

日々是Oracle APEX

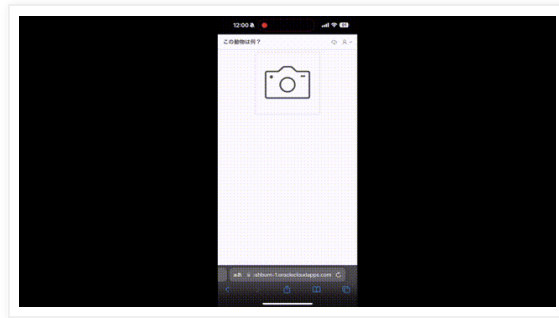
Oracle APEXを使った作業をしていて、気の付いたところを忘れないようにメモをとります。

2024年1月29日月曜日

OpenAIのText to speech APIを呼び出して写真の説明を読み上げる

前回の記事「[写真の動物の説明をOpenAI GPT-4 Turbo with visionにしまおう](#)」で作成したAPEXアプリケーションでは、写真の説明を文章として表示していました。この文章を、OpenAIのText to speech APIを呼び出して、音声に変換して読み上げるようにします。

作成したアプリケーションは以下のように動作します。



ORDSのREST APIにテンプレートとしてanimal-speechを作成し、POSTハンドラに以下のコードを記述します。パッケージUTL_OPENAI_VISIONに、OpenAIのspeechエンドポイントを呼び出すプロシージャとしてSPEECHを追加し、それをコード中で呼び出しています。

```
declare
    l_response clob;
    l_role varchar2(16);
    l_content clob;
    l_usage clob;
    l_prompt clob;
    l_input clob;
    l_audio blob;
begin
    l_prompt := 'この画像に写っている動物の名前と生息地および生態を教えてください。小学生にわかるような文章でお願いします';
    utl_openai_vision.chat_completions(
        p_text => l_prompt
        ,p_image => :body
        ,p_mimetype => :content_type
        ,p_max_tokens => 1000
        ,p_role => l_role
        ,p_content => l_content
```

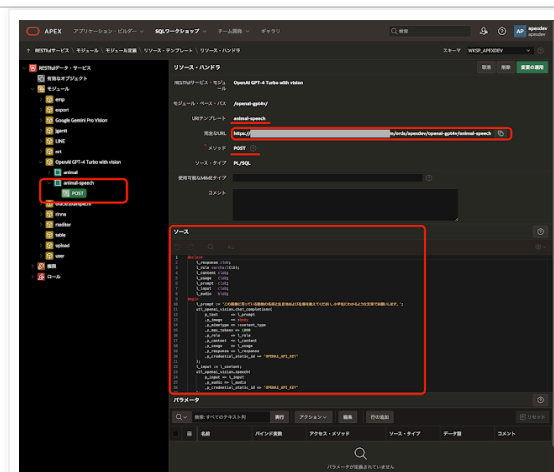
```

,p_usage      => l_usage
,p_response   => l_response
,p_credential_static_id => 'OPENAI_API_KEY'
);
l_input := l_content;
utl_openai_vision.speech(
    p_input => l_input
    ,p_audio => l_audio
    ,p_credential_static_id => 'OPENAI_API_KEY'
);
/* audio/mp3のファイルを返す。*/
:status := 200;
sys.http.init;
sys.http.p('Content-Length: ' || dbms_lob.getlength(l_audio));
sys.http.p('Content-Type: audio/mp3');
sys.http.p('Content-Disposition: attachment; filename=speech.mp3');
sys.owa_util.http_header_close;
sys.wpg_docload.download_file(l_audio);
exception
when others then
    if l_audio is not null then
        l_response := apex_util.blob_to_clob(l_audio);
    end if;
    http.p(l_response);
    :status := 400;
end;

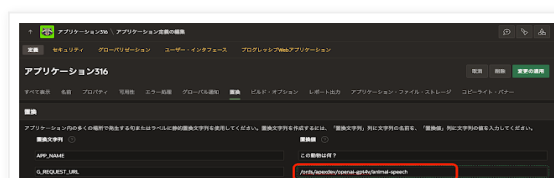
```

call-openai-gpt4v-and-speech.sql hosted with ❤️ by GitHub

[view raw](#)



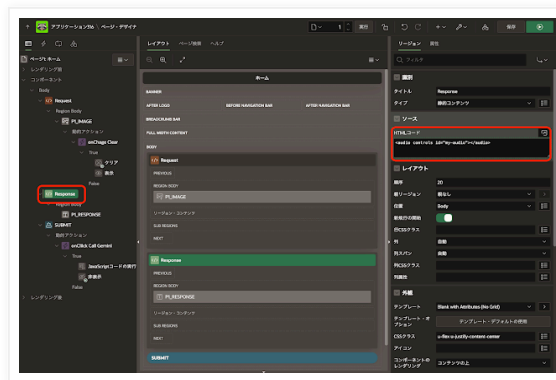
APEXアプリケーションのアプリケーション定義の置換文字列G_REQUEST_URLを、新しく作成した音声を返すRESTサービスの完全なURLに置き換えます。



リージョン**Response**に音声要素を埋め込みます。

ソースのHTMLコードに以下を記述します。

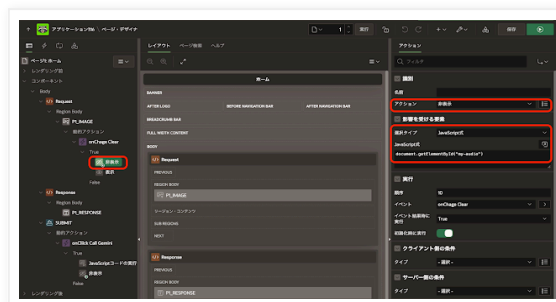
```
<audio controls id="my-audio"></audio>
```



写真の選択を変更したときに、応答の文章であるページ・アイテム**P1_RESPONSE**をクリアする代わりに、音声要素を**非表示**にします。

TRUEアクションのクリアを**非表示**に変更し、**影響を受ける要素の選択タイプ**に**JavaScript式**を選択し、**JavaScript式**として以下を記述します。

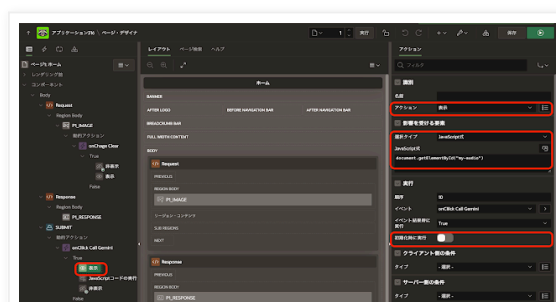
```
document.getElementById("my-audio")
```



選択タイプに**jQueryセクタ**を選択し、**jQueryセクタ**として**#my-audio**を指定することもできます。

ボタン**SUBMIT**（ラベルは**問い合わせる**）を押した時に、音声要素を**表示**する**TRUEアクション**を追加します。

TRUEアクションは**表示**、**影響を受ける要素**は先ほどの非表示のアクションと同じです。ページが最初に表示されるときは音声要素は表示しないため、**初期化時に実行はオフ**にします。



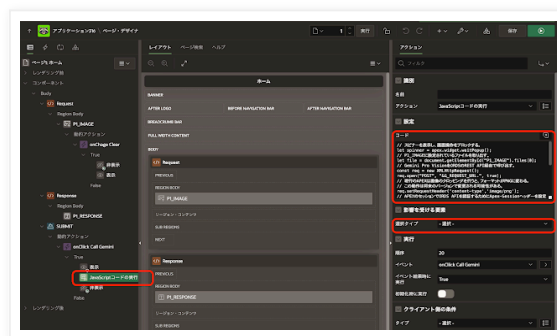
TRUEアクションのJavaScriptコードの実行の設定のコードを、以下に記述に変更します。macOSのChromeで実行するとelem.play()を呼び出した時点で再生が開始されますが、iPhoneのSafariでは、elem.play()では再生されないようです。再生ボタンをクリックする必要がありました。

```
// スピナーを表示し、画面操作をブロックする。
let spinner = apex.widget.waitPopup();
// P1_IMAGEに設定されているファイルを取り出す。
let file = document.getElementById("P1_IMAGE").files[0];
// OPenAI GPT-4VをORDSのREST API経由で呼び出す。
const req = new XMLHttpRequest();
req.open("POST", "&G_REQUEST_URL.", true);
// 現行のAPEXは画像のクロッピングを行うと、フォーマットがPNGに変わる。
// この動作は将来のバージョンで変更される可能性がある。
req.setRequestHeader('content-type', 'image/png');
// APEXのセッションでORDS APIを認証するためにApex-Sessionヘッダーを設定している。
const authHeader = apex.env.APP_ID.concat('-', apex.env.APP_SESSION);
req.setRequestHeader('Apex-Session', authHeader);
// text to speechを呼び出すので出力はaudio/mp3
req.responseType = "blob";
req.onload = (event) => {
    const audio = req.response;
    const audioURL = URL.createObjectURL(audio);
    const elem = document.getElementById("my-audio");
    elem.src = audioURL;
    // スピナーを解除して解説を読み上げ。
    spinner.remove();
    elem.play();
};
// 要求の送信。
req.send(file);
```

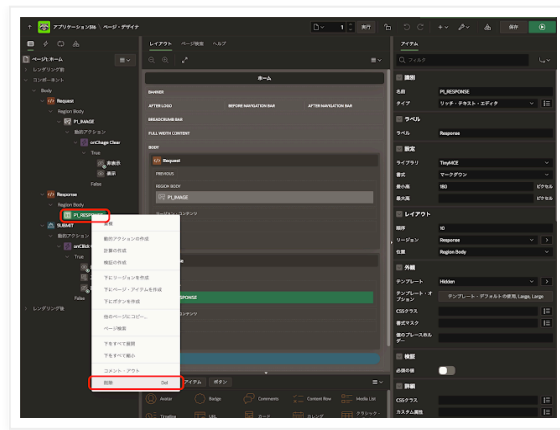
call-openai-thru-ords.js hosted with ❤ by GitHub

[view raw](#)

影響を受ける要素の選択タイプを- 選択 -に戻し、無指定にします。



ページ・アイテムP1_RESPONSEを削除します。



以上で写真の説明を音声として読み上げる実装ができました。

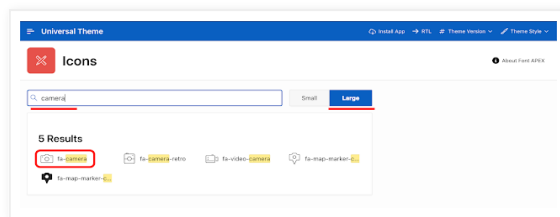
少しだけAPEXアプリケーションを改良します。

ページ・アイテムP1_IMAGEに表示されているアイコンを変更してみます。

FontAPEXのページを開きます。

<https://apex.oracle.com/fontapex>

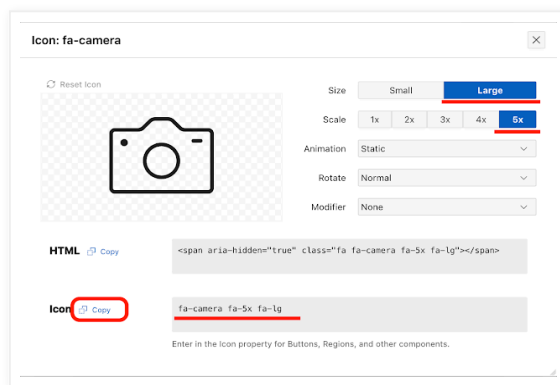
cameraのアイコンを検索します。fa-cameraを開きます。



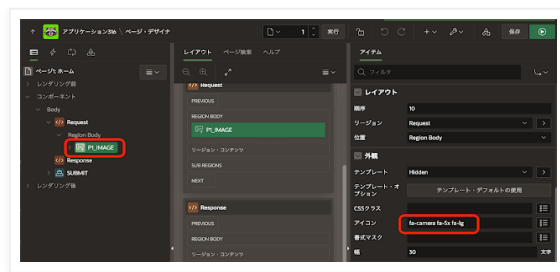
表示したいアイコンを修飾し、HTMLまたはCSSとしての記述を取得します。今回は、SizeはLarge、Scaleは5xを選択します。Iconの記述は以下になります。

fa-camera fa-5x fa-lg

Iconの記述をコピーします。



ページ・アイテムP1_IMAGEの外観のアイコンに、コピーした文字列を設定します。



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

OpenAIのText-to-speechのドキュメントに記載があるように、現在は英語に最適化されているようです。

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/text-to-speech.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: 13:00

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.