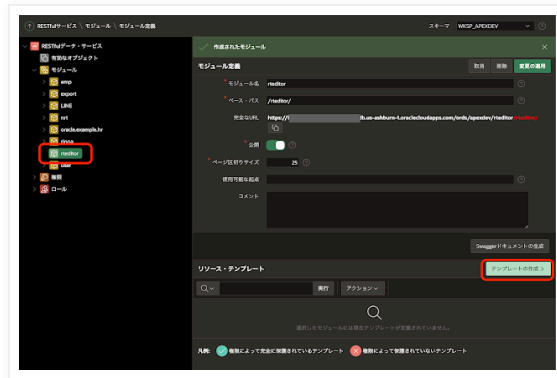
左ペインにて**モジュール**を選択し、**モジュールの作成**を実行します。



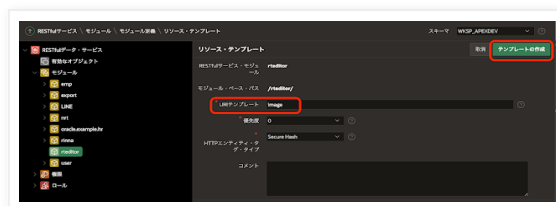
モジュール名としてrteeditor、**ベース・パス**として/rteeditor/を設定し、**モジュールの作成**を実行します。本番環境で使用するといったものではないため、**公開**はONにしておきます。



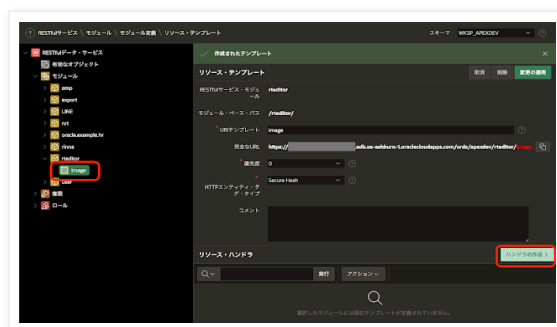
モジュールrtditorが作成されます。続けて、**テンプレートの作成**を実行します。



URIテンプレートとしてimageを入力します。**テンプレートの作成**を実行します。



テンプレートimageが作成されます。**ハンドラの作成**を実行します。



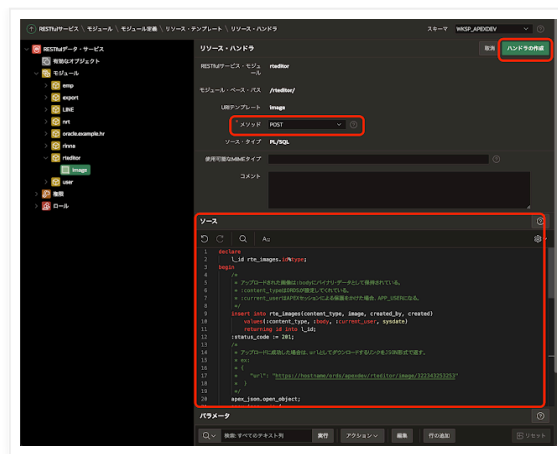
メソッドにPOSTを選択し、ソースに以下を記述します。

```
declare
    l_id rte_images.id%type;
begin
    /*
     * アップロードされた画像は:bodyにバイナリ・データとして保持されている。
     * :content_typeはORDSが設定してくれている。
     * :current_userはAPEXセッションによる保護をかけた場合、APP_USERになる。
     */
    insert into rte_images(content_type, image, created_by, created)
```

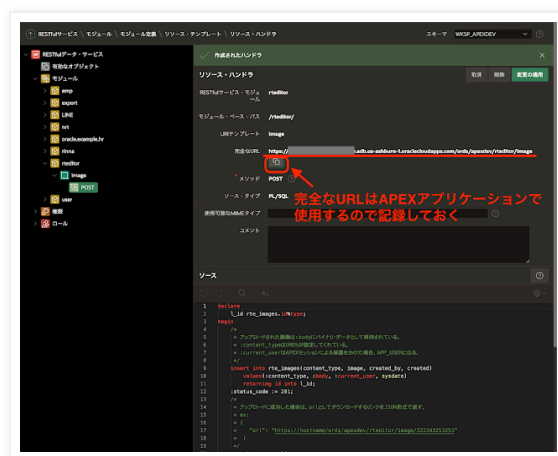
```

        values(:content_type, :body, :current_user, sysdate)
        returning id into l_id;
:status_code := 201;
/*
 * アップロードに成功した場合は、urlとしてダウンロードするリンクをJSON形式で返す。
 * ex:
 * {
 *   "url": "https://hostname/ords/apexdev/rtditor/image/322343253253"
 * }
 */
apex_json.open_object;
apex_json.write(
    p_name => 'url',
    p_value => owa_util.get_cgi_env('REQUEST_PROTOCOL') ||
        '://' ||
        owa_util.get_cgi_env('SERVER_NAME') ||
        owa_util.get_cgi_env('SCRIPT_NAME') ||
        owa_util.get_cgi_env('PATH_INFO') ||
        '/' || l_id
);
apex_json.close_object;
exception
when others then
    /* エラーを返す。
    * {
    *   "error": {
    *     "message": "Something wrong!"
    *   }
    * }
    */
    apex_json.open_object;
    apex_json.open_object('error');
    apex_json.write('message', 'Something wrong!');
    apex_json.close_object;
    apex_json.close_object;
end;
```

ハンドラの作成を実行します。



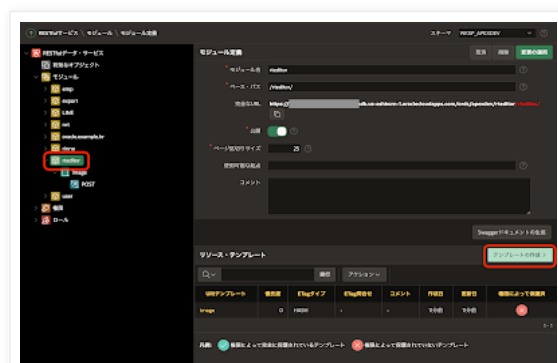
ハンドラが作成されます。完全なURLはAPEXアプリケーションで使用するため、記録しておきます。



以上で画像をアップロードするAPIの実装は完了です。

画像をダウンロードするAPIを作成する

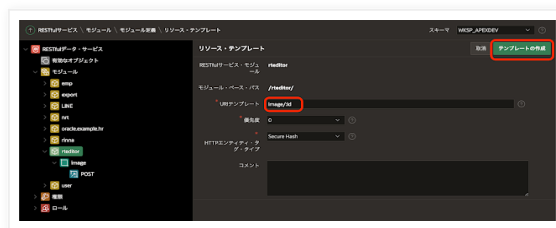
左ペインにてモジュールrteditorを選択し、テンプレートの作成を実行します。



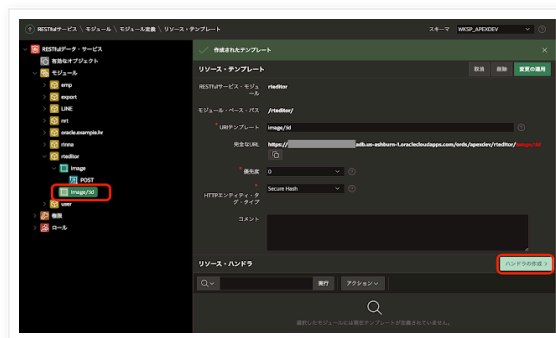
URIテンプレートとしてimage/:idを設定します。

画像のアップロードを受け付けるREST APIは、自分自身のURLに画像のIDを付加したURLを、画像のダウンロードURLとして返しています。そのため、ここで指定するテンプレートは（アップロードのテンプレートがimageなので）必ずimage/:idとします。

テンプレートの作成を実行します。



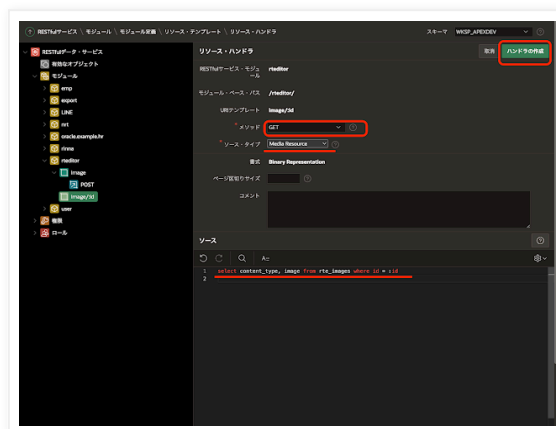
テンプレートimage/:idが作成されます。ハンドラの作成を実行します。



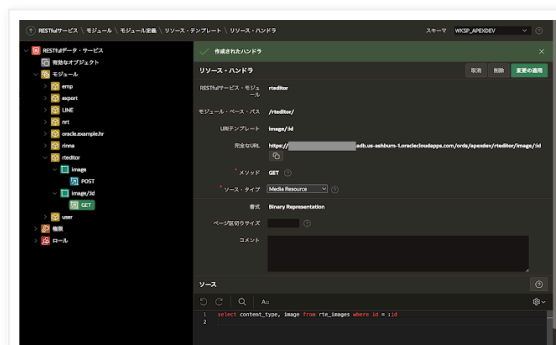
メソッドとしてGET、ソース・タイプとしてMedia Resourceを選択します。ソースに以下の1行を記載します。

```
select content_type, image from rte_images where id = :id
```

ハンドラの作成を実行します。



ハンドラが作成されました。

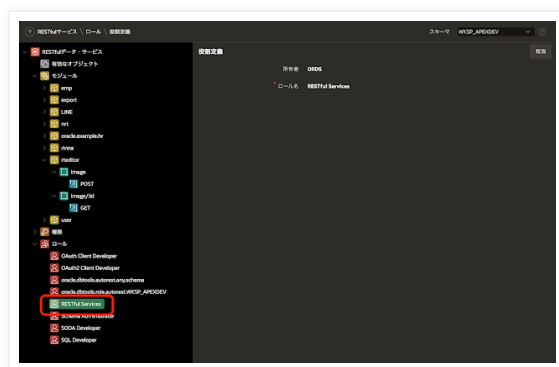


以上で画像をダウンロードするAPIの実装は完了です。

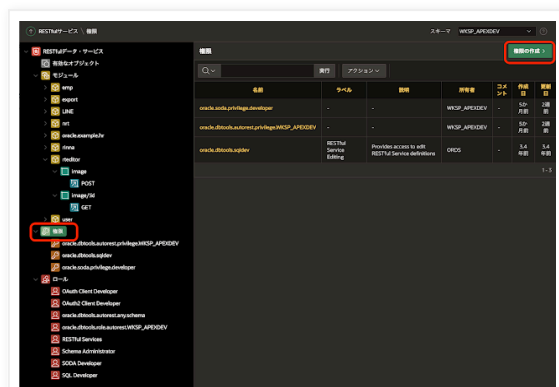
REST APIを保護する

作成したREST APIを、APEXで認証されたセッションからのみ呼び出しができるように保護します。マニュアルでは、[ファースト・パーティ認証](#)として説明されています。

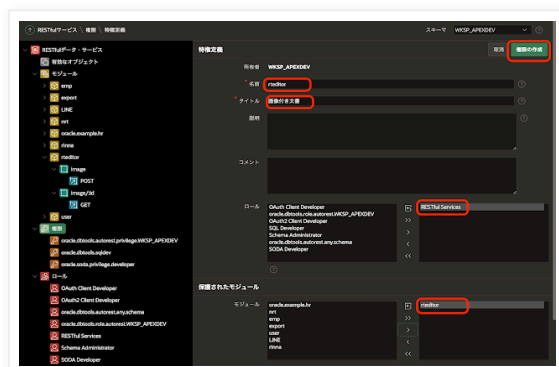
ロールは新規作成せず、デフォルトで作成されている**RESTful Services**を使います。



左ペインにて**権限**を選択します。**権限の作成**をクリックします。

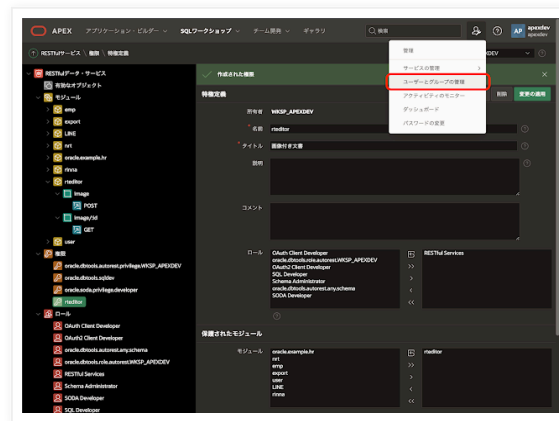


権限の名前を**rtditor**、タイトルは**画像付き文書**とします。ロールとして**RESTful Services**、保護されたモジュールとして**rtditor**をそれぞれ選択し、**権限の作成**をクリックします。



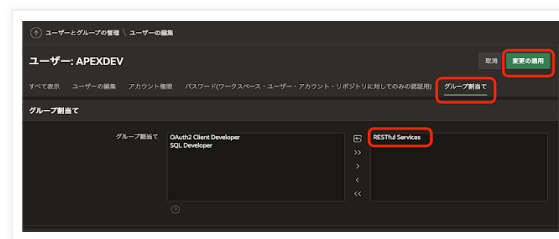
以上でモジュール**rtditor**のREST APIの保護が設定されました。

ユーザーとグループの管理を開き、モジュールrtditorのREST APIを呼び出すAPEXユーザーにRESTful Servicesのロールを割り当てます。



ロールを割り当てるユーザーの設定を開き、**グループ割当て**のセクションで**RESTful Services**を割り当てます。

変更の適用を実行します。



これで、**グループ割当て（ロール）**として**RESTful Services**が割り当たったユーザーでサインインしたAPEXアプリケーションより、作成したREST APIを呼び出すことができるようになりました。

続く

Yuji N. 時刻: 15:16

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.