

# 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。

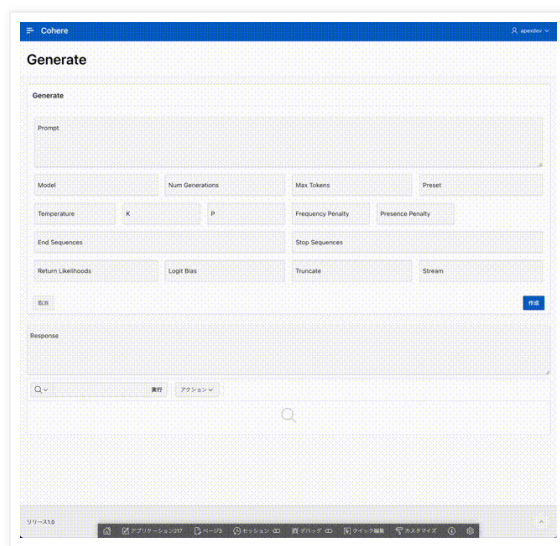
2023年6月26日月曜日

## CohereのCo.GenerateをOracle APEXから呼び出してみる

以前にCohereのCo.Classifyを呼び出すAPEXのアプリケーションを作成しました。[こちらの記事](#)です。同じアプリケーションに、Co.GenerateのAPIを呼び出すページを作ってみました。

CohereのCo.GenerateのAPIの説明はこちらです。

作成したページで、Examplesの[Ad Copy AIDA Framework](#)を呼び出してみます。



すでにCo.Classifyの実装が済んでいるところから始めます。

新しいページの名前を**Generate**、**ページ番号3**として作成します。

**ページの作成**をクリックします。



**フォーム**を選択します。



ページ番号は3、名前はGenerateとします。ページ・モードは標準です。

データ・ソースはローカル・データベース、ソース・タイプとしてSQL問合せを選択し、SQL SELECT文を入力に以下を記述します。ページ・アイテムを作成するためのSELECT文で、フォームのソースとしては使いません。

```
select
  0 as "ID"
, 'this column is for prompt string that should be long enough to reserve max string size' as "P"
, 'command' as "MODEL"
, 1 as "NUM_GENERATIONS"
, 20 as "MAX_TOKENS"
, '' as "PRESET"
, 0.75 as "TEMPERATURE"
, 0 as "K"
, 0.75 as "P"
, 0.0 as "FREQUENCY_PENALTY"
, 0.0 as "PRESENCE_PENALTY"
, '[]' as "END_SEQUENCES"
, '[]' as "STOP_SEQUENCES"
, 'NONE' as "RETURN_LIKELIHOODS"
, '{}' as "LOGIT_BIAS"
, 'END' as "TRUNCATE"
, 'N' as "STREAM"
from dual;
```

cohere-generate-source.sql hosted with ❤ by GitHub

[view raw](#)

ナビゲーションはデフォルトのまま、ブレッドキラムとナビゲーションの双方を生成します。

次へ進みます。

主キー列1としてID(Number)を選択します。ブランチ・ページの送信時にここにブランチ、取り消してページに移動の双方に、自分自身であるページ番号3を指定します。

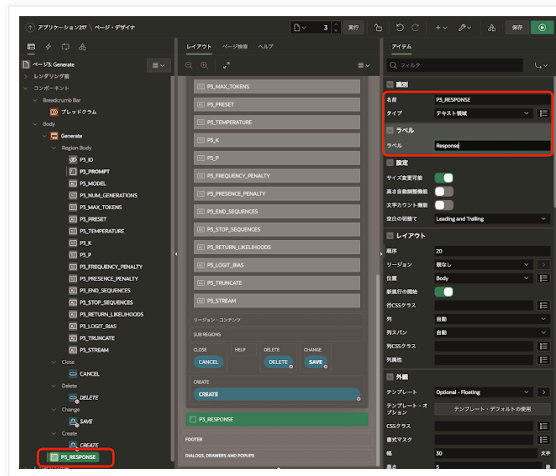
ページの作成をクリックします。

フォームのページが作成されます。最低限必要な調整を行います。

ページ・アイテムP3\_PROMPTを選択し、識別のタイプをテキスト領域に変更します。検証の必須の値をオン、最大長を4000とします。

Co.Generateのレスポンスを保持するページ・アイテムとして、P3\_RESPONSEを作成します。

識別の名前はP3\_RESPONSE、タイプはテキスト領域、ラベルはResponseとします。



レスポンスに含まれる生成された文字列を一覧する、対話モード・レポートを作成します。

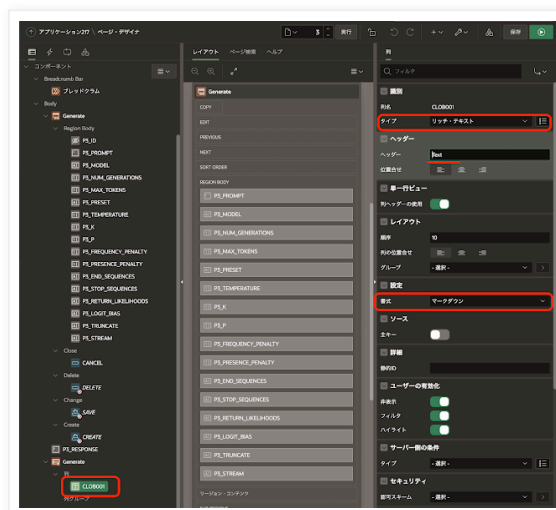
リージョンを作成し、識別の**タイトル**を**Generate**とします。**タイプ**として**対話モード・レポート**を選択します。**ソースのタイプ**として**SQL問合せ**を選択し、**SQL問合せ**に以下を記述します。

```
select clob001 from apex_collections where collection_name = 'GENERATE'
```



Cohereが生成した文字列には改行が含まれています。改行を認識させるため、列**CLOB001**の**タイプ**を**リッチ・テキスト**に変更し、**設定の書式**として**マークダウン**を選択します。

**ヘッダー**は**Text**に変更します。



Co.GenerateのAPIを呼び出すコードを記述します。

左ペインで**プロセス・ビュー**を表示し、**プロセス・フォームGenerate**を選択します。

設定のターゲット・タイプをPL/SQL Codeに変更し、挿入/更新/削除するPL/SQLコードとして以下を記述します。

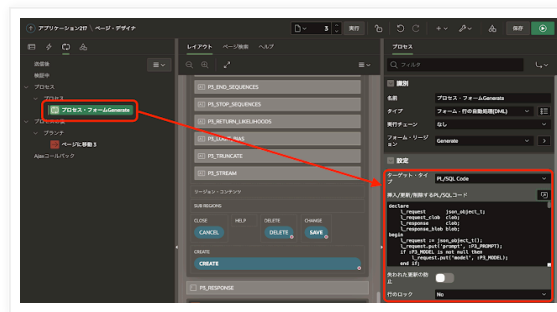
```
declare
    l_request      json_object_t;
    l_request_clob clob;
    l_response     clob;
    l_response_blob blob;
begin
    l_request := json_object_t();
    l_request.put('prompt', :P3_PROMPT);
    if :P3_MODEL is not null then
        l_request.put('model', :P3_MODEL);
    end if;
    if :P3_NUM_GENERATIONS is not null then
        l_request.put('num_generations', to_number(:P3_NUM_GENERATIONS));
    end if;
    if :P3_MAX_TOKENS is not null then
        l_request.put('max_tokens', to_number(:P3_MAX_TOKENS));
    end if;
    if :P3_PRESET is not null then
        l_request.put('preset', :P3_PRESET);
    end if;
    if :P3_TEMPERATURE is not null then
        l_request.put('temperature', to_number(:P3_TEMPERATURE));
    end if;
    if :P3_K is not null then
        l_request.put('k', to_number(:P3_K));
    end if;
    if :P3_P is not null then
        l_request.put('p', to_number(:P3_P));
    end if;
    if :P3_FREQUENCY_PENALTY is not null then
        l_request.put('frequency_penalty', to_number(:P3_FREQUENCY_PENALTY));
    end if;
    if :P3_PRESENCE_PENALTY is not null then
        l_request.put('presence_penalty', to_number(:P3_PRESENCE_PENALTY));
    end if;
    if :P3_END_SEQUENCES is not null then
        l_request.put('end_sequences', json_array_t.parse(:P3_END_SEQUENCES));
    end if;
    if :P3_STOP_SEQUENCES is not null then
        l_request.put('stop_sequences', json_array_t.parse(:P3_STOP_SEQUENCES));
    end if;
    if :P3_RETURN_LIKELIHOODS is not null then
        l_request.put('return_likelihoods', :P3_RETURN_LIKELIHOODS);
    end if;
```

```

if :P3_LOGIT_BIAS is not null then
    l_request.put('logit_bias', json_object_t.parse(:P3_LOGIT_BIAS));
end if;
if :P3_TRUNCATE is not null then
    l_request.put('truncate', :P3_TRUNCATE);
end if;
if :P3_STREAM is not null then
    if :P3_STREAM = 'Y' then
        l_request.put('stream', true);
    else
        l_request.put('stream', false);
    end if;
end if;
l_request_clob := l_request.to_clob();
apex_debug.info('Co.Generate Request: %s', l_request_clob);
apex_web_service.clear_request_headers;
apex_web_service.set_request_headers('Accept','application/json', p_reset => false);
apex_web_service.set_request_headers('Content-Type','application/json', p_reset => false);
l_response_blob := apex_web_service.make_rest_request_b(
    p_url => 'https://api.cohere.ai/v1/generate'
    ,p_http_method => 'POST'
    ,p_body => l_request_clob
    ,p_credential_static_id => 'COHERE_API'
);
-- 応答は整形して表示する。
select json_serialize(l_response_blob returning clob pretty) into l_response from dual;
:P3_RESPONSE := l_response;
-- 生成されたテキストをAPEXコレクションに投入する。
apex_collection.create_or_truncate_collection('GENERATE');
for r in (
    select id, text from json_table(l_response, '$.generations[*]'
        columns
        (
            id varchar2(80) path '$.id'
            ,text clob path '$.text'
        )
    )
)
loop
    apex_collection.add_member(
        p_collection_name => 'GENERATE'
        ,p_c001 => r.id
        ,p_clob001 => r.text
    );
end loop;
end;

```

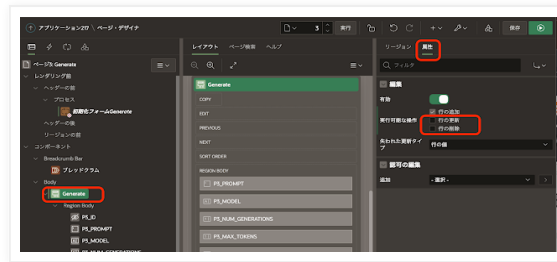
失われた更新の防止はオフ、行のロックはNoに変更します。



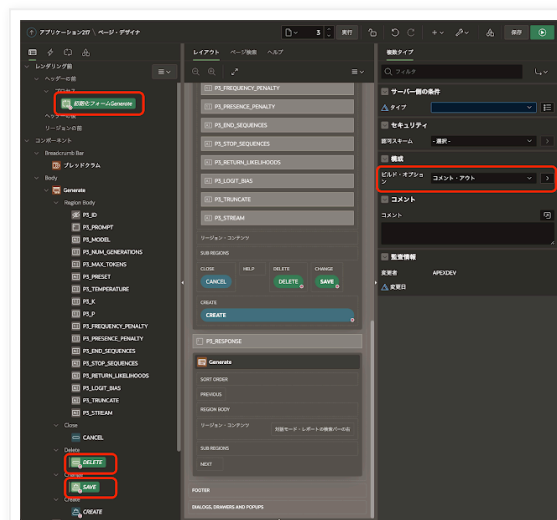
以上でアプリケーションとしては動作します。

動作や見た目を調整します。

フォームGenerateの属性を開き、編集の実行可能な操作より行の更新と行の削除のチェックを外します。このフォームには、行の更新と削除は実装していません。



プロセス初期化フォームGenerate、ボタンDELETE、ボタンSAVEを選択し、ビルド・オプションでコメント・アウトします。これらの機能も使われることはありません。



ページ・アイテムP3\_NUM\_GENERATIONS、P3\_MAX\_TOKENS、P3\_PRESET、P3\_K、P3\_P、P3\_FREQUENCY\_PENALTY、P3\_PRESENCE\_PENALTY、P3\_STOP\_SEQUENCES、P3\_LOGIT\_BIAS、P3\_TRUNCATE、P3\_STREAMを選択し、レイアウトの新規行の開始をオフにします。



作成したアプリケーションを実行すると、本記事の先頭のGIF動画のように動作します。

作成したアプリケーションのエクスポートとして、以前のエクスポートを更新しています。  
<https://github.com/ujnak/apexapps/blob/master/exports/sample-cohere-api.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: 14:11

共有

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.