

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。

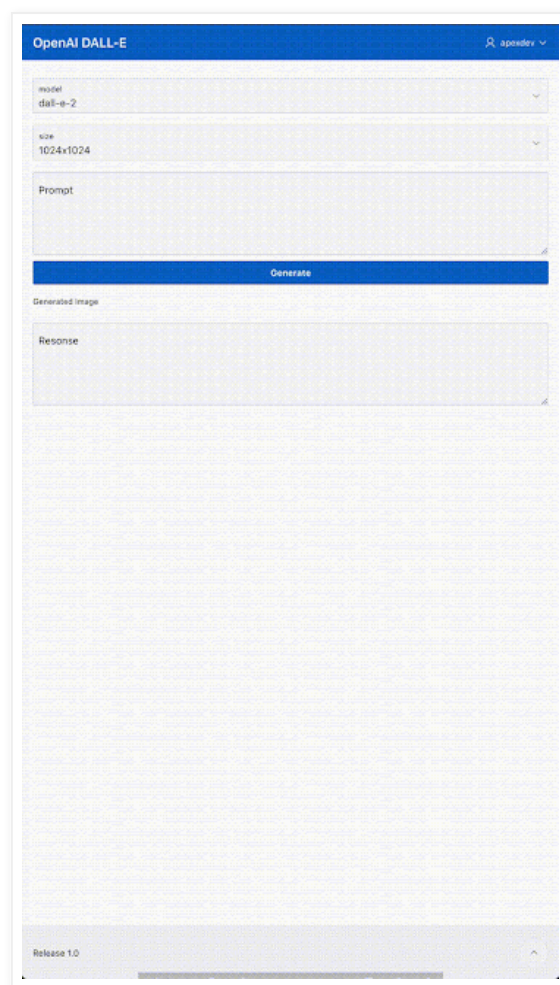
2024年1月26日 金曜日

OpenAIのDALL-Eを呼び出して画像を生成するAPEXアプリケーションを作る

OpenAIのDALL-Eを呼び出して画像を生成するAPEXアプリケーションを作ってみました。

アプリケーションを動かすには、OpenAIのAPIを呼び出すためのAPIキーをAPEXの**Web資格証明**として作成済みであることと、DALL-EのAPI呼び出しは課金されるので、購入したクレジットが残っている必要があります。

作成したAPEXアプリケーションは以下のように動作します。

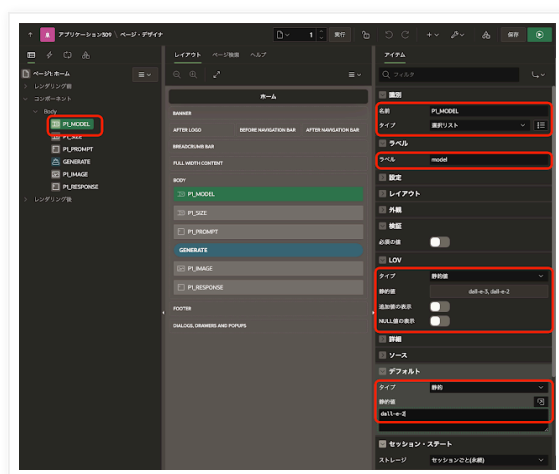


空のAPEXアプリケーションを作成し、デフォルトで作成されるホーム・ページに機能を実装しています。

以下より作成したアプリケーションを簡単に紹介します。

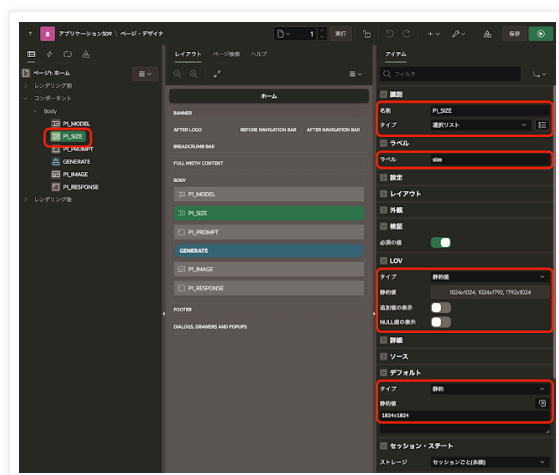
モデルを選択するページ・アイテムP1_MODELを作成しています。タイプは**選択リスト**、ラベルは**model**です。

LOVは**タイプ**を**静的値**として、**dall-e-2**と**dall-e-3**を選択できるようにしています。デフォルトは**dall-e-2**にしています。

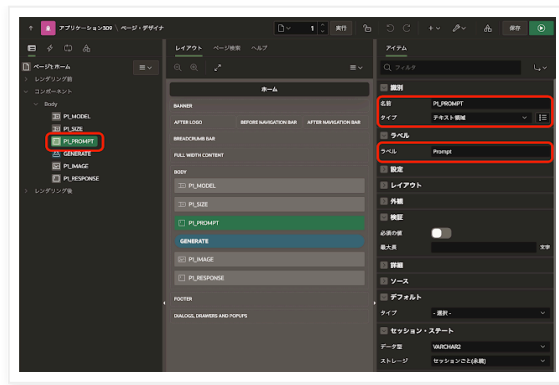


サイズを選択するページ・アイテムP1_SIZEも同様に、LOVの**タイプ**を**静的値**として、**1024x1024**、**1024x1792**、**1792x1024**のどれかを選択するようにしています。デフォルトは**1024x1024**です。

この他にも**指定可能なパラメータ** (n、quality、styleなど) はありますが、これだけ設定していれば最低限の画像は生成できるので、他は省略します。



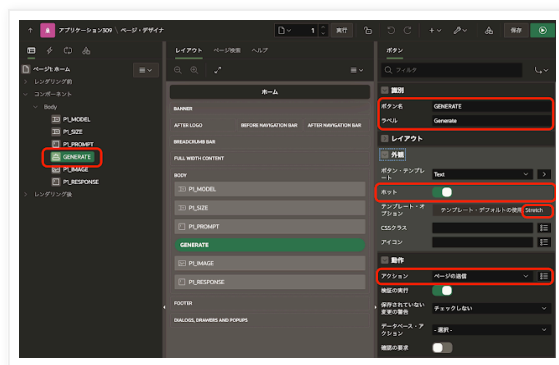
プロンプトを入力するページ・アイテムP1_PROMPTを作成します。ラベルは**Prompt**、タイプは**テキスト領域**を選択します。



DALL-Eによる画像生成を呼び出すボタン**GENERATE**を作成します。

外観のホットをオンにし、テンプレート・オプションのWidthにStretchを選択します。

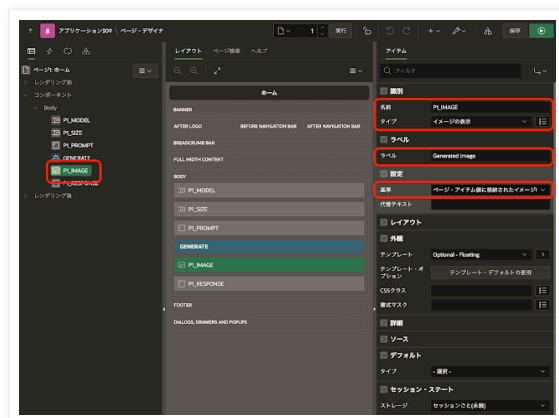
動作のアクションはデフォルトのページの送信とします。



OpenAIのCreate Image APIは、リクエストのresponse_formatで明示的にb64_jsonを指定していないときは、生成された画像が置かれている位置のURLを返します。

返されたURLを保持するページ・アイテムP1_IMAGEを作成します。タイプにイメージの表示を選択します。

設定の基準にページ・アイテム値に格納されたイメージURLを選択することにより、ページ・アイテムの値であるURLより取得できる画像をページに表示します。



デバッグ用にCreate Image APIのレスポンスをそのまま表示するページ・アイテムを作成します。

識別の名前はP1_RESPONSE、タイプはテキスト領域とします。

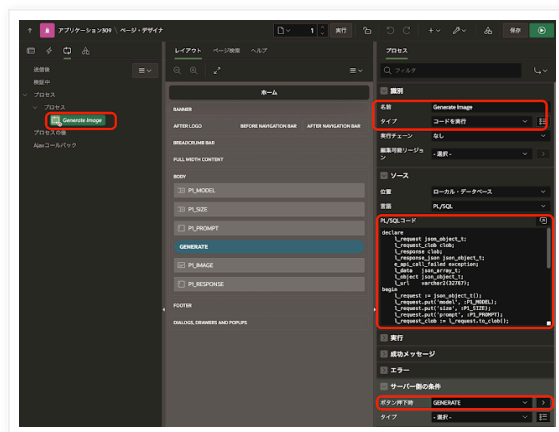


ボタン**GENERATE**がクリックされたときに、OpenAIのDALL-Eを呼び出す**プロセス**を作成します。

識別の名前は**Generate Image**とします。タイプは**コードを実行**です。ソースの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_json json_object_t;
    e_api_call_failed exception;
    l_data json_array_t;
    l_object json_object_t;
    l_url varchar2(32767);
begin
    l_request := json_object_t();
    l_request.put('model', :P1_MODEL);
    l_request.put('size', :P1_SIZE);
    l_request.put('prompt', :P1_PROMPT);
    l_request_clob := l_request.to_clob();
    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 => :G_REQUEST_URL
        ,p_http_method => 'POST'
        ,p_body => l_request_clob
        ,p_credential_static_id => :G_CREDENTIAL
    );
    if apex_web_service.g_status_code <> 200 then
        raise e_api_call_failed;
    end if;
    :P1_RESPONSE := l_response;
    l_response_json := json_object_t(l_response);
    l_data := l_response_json.get_array('data');
    l_object := treat(l_data.get(0) as json_object_t);
    l_url := l_object.get_string('url');
    :P1_IMAGE := l_url;
end;
```

サーバー側の条件のボタン押下時にGENERATEを指定します。



コードに含まれるバインド変数G_REQUEST_URLとG_CREDENTIALの値を、アプリケーション定義の置換に設定します。

G_CREDENTIALには、OpenAIのAPIキーが設定されたWeb資格証明の静的IDを設定します。
G_REQUEST_URLはAPIのエンドポイントである以下のURLを設定します。

<https://api.openai.com/v1/images/generations>



以上でアプリケーションは完成です。

今回作成した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/sample-openai-dall-e.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: 15:59

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.
