

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2月3日 金曜日

Google Blogger APIを呼び出し記事の一覧と記事の内容を取得する

このブログはGoogleのBloggerを使っているのですが、Google Search Consoleで確認すると、インデックスが作成されているページは120程度で少々残念な状況でした。またコード・スニペットはGistに載せて埋め込んでいるため、検索の対象になっていません。

これらのデータをデータベースに保存しOracle Textで検索できるようにするため、まずはBloggerから記事の内容を取得することにしました。

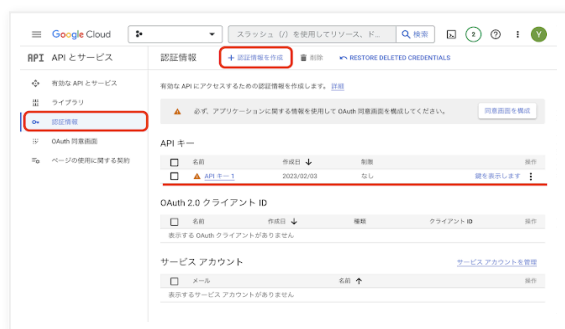
GoogleのBloggerを使っている人以外にはあまり関係ないかもしれませんが、JSONを返すGoogleのAPI v3.0は他のAPIと共通の仕様もあるようです。例えば記事の一覧をすべて取得するにはAPIを複数回発行する必要がありますが、そのために以下のような指定を行っています。

- URLパラメータのfieldsにnextPageTokenを含める。
- JSONの応答からnextPageTokenを取り出す。
- URLパラメータpageTokenにnextPageTokenの値を指定し、その後は同じAPIを発行する。
- nextPageTokenが無くなるまで繰り返す。

実施した作業について、簡単に紹介します。

Google Cloudのコンソールよりプロジェクトを作成し、Blogger APIを有効にします。Blogger APIを有効にした後、認証情報の画面を開き、APIキーを作成します。

今回は公開された記事しか扱わないため、APIキーでAPIの認証を行なうことにしました。



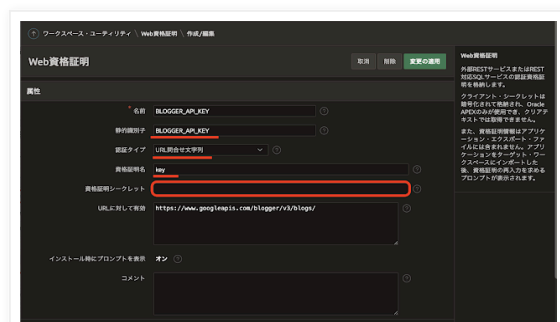
GoogleのAPIキーによるWeb資格証明を作成します。

Oracle APEXのワークスペース・ユーティリティよりWeb資格証明を開きます。

Web資格証明の名前と静的識別子は任意です。今回はBLOGGER_API_KEYとしています。

認証タイプとしてURL問合せ文字列を選択し、資格証明名としてkeyを指定します。その上で、資格証明シークレットにAPIキーを入力します。

URLに対して有効は、Blogger APIのURLを設定しています。これはAPIの種類によって変わるでしょう。



このWeb資格証明をAPEX_WEB_SERVICE.MAKE_REST_REQUESTの引数p_credential_static_idに指定すると、URL問合せ文字列としてkey=APIキーが追加されます。

APIを呼び出すコードはパッケージUTL_BLOGGER_APIとして作成しました。

```
create or replace package UTL_BLOGGER_API as

/*
 * 記事を保存する表の定義
 */

create table cds_documents(
    id varchar2(20) primary key
    ,published timestamp with time zone not null
    ,updated    timestamp with time zone not null
    ,url varchar2(400) not null
    ,title varchar2(4000) not null
    ,content clob
);

*/

/**
 * Google BloggerのBlog IDを指定して、記事の一覧を取得する。
 *
 * 一回のAPI呼び出しで取得できる記事が10件なので、すべて取得できるまで繰り返しAPIを発行する。
 *
 * API Ref:
 * https://developers.google.com/blogger/docs/3.0/reference/posts/list
 *
 * @param p_blog_id    BloggerのBlogID
 * @param p_credential_static_id  Blogger APIに与えるAPIキーであるWeb資格証明
 * @param p_limit       繰り返しAPIを発行する上限。
 * @param p_max_results  応答に含まれる最大の行数。
 */
```

```

*/
procedure get_posts (
    p_blog_id in varchar2
    ,p_credential_static_id in varchar2
    ,p_limit in number default 20
    ,p_max_results in number default 100
);

/**
 * 記事のIDを指定して、記事の内容を取得する。
 *
 * API Ref:
 * https://developers.google.com/blogger/docs/3.0/reference/posts/get
 *
 * @param p_post_id 記事のID
 * @param p_blog_id BloggerのBlogID
 * @param p_credential_static_id Web資格証明
 */
procedure get_post (
    p_post_id in varchar2
    ,p_blog_id in varchar2
    ,p_credential_static_id in varchar2
);
end;
/

create or replace package body utl_blogger_api is
G_BLOGGER_URL constant varchar2(400) := 'https://www.googleapis.com/blogger/v3/blogs/';

function get_posts_priv(
    p_page_token in varchar2
    ,p_blog_id in varchar2
    ,p_credential_static_id in varchar2
    ,p_max_results in number
)
return varchar2
as
    l_api_url varchar2(600);
    l_response clob;
    l_response_json json_object_t;
begin
    /* Bloggerのポスト一覧を取得する */
    l_api_url := G_BLOGGER_URL || p_blog_id || '/posts?fields=nextPageToken,items(id,published,updateTime)&maxResults=' || p_max_results;
    /* ページ送りの指定がある場合は、pageTokenを追加する。 */
    if p_page_token is not null then
        l_api_url := l_api_url || '&pageToken=' || p_page_token;
    end if;

```

```

end if;
/* ポスト一覧の取得 */
apex_web_service.clear_request_headers();
apex_web_service.set_request_headers('Content-Type','application/json',p_reset = false);
l_response := apex_web_service.make_rest_request(
    p_url => l_api_url
    ,p_http_method => 'GET'
    ,p_credential_static_id => p_credential_static_id
);
if apex_web_service.g_status_code <> 200 then
    raise_application_error(-20001,'Google API Request Failed');
end if;
/* レスポンスのJSON配列を表CDS_DOCUMENTSにマージする */
merge into cds_documents a
using (
    select line_number
        ,col001 id
        ,col002 url
        ,col003 title
        ,to_timestamp_tz(col004,'YYYY-MM-DD"T"HH24:MI:SSTZH:TZM') published
        ,to_timestamp_tz(col005,'YYYY-MM-DD"T"HH24:MI:SSTZH:TZM') updated
    from apex_data_parser.parse(
        p_content => apex_util.clob_to_blob(l_response)
        ,p_file_type => 4
    )
) r
on (a.id = r.id)
when matched then
    update set
        url = r.url
        ,title = r.title
        ,published = r.published
        ,updated = r.updated
when not matched then
    insert
        (id, url, title, published, updated)
    values
        (r.id, r.url, r.title, r.published, r.updated)
;
-- dbms_output.put_line(l_response);
/* 応答からnextPageTokenを見つけて、それを返す */
l_response_json := json_object_t(l_response);
return l_response_json.get_string('nextPageToken');
end get_posts_priv;

procedure get_posts(
    p_blog_id IN VARCHAR2

```

```

,p_credential_static_id IN VARCHAR2
,p_limit in number
,p_max_results in number
)
as
    l_next_page_token varchar2(40);
begin
    l_next_page_token := null;
    for i in 1..p_limit
    loop
        l_next_page_token := get_posts_priv(
            p_page_token => l_next_page_token
            ,p_blog_id => p_blog_id
            ,p_credential_static_id => p_credential_static_id
            ,p_max_results => p_max_results
        );
        if l_next_page_token is null then
            exit;
        end if;
    end loop;
end get_posts;

procedure get_post(
    p_post_id IN VARCHAR2
    ,p_blog_id IN VARCHAR2
    ,p_credential_static_id IN VARCHAR2
)
as
    l_api_url varchar2(600);
    l_response clob;
    l_response_json json_object_t;
    l_content clob;
begin
    l_api_url := G_BLOGGER_URL || p_blog_id || '/posts/' || p_post_id;
    /* 記事の取得 */
    apex_web_service.clear_request_headers();
    apex_web_service.set_request_headers('Content-Type','application/json',p_reset = false);
    l_response := apex_web_service.make_rest_request(
        p_url => l_api_url
        ,p_http_method => 'GET'
        ,p_credential_static_id => p_credential_static_id
    );
    if apex_web_service.g_status_code <> 200 then
        raise_application_error(-20001,'Google API Request Failed');
    end if;
    -- dbms_output.put_line(l_response);

```

```
l_response_json := json_object_t(l_response);
l_content := l_response_json.get_clob('content');
-- content = l_response_json.get_clob('content') と書くと実行時にエラーが発生する。
update cds_documents set content = l_content where id = p_post_id;
end get_post;

end utl_blogger_api;
/
```

utl_blogger_api.sql hosted with ❤ by GitHub

[view raw](#)

これでBloggerから取得した記事の一覧を表BLOGGER_ARTICLESに保存し、それぞれの記事の内容も取り出して表に保存できるようになりました。

完

Yuji N. 時刻: 18:45

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.