

日々是Oracle APEX

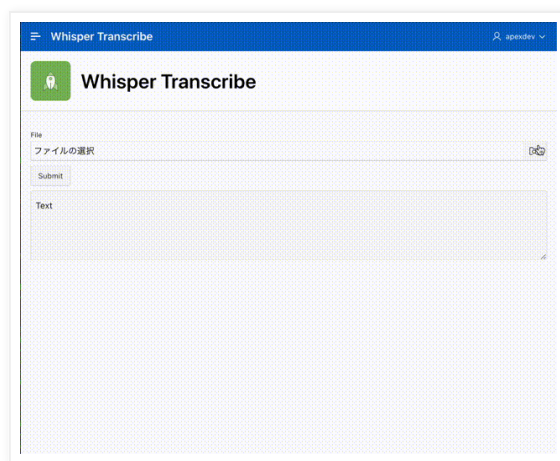
Oracle APEXを使った作業をしていて、気の付いたところを忘れないようにメモをとります。

2023年1月24日 火曜日

OpenAI Whisperを使った文字起こしアプリの作成(4) - APEXアプリの作成

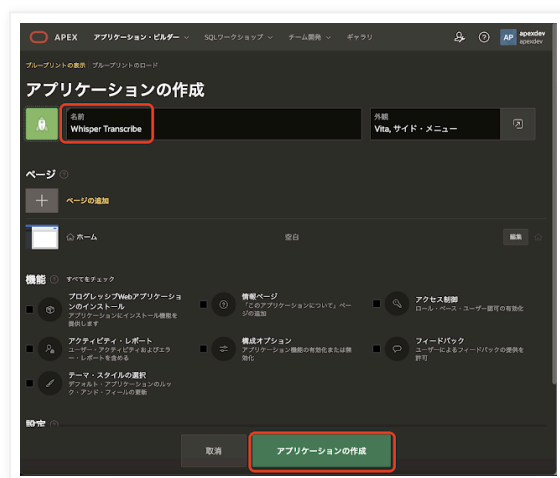
Whisperによる文字起こしを行うAPIが作成できました。このAPIを呼び出すAPEXアプリケーションを作成します。

作成したアプリケーションは、以下のように動作します。



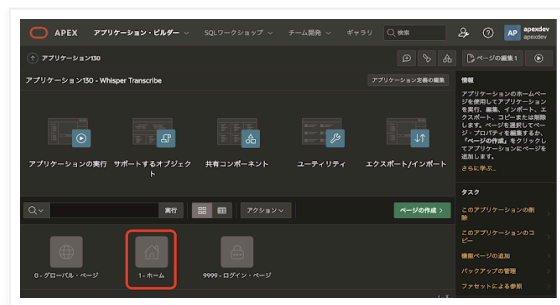
アプリケーション作成ウィザードを起動します。

名前は**Whisper Transcribe**とします。すべての機能はデフォルトで作成されるホーム・ページに実装します。そのため、設定は変更せずに**アプリケーションの作成**を実行します。



アプリケーションが作成されます。

ページ・デザイナーにてホーム・ページを開きます。



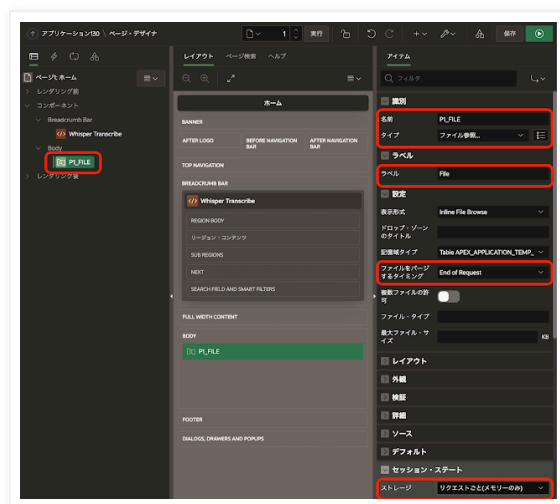
最初に音声ファイルを指定するページ・アイテムを作成します。

Bodyにページ・アイテムを作成します。

識別の名前はP1_FILE、タイプとしてファイル参照...を選択します。

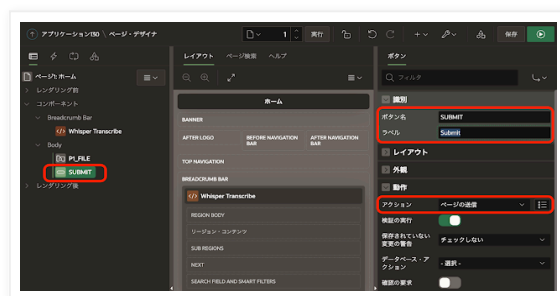
設定のファイルをパージするタイミングとして、End of Requestを選択します。セッション・スタートのストレージとしてリクエストごと(メモリーのみ)を選択します。

音声ファイルのアップロードとAPI呼び出しによる文字起こしは、1回のHTTPリクエストで完結します。そのため、アップロードした音声ファイルを処理完了後に保持する必要はありません。



音声ファイルのアップロードと文字起こしを実行するボタンを作成します。

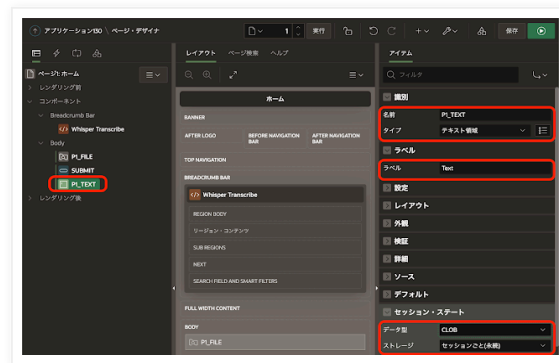
作成したボタンのボタン名はSUBMIT、ラベルはSubmitとします。動作のアクションはデフォルトでページの送信になります。



文字起こしの結果を保持するページ・アイテムを作成します。

作成したページ・アイテムの識別の名前はP1_TEXT、タイプとしてテキスト領域を選択します。ラベルはTextとします。

セッション・ステートのデータ型はCLOB、ストレージとしてセッションごと(永続)を選択します。
このように設定すると、ページをリロードしても文字起こしの結果が維持されます。



ボタンSUBMITをクリックしたときに、文字起こしのAPIを呼び出す処理を実装します。

以下のコードを実行し、パッケージUTL_WHISPERを作成します。

```
create or replace package utl_whisper
as
/**
 * 音声データをアップロードし、Whisperの文字起こしを実装したAPIを呼び出す。
 *
 * @param p_url Whisper APIのエンドポイントURL
 * @param p_file APEX_APPLICATION_TEMP_FILESのnameとなるファイル名
 * @param p_text Whisperによって、音声より文字起こしされた文字列。
 */
procedure transcribe(
    p_url in varchar2
    ,p_file in varchar2
    ,p_text out clob
);
end utl_whisper;
/

create or replace package body utl_whisper
as
procedure transcribe(
    p_url in varchar2
    ,p_file in varchar2
    ,p_text out clob
)
as
    l_filename apex_application_temp_files.filename%type;
    l_blob_content blob;
    l_multipart apex_web_service.t_multipart_parts;
    l_multipart_request blob;
    l_response clob;
    l_response_json json_object_t;
```

```

l_segments json_array_t;
l_count    pls_integer;
l_segment  json_object_t;
l_temp_clob clob;
l_text     varchar2(32767);

begin
    /* APEXにアップロードされた音声データをBLOBに取り出す。 */
    select filename, blob_content into l_filename, l_blob_content
    from apex_application_temp_files
    where name = p_file;
    /*
    * 取り出した音声データをmultipart/form-dataとして、WhisperのAPIを呼び出す。
    */
    apex_web_service.clear_request_headers;
    /* WhisperのAPI側ではファイル名もMIMEタイプも確認していない。 */
    apex_web_service.append_to_multipart(
        p_multipart => l_multipart
        ,p_name => 'file'
        ,p_filename => l_filename
        ,p_content_type => 'application/octet-stream'
        ,p_body_blob => l_blob_content
    );
    l_multipart_request := apex_web_service.generate_request_body(l_multipart);
    l_response := apex_web_service.make_rest_request(
        p_url => p_url
        ,p_http_method => 'POST'
        ,p_body_blob => l_multipart_request
    );
    /* デバッグでJSON自体を見たいときは、l_responseを確認する。 */
    -- p_text := l_response;
    l_response_json := json_object_t(l_response);
    /*
    * 返されたJSONデータより、文字起こしされたテキストを取り出す。
    */
    /* そのまま取り出しても、文字起こしの結果としては同じ。 */
    -- p_text := l_response_json.get_string('text');
    /* segmentsの配列から取り出す、手間のかかる方法。 */
    l_segments := l_response_json.get_array('segments');
    l_count := l_segments.get_size();
    dbms_lob.createTemporary(l_temp_clob, false, DBMS_LOB.CALL);
    for i in 0..(l_count - 1)
    loop
        l_segment := treat(l_segments.get(i) as json_object_t);
        l_text := l_segment.get_string('text');
        dbms_lob.writeappend(
            lob_loc => l_temp_clob
            ,amount => length(l_text)

```

```

        ,buffer => l_text
    );
end loop;
p_text := l_temp_clob;
dbms_lob.freeTemporary(l_temp_clob);
end transcribe;
end utl_whisper;
/

```

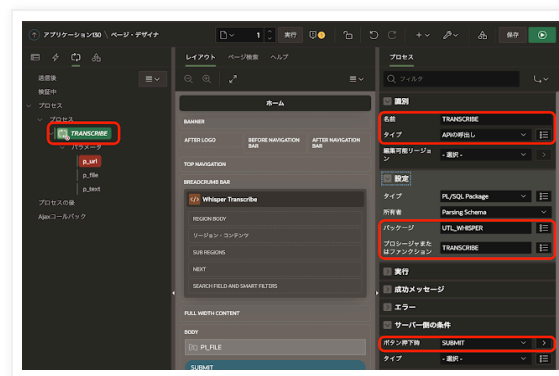
utl_whisper.sql hosted with ❤ by GitHub

[view raw](#)

左ペインで**プロセス・ビュー**を開き、文字起こしのAPIを呼び出すプロセスを作成します。

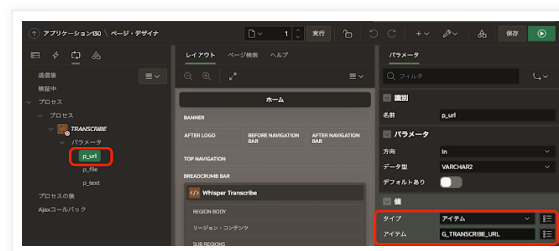
作成したプロセスの**識別の名前**は**TRANSCRIBE**、**タイプ**として**APIの呼出し**を選択します。**設定のパッケージ**として**UTL_WHISPER**、**プロシージャ**または**ファンクション**として**TRANSCRIBE**を選択します。

サーバー側の条件のボタン押下時に**SUBMIT**を選択し、ボタンが押されたときに**UTL_WHISPER**のプロシージャ**TRANSCRIBE**が呼び出されるようにします。



パラメータの**p_file**、**p_text**はデフォルトでページ・アイテム**P1_FILE**、**P1_TEXT**が割り当たります。

パラメータ**p_url**は、**値のタイプ**を**アイテム**、**アイテム**として**G_TRANSCRIBE_URL**を指定します。**G_TRANSCRIBE_URL**は置換文字列として、この後に値を設定します。



アプリケーション定義の置換を開き、置換文字列を**G_TRANSCRIBE_URL**として、**置換値**にWhisperによる文字起こしを行う**APIを実装したURL**を設定します。



以上でアプリケーションは完成です。アプリケーションを実行すると、記事の先頭にあるGIF動画のように動作します。

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/whisper-transcribe.zip>

無料の範囲だと処理に時間がかかるため、実用レベルで使用するのには難しいと思います。とはいえ、どのように使えるのかという評価はできるのではないのでしょうか。

次の記事で、JavaScriptから直接Whisperを呼び出すような実装を追加します。

続く

Yuji N. 時刻: 11:20

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.