

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。

2024年2月7日水曜日

APEXアプリケーションよりGoogle Analytics 4 Data APIを呼び出す

Googleから提供されているGoogle Analytics 4のData APIを、Oracle APEXのアプリケーションから呼び出すことができるか確認してみました。

今の所、Analytics Data APIはベータ版での提供のようです。APIの概要については、Googleの以下のページに記載されています。

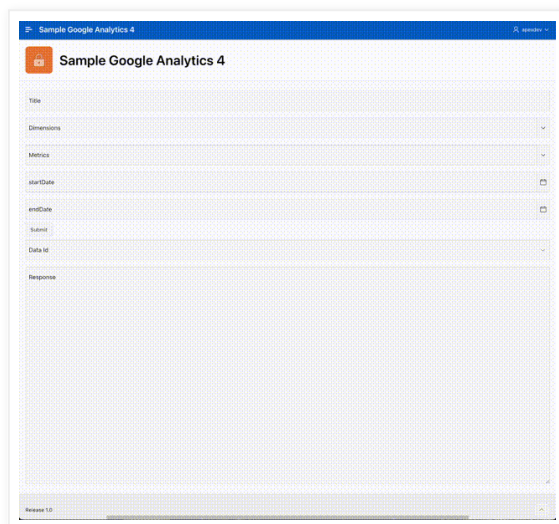
Analytics Data APIの概要

<https://developers.google.com/analytics/devguides/reporting/data/v1?hl=ja>

呼び出し元となるAPEXアプリケーションは、Always FreeのAutonomous Databaseで作成しています。

APIの呼び出しテストに使用したアプリケーションは以下のように動作します。

Analytics Data APIのリクエストに含める**dimensions**、**metrics**および**dateRanges**を指定してボタンSUBMITをクリックすると、Analytics Data APIの**runReport**を呼び出します。その応答をページ・アイテムに設定して確認するだけの簡単なアプリケーションです。



上記の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/sample-google-analytics-4.zip>

アプリケーションをインストールすると表GA4_DATA、GA4_DIMENSIONS、GA4_METRICSの3つの表が作成されます。表GA4_DIMENSIONSにはdimensionsとして選択可能なAPI名、表

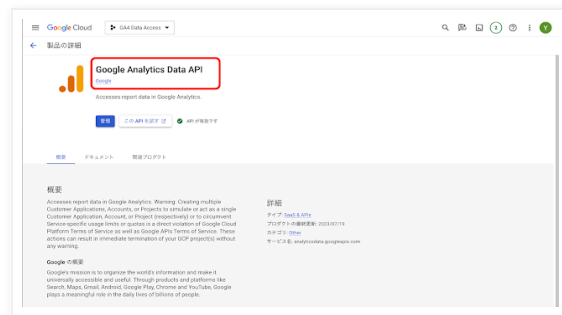
GA4_METRICSにはmetricsとして選択可能なAPI名が設定されています。

アプリケーション定義の置換文字列のG_PROPERTY_IDに参照するGA4のプロパティID、G_CREDENTIALとしてGoogleのサービスアカウントによるアクセストークンを保持する Web資格証明の静的IDを設定します。



以下、実施した作業を紹介します。

Google Cloudのコンソールより、プロジェクトにたいしてGoogle Analytics Data APIを有効にしています。



IAMと管理のサービスアカウントを開き、サービスアカウントを作成します。

サービスアカウントの作成とキーの追加の作業については、以前の記事「Google Indexing APIを呼び出す」での作業と同じです。

openssl pkcs12コマンドを実行して、PKCS#8形式の秘密キー・ファイルの作成までを実施します。



Google Analyticsのページに移り、管理のプロパティの詳細を開きます。

プロパティIDの値をメモしておきます。この値は、Analytics Data APIを呼び出すエンドポイントURLに含まれます。



アカウントのアクセス管理を開き、Google Cloudのプロジェクトに作成したサービスアカウントを追加します。役割としては閲覧者を選んでいきます。



Google側の準備は以上で完了です。

Oracle APEX側での作業を行います。

Googleのサービスアカウントで認証してAPIを呼び出すために使用するアクセス・トークンは、以前の記事「GoogleのGemini APIをOracle APEXから呼び出す」で紹介しているパッケージ UTL_CRED_GOOGLEを使って取得します。

以下のコードを実行します。引数p_scopeには以下のどちらかを与えます。サービスアカウントの役割として閲覧者を与えているので、今回はreadonlyの方をp_scopeに与えています。

<https://www.googleapis.com/auth/analytics>
<https://www.googleapis.com/auth/analytics.readonly>

```
set serveroutput on
declare
    C_RSA_KEY constant varchar2(32767) := q'~
-----BEGIN PRIVATE KEY-----
サービスアカウントの秘密鍵
-----END PRIVATE KEY-----
~';
    l_jwt    varchar2(32767);
    l_token  varchar2(32767);
begin
    l_jwt := utl_cred_google.generate_jwt(
        p_secret => C_RSA_KEY
        ,p_scope => 'https://www.googleapis.com/auth/analytics.readonly'
        ,p_iss  => 'サービスアカウントのメール'
    );
    dbms_output.put_line(l_jwt);
```

```

utl_cred_google.create_or_update_apex_credential(
    p_jwt => l_jwt
    ,p_credential_static_id => 'GOOGLE_GA4_API_TOKEN'
    ,p_workspace_name => 'APEXDEV' -- ワークスペースに合わせて変更
    ,p_token => l_token
);
dbms_output.put_line(l_token);
commit;
end;
/

```

create_apex_credential_for_ga4.sql hosted with ❤ by GitHub

[view raw](#)

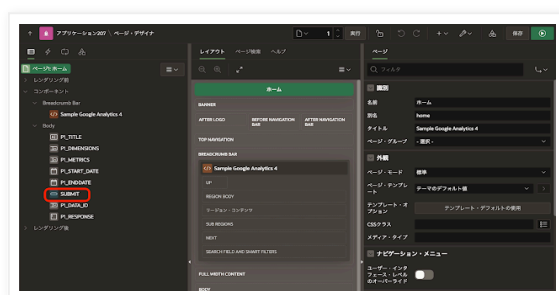
上記のコードを実行すると、Web資格証明GOOGLE_GA4_API_TOKENにアクセス・トークンが保存されます。APEX_WEB_SERVICE.MAKE_REST_REQUESTを呼び出す際にp_credential_static_idとしてこのWeb資格証明を指定すると、保存されているアクセス・トークンを使ってAPIにアクセスします。

アクセス・トークンの有効期間は1時間なので、1時間ごとに同じスクリプトを実行してアクセス・トークンを更新する必要があります。

Analytics Data APIを呼び出す処理はホーム・ページに実装しています。

runReportの呼び出しに含めるデータを、ページ・アイテムP1_TITLE、P1_DIMENSIONS、P1_METRICS、P1_START_DATEとP1_END_DATEに設定します。runReportではdateRangesとして4つまでstartDateとendDateの組み合わせを指定できるようですが、今回の実装では1つに限定しています。その他のパラメータについても、細かな設定については省いています。(dimensionのdimensionExpressionやmetricのexpressionなど)

ボタンSUBMITをクリックするとページが送信され、APIの呼び出しが行われます。



APIを呼び出すプロセスをCall GA4 DATA APIとして作成しています。

ソースのPL/SQLコードに、以下を記述しています。

```

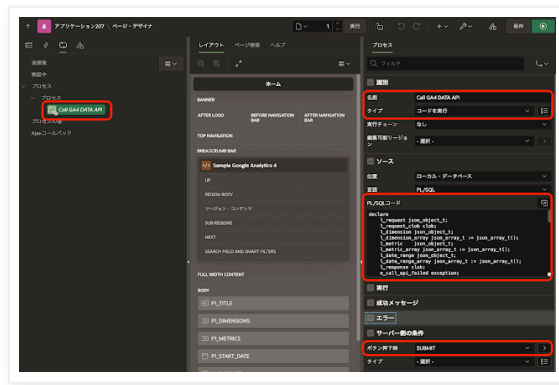
declare
    l_request json_object_t;
    l_request_clob clob;
    l_dimension json_object_t;
    l_dimension_array json_array_t := json_array_t();
    l_metric      json_object_t;

```

```

l_metric_array json_array_t := json_array_t();
l_date_range json_object_t;
l_date_range_array json_array_t := json_array_t();
l_response clob;
e_call_api_failed exception;
begin
  l_request := json_object_t();
  /* prep dimension */
  for r in (select column_value from apex_string.split(:P1_DIMENSIONS,':'))
  loop
    l_dimension := json_object_t();
    l_dimension.put('name', r.column_value);
    l_dimension_array.append(l_dimension);
  end loop;
  l_request.put('dimensions', l_dimension_array);
  /* prep metrics */
  for r in (select column_value from apex_string.split(:P1_METRICS, ':'))
  loop
    l_metric := json_object_t();
    l_metric.put('name', r.column_value);
    l_metric_array.append(l_metric);
  end loop;
  l_request.put('metrics', l_metric_array);
  /* date range */
  l_date_range := json_object_t();
  l_date_range.put('startDate', :P1_START_DATE);
  l_date_range.put('endDate', :P1_ENDDATE);
  l_date_range.put('name', 'first_period');
  l_date_range_array.append(l_date_range);
  l_request.put('dateRanges', l_date_range_array);
  l_request_clob := l_request.to_clob();
  apex_web_service.clear_request_headers();
  apex_web_service.set_request_headers('Content-Type', 'applicaton/json', p_reset => false);
  l_response := apex_web_service.make_rest_request(
    p_url => 'https://analyticsdata.googleapis.com/v1beta/properties/' || :G_PROPERTY_ID ||
    ,p_http_method => 'POST'
    ,p_body => l_request_clob
    ,p_credential_static_id => :G_GA4_CREDENTIAL_TOKEN
  );
  if apex_web_service.g_status_code <> 200 then
    raise e_call_api_failed;
  end if;
  insert into ga4_data(title, response) values(:P1_TITLE, l_response);
end;

```



APIを呼び出した結果は表GA4_DATAにJSONのまま保存し、ページ・アイテムP1_RESPONSEに表示させています。

とりあえずGoogle Analytics 4 Data APIをAPEXのアプリケーションから呼び出せることを確認しました。

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: 17:48

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.