

# 日々はOracle APEX

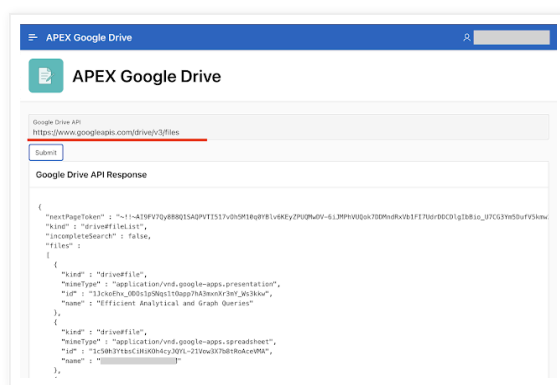
Oracle APEXを使った作業をしていて、気の付いたところを忘れないようにメモをとります。

2023年2月8日 水曜日

## Googleで認証してGoogle Drive APIを呼び出す

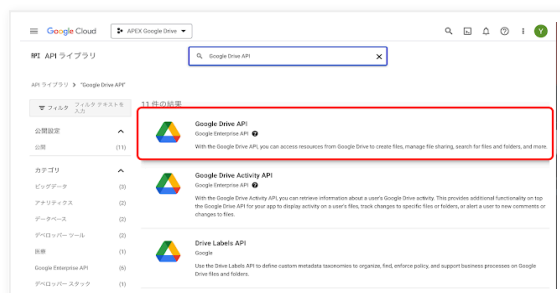
以前にAzure ADで認証した（プロトコルはOpen ID Connect）APEXアプリから、Microsoft Graph APIを呼び出せることを確認しています。同様の確認をGoogleアカウントで行ってみました。

確認のためにAPEXアプリを作成し、Google Drive APIを呼び出しています。MS Graph APIのときと同様に、APIの呼び出しはできました。



以下に実施した作業について記述します。

作成済みのGoogleのプロジェクトで、**Google Drive API**を有効にしたところから始めます。



**OAuth同意画面**を設定します。画面左のメニューより、**OAuth同意画面**を選択します。

今回はテストなので**外部**を選びます。Google Workspaceを利用している場合は、内部を選ぶことができるようです。

**作成**をクリックします。

RPI API とサービス	OAuth 同意画面
<ul style="list-style-type: none"> <li>有効な API とサービス</li> <li>ライブラリ</li> <li>認証情報</li> <li><b>OAuth 同意画面</b></li> <li>ページの使用に関する契約</li> </ul>	<p>アプリをどのように構成および登録するか（ターゲット ユーザーを含む）を選択します。プロジェクトに関連付けられているアプリは 1 つだけです。</p> <p>User Type</p> <p><input type="radio"/> 内部</p> <p><input checked="" type="radio"/> 外部</p> <p>組織内のユーザーのみが使用できます。確認を受けるためにアプリを送信する必要はありません。 <a href="#">ユーザーの種類の詳細</a></p> <p>Google アカウントを持つすべてのテストユーザーが使用できます。アプリはテストモードで起動し、アプリを使用できるのは、テストユーザーのリストに追加されたユーザーに限られます。アプリを本番環境に移す準備ができたら、アプリの確認が必要となる場合があります。 <a href="#">ユーザーの種類の詳細</a></p> <p><b>作成</b></p> <p><a href="#">Google の OAuth に関するご意見やご要望をお聞かせください。</a></p>

OAuth同意画面の設定を行います。

アプリ名は任意です。今回はAPEX Google Driveとしています。ユーザーサポートメールは、今回のようなケースではこのコンソールにサインインしている人のメール・アドレスを指定することになると思います。必須ではないようですが、アプリのロゴとなる画像をアップロードしています。

それ以外で必須の情報はデベロッパーの連絡先情報です。このコンソールにサインインしている人のメール・アドレスを指定しています。

今回の作業はAlways FreeのAutonomous Databaseで行っています。そのため承認済みドメインはoraclecloudapps.comになります。認証情報の設定時に自動で追加されるため、このページでは必ずしも設定する必要はありません。

保存して次へ進みます。

RPI API とサービス	アプリ登録の編集
<ul style="list-style-type: none"> <li>有効な API とサービス</li> <li>ライブラリ</li> <li>認証情報</li> <li><b>OAuth 同意画面</b></li> <li>ページの使用に関する契約</li> </ul>	<p>1 OAuth 同意画面 2 スコープ 3 テストユーザー 4 概要</p> <p>アプリ情報</p> <p>この情報は同意画面に表示されるため、デベロッパーのユーザー情報とデベロッパーへの問い合わせ方法をエンドユーザーが認識できます。</p> <p>アプリ名 * APEX Google Drive</p> <p>同意を求めるアプリの名前</p> <p>ユーザーサポートメール *</p> <p>ユーザーが同意に関して問い合わせるために使用</p> <p>アプリのロゴ</p> <p>これがお客様のロゴです。このロゴは、ユーザーがアプリを認識できるよう、OAuth 同意画面に表示されます。</p> <p>アプリのロゴのプレビュー</p> <p>アップロードするロゴファイル apex-apps.png</p> <p>ユーザーがアプリを認識できるように、同意画面に 1 MB 以下の画像をアップロードします。使用できる拡張子は、.png、.jpg、.gif、.bmp です。最適な表示を得るには、ロゴを 120 x 120 ピクセルの正方形にすることを推奨します。</p> <p>アプリのドメイン</p> <p>デベロッパーとユーザーを保護するために、Google では、OAuth を使用するアプリのみが承認済みドメインの使用を許可しています。同意画面では、次の情報がユーザーに表示されます。</p> <p>アプリケーションのホームページ</p> <p>ホームページのリンクをユーザーに提供します</p> <p>[アプリケーション プライバシー ポリシー] リンク</p> <p>一般公開のプライバシーポリシーへのリンクをユーザーに提供します</p> <p>[アプリケーション利用規約] リンク</p> <p>一般公開の利用規約へのリンクをユーザーに提供します</p> <p>承認済みドメイン</p> <p>同意画面または OAuth クライアントの構成でドメインが使用されている場合は、ここで承認済みドメインを指定する必要があります。アプリの検証に必要な情報は、Google Search Console にアクセスして、ドメインが承認済みであるかどうかを確認してください。承認済みドメインの上側の情報をご覧ください。</p> <p>+ ドメインの追加</p> <p>デベロッパーの連絡先情報</p> <p>メールアドレス *</p> <p>これらのメールアドレスは、プロジェクトの変更について Google からお知らせするため 사용됩니다。</p> <p><b>保存して次へ</b> キャンセル</p>

スコープを設定します。スコープを追加または削除をクリックします。

フィルタにGoogle Drive APIを指定します。次のスコープを選択します。

<https://www.googleapis.com/auth/drive>

画面では<https://www.googleapis.com/auth/docs>も指定していますが、APEXアプリケーションでは指定しないため不要です。

APIの呼び出しに必要なスコープは、[Google Drive APIのReference](#)のそれぞれのAPIの説明のAuthorizationのセクションに記載されています。

このスコープは、**Googleドライブのすべてのファイルの表示、編集、作成、削除**なので、非常に強い権限になります。

スコープを選択して、**更新**します。



機密性の高いスコープとして、選択したスコープが追加されています。

保存して次へ進みます。



テストユーザーを追加します。

これから作成するAPEXアプリケーションにサインインする開発者を追加します。

保存して次へ進みます。

The screenshot shows the 'API and Services' page in the Google Cloud console. The 'OAuth consent screen' is selected in the left sidebar. The 'Test users' section is highlighted with a red box, and the 'Save and next' button is also highlighted with a red box.

以上でOAuth同意画面の設定は完了です。

The screenshot shows the 'API and Services' page in the Google Cloud console. The 'OAuth consent screen' is selected in the left sidebar. The 'Scopes' section is highlighted with a red box, and the 'Test users' section is also highlighted with a red box.

続いて認証情報を設定します。

画面右より認証情報を開き、認証情報の作成からOAuthクライアントIDを選択します。

The screenshot shows the 'API and Services' page in the Google Cloud console. The 'Authentication' section is selected in the left sidebar. The 'Create new OAuth client ID' button is highlighted with a red box, and the 'OAuth 2.0 Client ID' option is also highlighted with a red box.

認証情報の作成に必要な情報を入力します。

アプリケーションの種類はウェブアプリケーションです。選択した種類によって入力する情報が変わります。承認済みのJavaScript生成元、承認済みのリダイレクトURIの入力が無い場合は、アプリケーションの種類の選択が間違っています。

アプリケーションの名前は任意です。今回はAPEX Google Driveとしています。

以下はAlways FreeのAutonomous Databaseでの例です。

承認済みのJavaScript生成元は以下のような指定になります。

`https://ADBの識別子-データベース名.adb.リージョン名.oraclecloudapps.com`

承認済みのリダイレクトURIは以下になります。

`https://ADBの識別子-データベース名.adb.リージョン名.oraclecloudapps.com/ords/apex_authentication.callback`

APEXのベースURLに`apex_authentication.callback`を繋げます。

作成をクリックします。



OAuthクライアントを作成しましたとポップアップが開きます。JSONをダウンロードをクリックして認証情報をファイルに保存します。

このファイルの内容より、Oracle APEXのWeb資格証明を作成します。



以上でGoogle側での作業は完了です。



一応、承認済みドメインの設定を確認しておきます。承認済みのJavaScript生成元、承認済みのリダイレクトURIとして設定したURLのトップドメインが設定されているはずです。



Oracle APEX側の作業に移ります。

アプリケーション・ビルダーからワークスペース・ユーティリティを開きます。

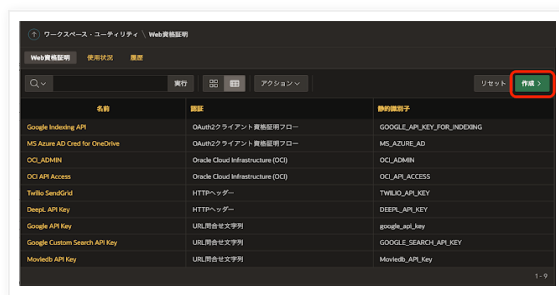


ワークスペース・ユーティリティよりWeb資格証明を開きます。



作成済みのWeb資格証明が一覧されます。

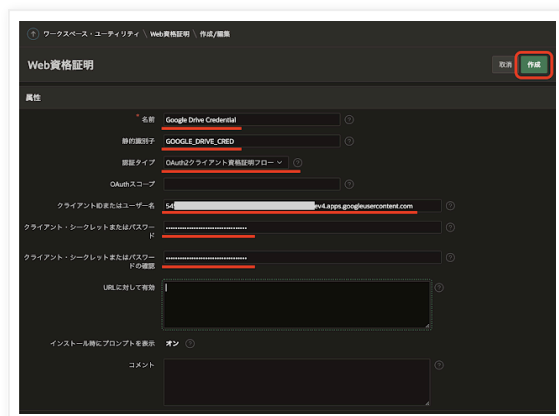
作成をクリックします。



名前はGoogle Drive Credentialとします。静的識別子はGOOGLE\_DRIVE\_CREDとしています。  
(Googleのコンソールでアプリにスコープを追加すると、Google Drive以外のAPIも呼び出せるはず  
です。そのような計画があるときはWeb資格証明にGoogle Driveという名前はつけない方がよいか  
もしれません。)

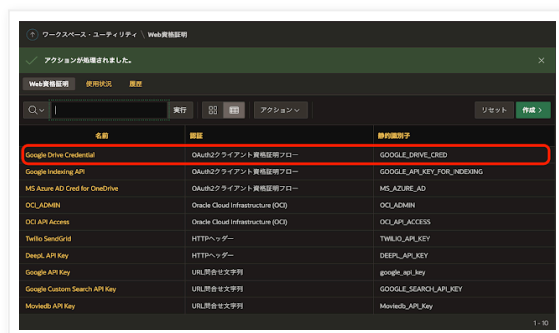
認証タイプはOAuth2クライアント資格証明フローです。クライアントIDまたはユーザー名には、  
ダウンロードしたJSONファイルに含まれるclient\_idの値を入力します。クライアント・シークレ  
ットまたはパスワードにはclient\_secretの値を入力します。

以上で作成をクリックします。



新たにWeb資格証明Google Drive Credentialが作成されました。

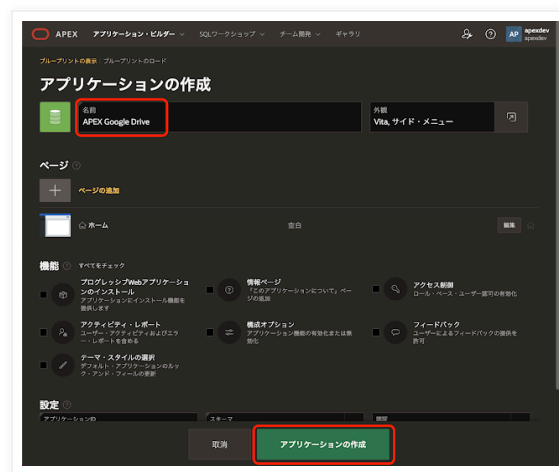
スクリーンショットには他にも色々なWeb資格証明があります。これらのWeb資格証明を使ったAPI  
連携は、すべて本ブログの記事として公開しています。



Google Drive APIを呼び出すAPEXアプリケーションを作成します。

アプリケーション作成ウィザードを起動し、空のアプリケーションを作成します。名前はAPEX Google Driveとします。

アプリケーションの作成をクリックします。



最初に認証スキームを作成します。

共有コンポーネントの認証スキームを開きます。



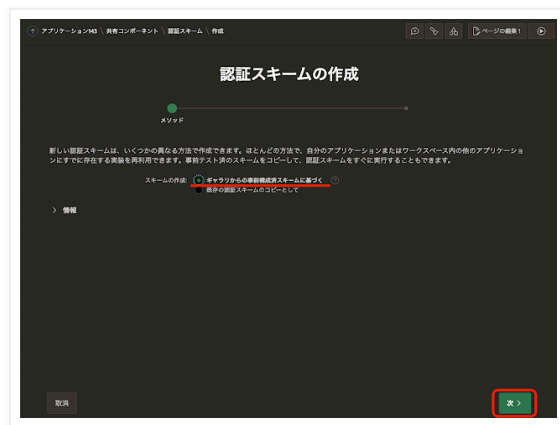
作成をクリックします。



スキームの作成はギャラリーからの事前構成済スキームに基づくを選択します。

次へ進みます。





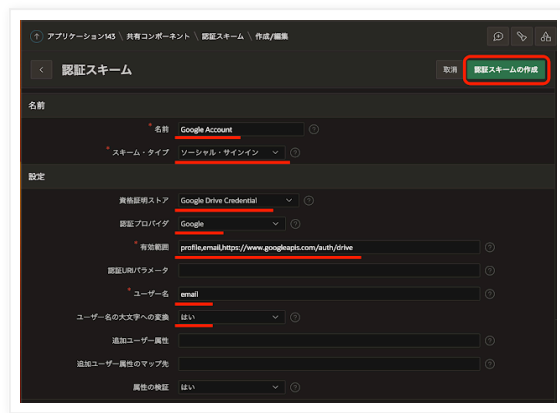
認証スキームの名前はGoogle Accountとします。スキーム・タイプとしてソーシャル・サインインを選択します。

設定の資格証明ストアとして、先ほど作成したGoogle Drive Credentialを選択します。認証プロバイダはGoogleです。有効範囲（スコープのこと）に以下を指定します。

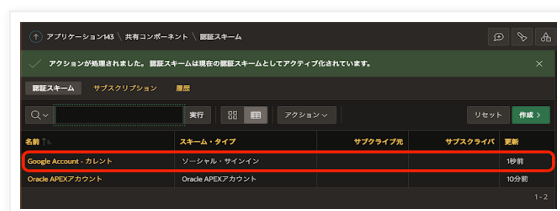
profile,email,https://www.googleapis.com/auth/drive

ユーザー名としてemailを指定します。メール・アドレスでは一意性を保証できない場合はsubを指定した方が良いでしょう。ユーザー名の大文字への変換は、APEXのデフォルトに合わせてはいに変更します。

以上で認証スキームの作成をクリックします。



認証スキームGoogle Accountが作成され、カレントの認証スキームになります。



アプリケーションを実行します。

Googleにログインが求められます。Chromeを利用していてGoogleにログイン済みの場合はスキップされるようです。OAuth同意画面の設定で、テストユーザーとして登録してユーザーのメール・アドレスを入力します。

Google にログイン

## ログイン

[「oraclecloudapps.com」に移動](#)

メールアドレスまたは電話番号

y m

[メールアドレスを忘れた場合](#)

続行するにあたり、Google はあなたの名前、メールアドレス、言語設定、プロフィール写真を oraclecloudapps.com と共有します。

[次へ](#)

[日本語](#) [ヘルプ](#) [プライバシー](#) [規約](#)

パスワードを入力します。

Google にログイン

## ログイン

[「oraclecloudapps.com」に移動](#)

パスワードを入力

パスワードを表示する

続行するにあたり、Google はあなたの名前、メールアドレス、言語設定、プロフィール写真を oraclecloudapps.com と共有します。

[パスワードをお忘れの場合](#)

[次へ](#)

[日本語](#) [ヘルプ](#) [プライバシー](#) [規約](#)

2段階認証を設定している場合は、認証を求められます。

Google にログイン

## 2段階認証プロセス

アカウントを安全に保つため、ログインするには本人確認を行う必要があります

 y...n



デバイスを確認してください

Google からお使いの...、他 2 個のデバイスに通知を送信しました。本人確認を行うには、Gmail アプリを開き、表示されたメッセージで **[はい]** をタップします。

☒ このデバイスでは次回から表示しない

再送信

[別の方法を試す](#)

日本語 ヘルプ プライバシー 規約

初回アクセス時に、**このアプリはGoogleで確認されていません**と表示されます。

**続行**をクリックします。

Google

## このアプリは Google で確認されていません

現在テスト中のアプリへのアクセスが許可されました。招待元の**デベロッパー**を信頼できる場合のみ、続行してください。

**続行** [安全なページに戻る](#)

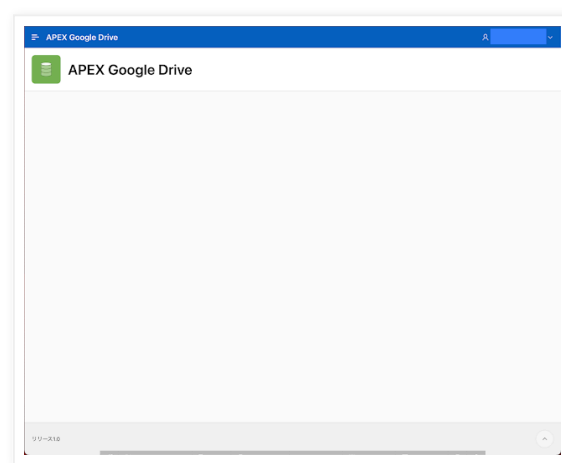
日本語 ヘルプ プライバシー 規約

このアプリが**Googleドライブのすべてのファイルの表示、編集、作成、削除**（つまりスコープ `https://www.googleapis.com/auth/drive` で可能な作業）を行なうことを許可します。

**チェック**を入れて、**続行**します。



アプリケーションのホーム・ページが開きます。



以上でGoogleを使用したソーシャル・サインインの認証を構成できました。

Google Drive APIを呼び出す機能を実装します。MS Graph APIを呼び出すアプリと、ほとんど同じです。

識別の名前はP1\_GOOGLE\_API、タイプはテキスト・フィールド、ラベルはGoogle APIとします。  
このページ・アイテムにGoogle Drive APIのURLを入力します。



Google Drive APIを呼び出すボタンを作成します。

識別のボタン名をSUBMIT、ラベルをSubmitとします。動作のアクションとして動的アクションで定義を選択します。



新規にリージョンを作成します。

識別のタイトルはGoogle API Responseとします。タイプとして動的コンテンツを選択します。

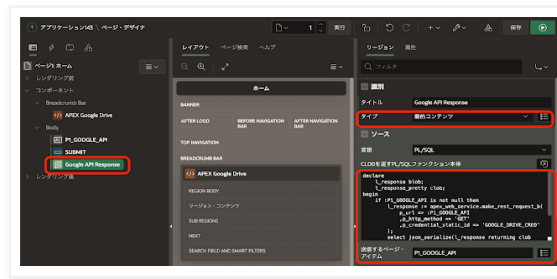
ソースのCLOBを返すPL/SQLファンクション本体として、以下を記述します。

```
declare
    l_response blob;
    l_response_pretty clob;
begin
    if :P1_GOOGLE_API is not null then
        l_response := apex_web_service.make_rest_request_b(
            p_url => :P1_GOOGLE_API
            ,p_http_method => 'GET'
            ,p_credential_static_id => 'GOOGLE_DRIVE_CRED'
        );
        select json_serialize(l_response returning clob pretty) into l_response_pretty from dual;
    end if;
    return '<pre><code>' || l_response_pretty || '</code></pre>';
end;
```

google-api.sql hosted with ❤ by GitHub

[view raw](#)

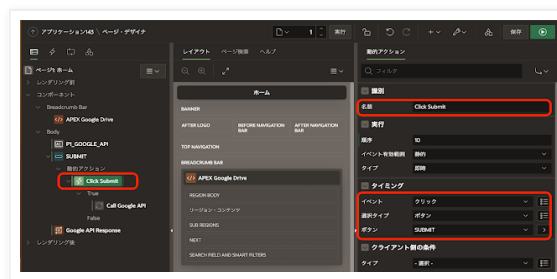
送信するページ・アイテムとしてP1\_GOGLE\_APIを設定します。



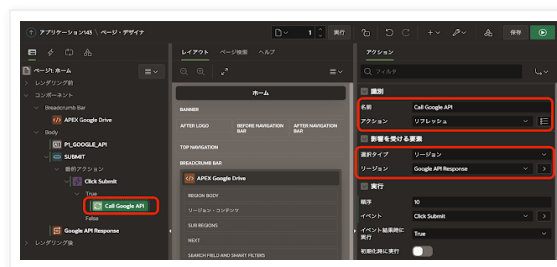
ボタンSUBMITをクリックしたときに、リージョンGoogle API Responseをリフレッシュする動的アクションを作成します。

ボタンSUBMIT上で動的アクションの作成を実行します。

識別のタイトルはClick Submitとします。タイミングはデフォルトで、イベントがクリック、選択タイプはボタン、ボタンとしてSUBMITが設定されます。



TRUEアクションの識別の名前はCall Google API、アクションとしてリフレッシュを選択します。影響を受ける要素の選択タイプはリージョン、リージョンはGoogle API Responseを指定します。

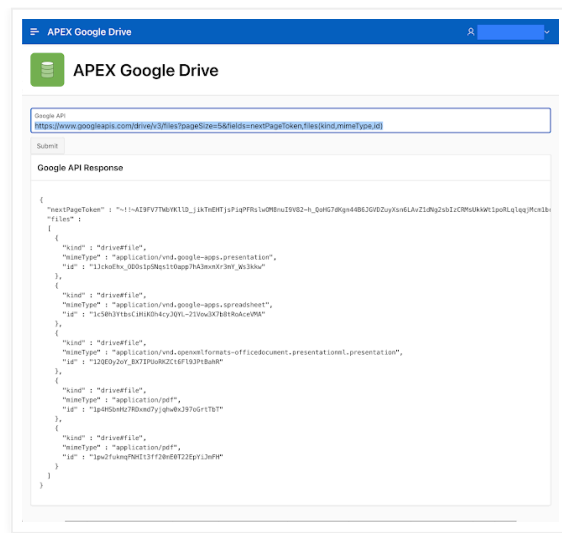


以上でGoogle Drive APIを呼び出して応答を確認するページは完成です。

ページを実行し、テストしてみます。

Google Drive APIとして以下を指定し、結果を確認します。

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



Google Drive APIがAPEXのアプリケーションにサインインしたユーザーの権限で実行されているが確認できます。また、pageTokenとnextPageTokenを使用したページ送りやfields指定が有効なことも確認できます。

今回の作業は以上になります。

作成したAPEX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/apex-google-drive.zip>

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: 14:43

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.