

日々是Oracle APEX

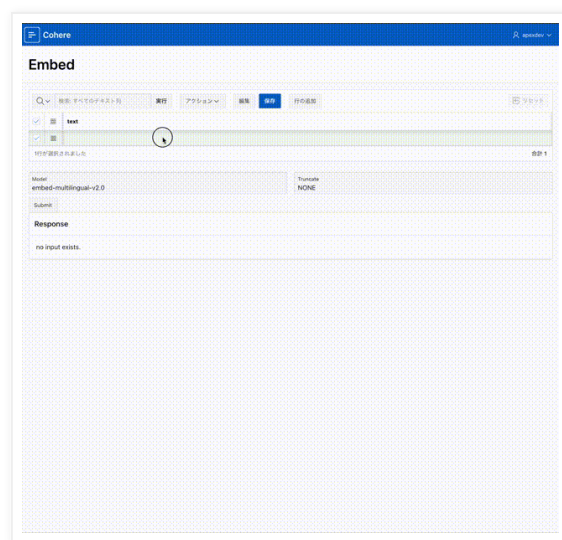
Oracle APEXを使った作業をしていて、気の付いたところを忘れないようにメモをとります。

2023年6月26日月曜日

CohereのCo.EmbedをOracle APEXから呼び出してみる

CohereのAPIのCo.Classify、Co.Generateを呼び出すAPEXアプリケーションに、Co.Embedを呼び出すページを追加しました。Co.EmbedのAPIリファレンスはこちらになります。

作成したページで、Examplesの[Restaurant Customer Inquiries](#)（の一部）を呼び出してみます。



以下、作り方になります。

Co.Embedを呼び出すページを作成します。

ページの作成を実行します。



空白ページを選択します。



ページ番号は4、名前はEmbedとします。

ページの作成をクリックします。



Co.Embedはテキストの配列を入力としているため、対話グリッドを使ってテキストの配列を作成します。テキストの配列はAPEXコレクションEMBED_TEXTSに保持します。

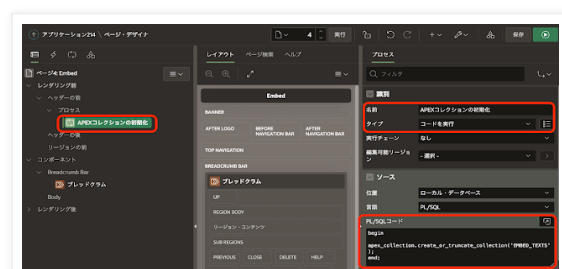
レンダリング前、ヘッダーの前にプロセスを作成し、APEXコレクションEMBED_TEXTSを初期化します。

識別の名前はAPEXコレクションの初期化とします。タイプはコードを実行、ソースのPL/SQLコードとして以下を記述します。

```
begin
    apex_collection.create_or_truncate_collection('EMBED_TEXTS');
end;
```

initialize-embed-texts.sql hosted with ❤ by GitHub

[view raw](#)



対話グリッドを作成します。

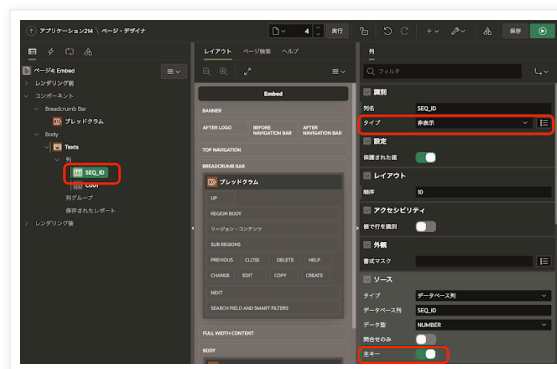
識別の名前は**Texts**、**タイプ**として**対話グリッド**を選択します。**ソースのタイプ**として**SQL問合せ**、**SQL問合せ**として以下を記述します。

```
select seq_id, c001 from apex_collections where collection_name = 'EMBED_TEXTS'
```

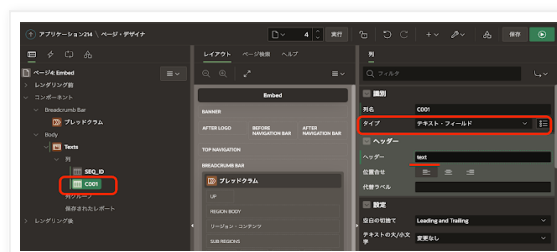


APEXコレクションでは列SEQ_IDが一意列です。そのため、対話グリッドが列SEQ_IDを主キーとして扱うように設定します。

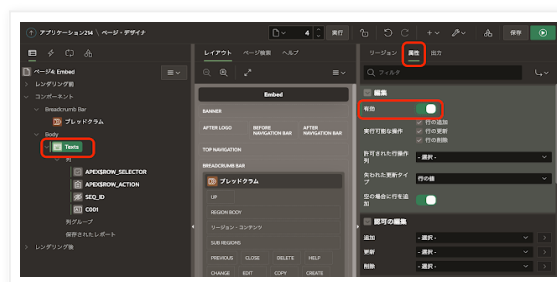
列SEQ_IDを選択し、識別の**タイプ**を**非表示**に変更します。**ソースの主キー**を**オン**にします。



列C001を選択し、識別の**タイプ**を**テキスト・フィールド**に変更します。**ヘッダー**は**text**にします。



対話グリッド**Texts**の**属性**を開き、**編集の有効**を**オン**に変更します。



プロセス・ビューを開き、**プロセスTexts - 対話グリッド・データの保存**を選択します。

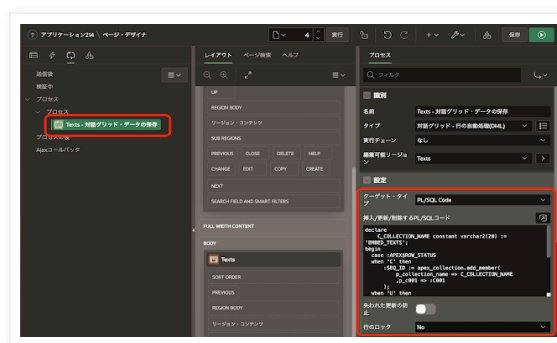
設定のターゲット・タイプをPL/SQL Codeに変更し、挿入/更新/削除するPL/SQLコードに以下を記述します。

```
declare
    C_COLLECTION_NAME constant varchar2(20) := 'EMBED_TEXTS';
begin
    case :APEX$ROW_STATUS
    when 'C' then
        :SEQ_ID := apex_collection.add_member(
            p_collection_name => C_COLLECTION_NAME
            ,p_c001 => :C001
        );
    when 'U' then
        apex_collection.update_member(
            p_collection_name => C_COLLECTION_NAME
            ,p_c001 => :C001
            ,p_seq => :SEQ_ID
        );
    when 'D' then
        apex_collection.delete_member(
            p_collection_name => C_COLLECTION_NAME
            ,p_seq => :SEQ_ID
        );
    end case;
end;
```

embed-texts.sql hosted with ❤ by GitHub

[view raw](#)

失われた更新の防止はオフ、行のロックはNoにします。

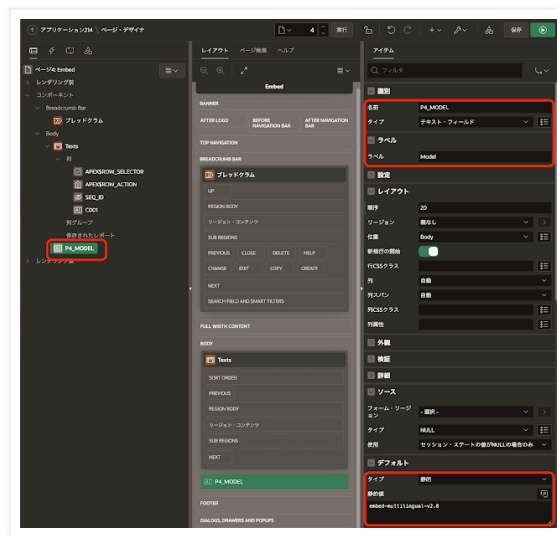


対話グリッドの設定は以上で完了です。

Co.EmbedのAPIの入力のひとつであるmodelを指定するページ・アイテムを作成します。

識別の名前はP4_MODEL、タイプはテキスト・フィールド、ラベルはModelとします。

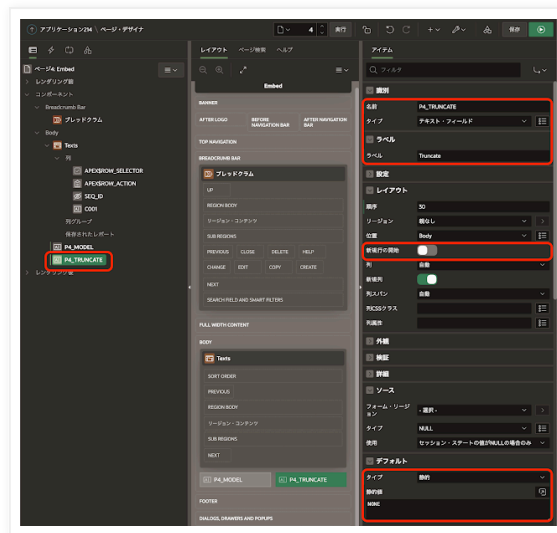
デフォルトのタイプに静的を選択し、静的値としてembed-multilingual-v2.0を設定します。



APIのもうひとつの入力truncateを指定するページ・アイテムを作成します。

識別の名前はP4_TRUNCATE、タイプはテキスト・フィールド、ラベルはTruncateとします。レイアウトの新規行の開始をオフにし、P4_MODELの右に配置します。

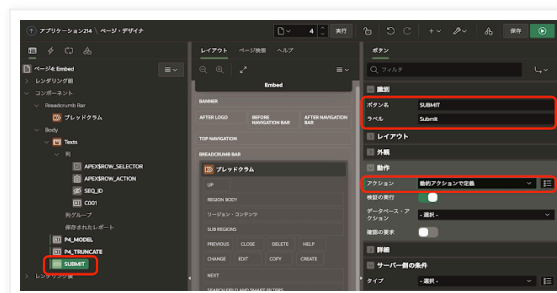
デフォルトのタイプに静的を選択し、静的値としてNONEを設定します。



CohereのCo.Embedを呼び出すボタンを作成します。

識別のボタン名はSUBMIT、ラベルは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。

動的アクションは、Co.Embedを呼び出すリージョンを作成した後に定義します。



Co.Embedを呼び出し、受け取ったレスポンスを表示するリージョンを作成します。

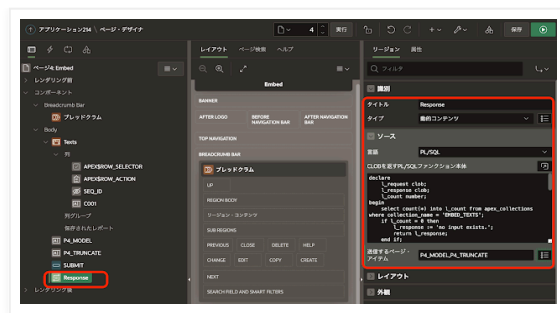
識別の**タイトル**は**Response**、**タイプ**として**動的コンテンツ**を選択します。ソースの**CLOB**を返す**PL/SQLファンクション**本体として、以下を記述します。

```
declare
    l_request clob;
    l_response clob;
    l_count number;
begin
    select count(*) into l_count from apex_collections where collection_name = 'EMBED_TEXTS';
    if l_count = 0 then
        l_response := 'no input exists.';
        return l_response;
    end if;
    -- Co.Embedへのリクエストを生成する。
    select json_object(
        key 'texts' value
        (
            select json_arrayagg(c001 order by seq_id) from apex_collections
            where collection_name = 'EMBED_TEXTS'
        )
        -- モデルを指定する。
        ,key 'model' value :P4_MODEL
        -- トランケートを指定する。
        ,key 'truncate' value :P4_TRUNCATE
    returning clob)
    into l_request
    from dual;
    -- Cohere Co.Embedを呼び出す。
    -- https://docs.cohere.com/reference/embed
    apex_debug.info(l_request);
    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 := apex_web_service.make_rest_request(
        p_url => 'https://api.cohere.ai/v1/embed'
        ,p_http_method => 'POST'
        ,p_body => l_request
        ,p_credential_static_id => 'COHERE_API'
    );
    return l_response;
end;
```

call-co-embed-cohere.sql hosted with ❤ by GitHub

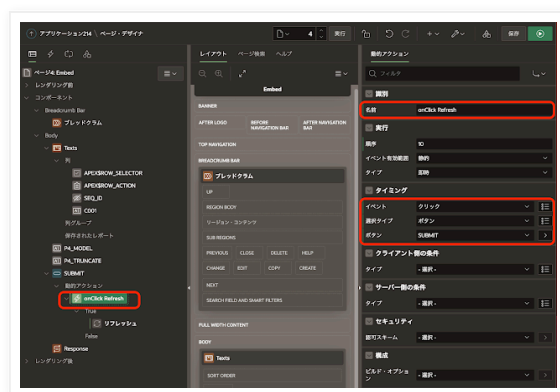
[view raw](#)

送信するページ・アイテムとしてP4_MODEL,P4_TRUNCATEを指定します。

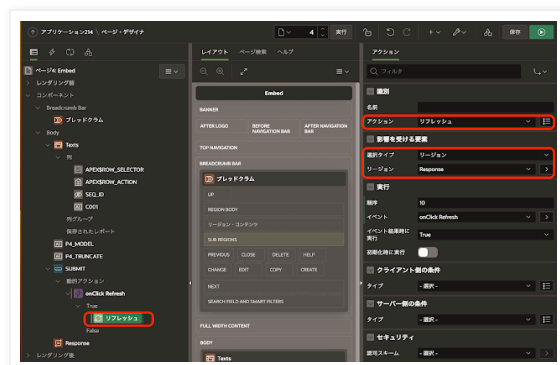


ボタンSUBMITの動的アクションを作成します。

識別の名前はonClick Refreshとします。タイミングのイベントは、ボタンのデフォルトであるクリックです。



TRUEアクションはリフレッシュ、影響を受ける要素の選択タイプはリージョン、リージョンとしてResponseを選択します。



以上でCo.Embedを呼び出すページは完成です。

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。

オラクル・データベースでpgvectorが提供するような、embeddingを保存してvector similarity searchを実行する方法を見つけることはできませんでした。

完

[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.