

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。

2023年6月7日水曜日

GoogleのOpen ID Connect認証でリフレッシュ・トークンを取得する

Microsoft Azure ADのOpen ID Connect認証でリフレッシュ・トークンを取得し、それを使用してアクセス・トークンを更新する手順について、[以前に確認しました](#)。

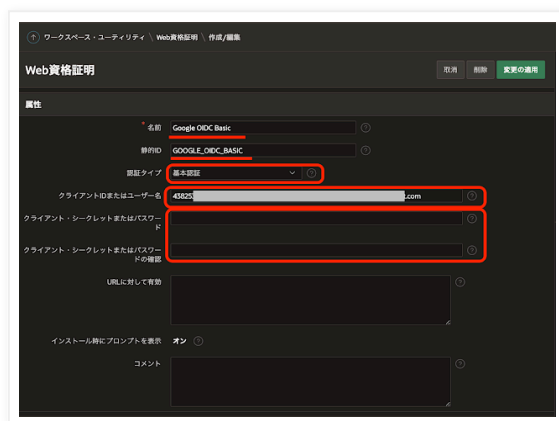
GoogleのOpen ID Connect認証について、同様に確認してみました。

GoogleのOpen ID Connectを使う実装は、以下の記事で紹介しています。このアプリケーションを元にして、リフレッシュ・トークンを扱う実装を追加します。

Googleで認証してGoogle Drive APIを呼び出す

GoogleにおいてもAzure ADのときと同様に、**認証タイプ**を**基本認証**とした**Web資格証明**を作成します。**クライアントID**または**ユーザー名**および**クライアント・シークレット**または**パスワード**は、**認証タイプ**が**OAuth2クライアント資格証明フロー**の**Web資格証明**に設定している**クライアントID**または**ユーザー名**および**クライアント・シークレット**または**パスワード**と同じ値を設定します。

今回の例では、**OAuth2**の資格証明の**静的ID**が**GOOGLE_DRIVE_CRED**、**基本認証**の**静的ID**が**GOOGLE_OIDC_BASIC**としています。



GoogleのOpen ID Connectの場合、リフレッシュ・トークンを発行するために、**認証URIパラメータ**として**access_type=offline**を含める必要があります。

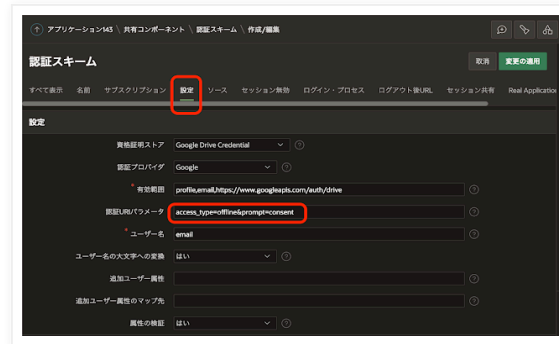
<https://developers.google.com/identity/openid-connect/openid-connect?hl=ja>

実装してみると、これに加えて再同意を含める必要がありました。そのため、**prompt=consent**も**認証URIパラメータ**に追加しています。

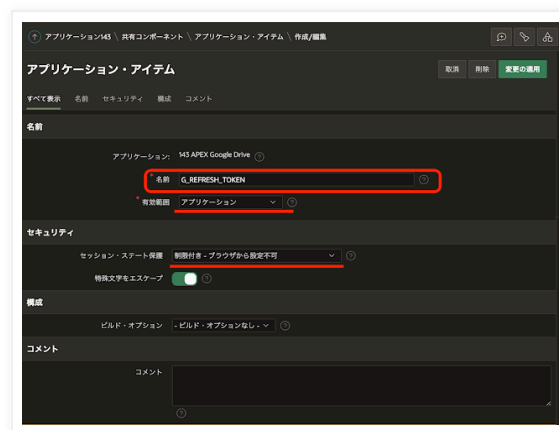
Webを調べるとprompt=true、prompt=forceといった記載も見つかります。上記のGoogleのドキュメントではnone、consent、select_accountが指定可能な値となっています。上記の3つの中では、consentが適切なようです。

認証URIパラメータは認証スキームに設定します。

access_type=offline&prompt=consent



取得したリフレッシュ・トークンを保存するアプリケーション・アイテムを、G_REFRESH_TOKENとして作成します。作成する手順はAzure ADと同じです。

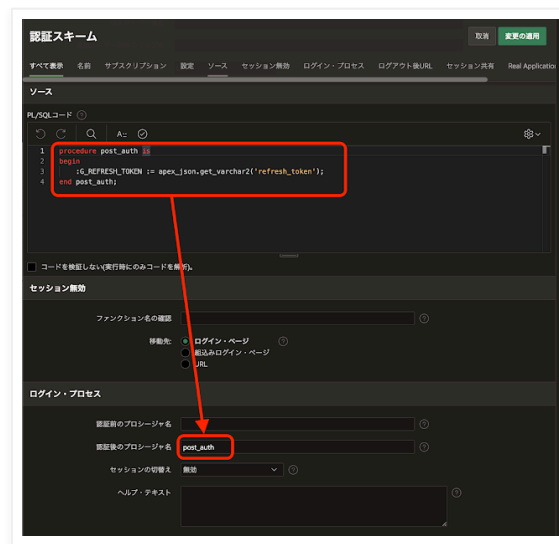


Googleが返す応答からリフレッシュ・トークンを取り出し、アプリケーション・アイテムG_REFRESH_TOKENに保存します。これもAzure ADと同じコードを、認証スキームの認証後のプロシージャとして実装します。

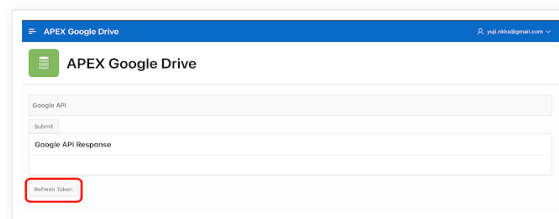
```
procedure post_auth is
begin
    :G_REFRESH_TOKEN := apex_json.get_varchar2('refresh_token');
end post_auth;
```

google-post-auth-refresh-token.sql hosted with ❤ by GitHub

[view raw](#)



アプリケーションの画面に、ボタンRefresh Tokenを作成します。



GoogleのOpen ID ConnectのDiscoveryドキュメントURIにアクセスし、トークン・エンドポイントを確認します。

<https://accounts.google.com/.well-known/openid-configuration>

今回の作業ではtoken_endpointとして、<https://oauth2.googleapis.com/token>が確認できました。

ボタンRefresh Tokenを押したときに実行されるプロセスのコードとして、以下を記述します。

```
declare
    l_refresh_token varchar2(4000) := :G_REFRESH_TOKEN;
    l_response clob;
    l_response_json json_object_t;
    l_access_token varchar2(4000);
    l_expires_in_sec number;
    C_CRED_OAUTH2 constant varchar2(20) := 'GOOGLE_DRIVE_CRED';
    C_CRED_BASIC constant varchar2(20) := 'GOOGLE_OIDC_BASIC';
begin
    apex_web_service.clear_request_headers;
    apex_web_service.set_request_headers('Content-Type','application/x-www-form-urlencoded',p_r
    l_response := apex_web_service.make_rest_request(
        p_url => 'https://oauth2.googleapis.com/token'
        ,p_http_method => 'POST'
        ,p_body => 'grant_type=refresh_token' || chr(38) || 'refresh_token=' || l_refresh_token
        ,p_credential_static_id => C_CRED_BASIC
```

```

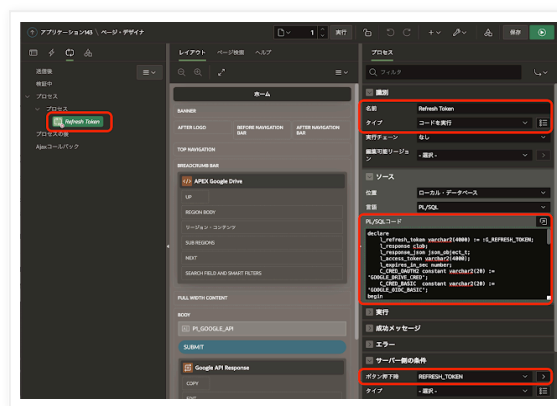
);
l_response_json := json_object_t.parse(l_response);
/* アクセス・トークンのアップデート */
l_access_token := l_response_json.get_string('access_token');
l_expires_in_sec := l_response_json.get_number('expires_in');
if l_access_token is not null then
    apex_credential.set_session_token(
        p_credential_static_id => C_CRED_OAUTH2
        ,p_token_type          => APEX_CREDENTIAL.C_TOKEN_ACCESS
        ,p_token_value         => l_access_token
        ,p_token_expires       => (sysdate + l_expires_in_sec/86400)
    );
    apex_debug.info('Access Token is updated by %s', substr(l_access_token, 1,20));
else
    apex_debug.info('Access Token is not updated. ');
end if;
/*
 * Googleの場合、リフレッシュ・トークンに有効期限がなく、アクセス・トークンのリフレッシュ時に
 * 再発行もされない模様。
 */
end;

```

google-refresh-access-token.sql hosted with ❤ by GitHub

[view raw](#)

Googleではリフレッシュ・トークンに有効期限がなく、また、Azure ADと異なりアクセス・トークンの再発行時にリフレッシュ・トークンが更新されることはないようです。(Googleのドキュメントより)

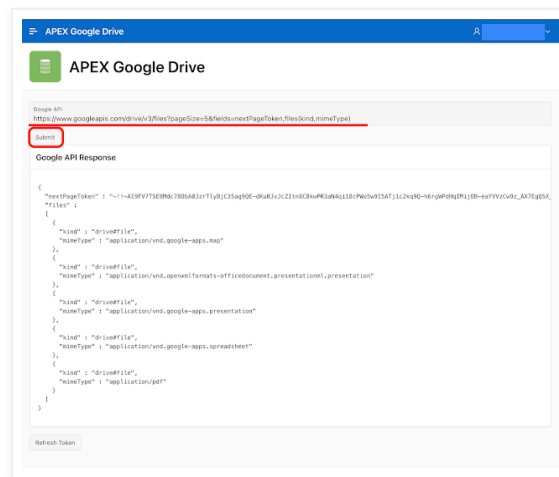


以上で実装は完了です。

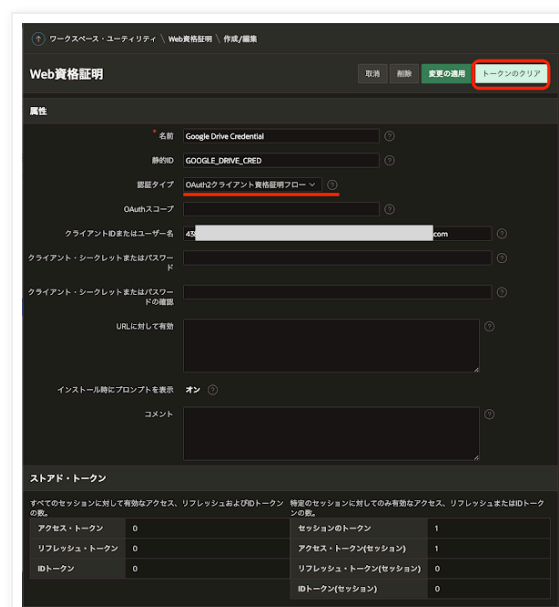
以下のGoogle Drive APIを呼び出し、動作を確認してみます。

[https://www.googleapis.com/drive/v3/files?pageSize=5&fields=nextPageToken,files\(kind,mimeType\)](https://www.googleapis.com/drive/v3/files?pageSize=5&fields=nextPageToken,files(kind,mimeType))

Submitをクリックすると、正常に応答が返ってきます。

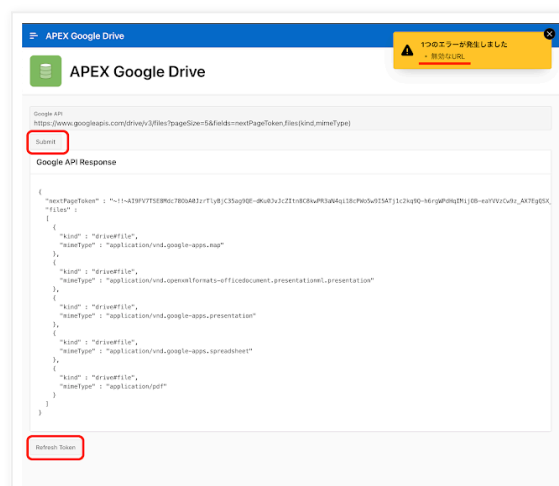


認証タイプがOAuth2クライアント資格証明フローのWeb資格証明を開き、トークンのクリアを実行します。



再度、Submitをクリックします。無効なURLというエラーが発生します。

エラーを確認した後、ボタンRefresh Tokenをクリックしアクセス・トークンを更新します。



続いてSubmitをクリックすると、API呼び出しが正常に終了することが確認できます。

以前に公開しているアプリケーションのエクストポートに、今回の実装を追加しました。
<https://github.com/ujnak/apexapps/blob/master/exports/apex-google-drive.zip>

以上になります。

Oracle APEXのアプリケーション作成の参考になれば幸いです。

完

Yuji N. 時刻: 15:41

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.
