

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。

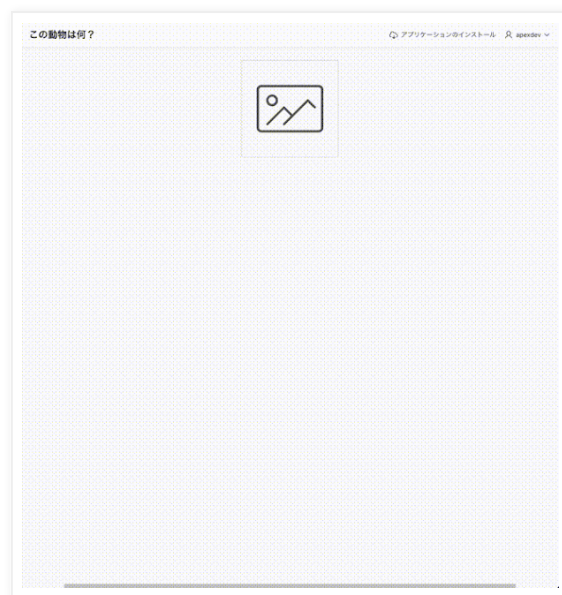
2024年1月26日 金曜日

写真の動物の説明をOpenAI GPT-4 Turbo with visionにしてみよう

以前にこちらの記事「Google Gemini Pro Visionを呼び出して写真の動物を説明してもらうアプリを作る」で作成したAPEXアプリケーションの問い合わせ先を、Google Gemini Pro VisionからOpenAI GPT-4 Turbo with visionに変えてみました。

APEXアプリケーション自体は変更せず、ORDS REST APIの呼び出し先だけを変更します。

PCでアプリケーションを実行してみました。



OpenAI GPT-4 Turbo with visionを呼び出すパッケージUTL_OPENAI_VISIONを作成します。パッケージのコードは以下になります。

```
create or replace package utl_openai_vision
as
  C_REQUEST_URL constant varchar2(80) := 'https://api.openai.com/v1/chat/completions';
  C_MODEL        constant varchar2(20) := 'gpt-4-vision-preview';
  C_TTS_URL       constant varchar2(80) := 'https://api.openai.com/v1/audio/speech';
  C_TTS_MODEL     constant varchar2(10) := 'tts-1';
  C_TTS_MODEL_HD constant varchar2(10) := 'tts-1-hd';
  C_TTS_VOICE_ALLOY constant varchar2(10) := 'alloy';
  C_TTS_VOICE_ECHO constant varchar2(10) := 'echo';
  C_TTS_VOICE_FABLE constant varchar2(10) := 'fable';
  C_TTS_VOICE_ONYX constant varchar2(10) := 'onyx';
```

```
C_TTS_VOICE_NOVA    constant varchar2(10) := 'nova';
C_TTS_VOICE_SHIMMER constant varchar2(10) := 'shimmer';
```

```
/**
 * GPT-4 Turbo with visionを呼び出して、画像を説明してもらう。
 */
```

```
procedure chat_completions(
    p_text          in varchar2
    ,p_image        in blob
    ,p_mimetype     in varchar2
    ,p_max_tokens   in number default 300
    ,p_role         out varchar2
    ,p_content      out clob
    ,p_usage        out clob
    ,p_response     out clob
    ,p_credential_static_id in varchar2
);
```

```
/**
 * text to speechを呼び出して音声ファイルを取得する。
 */
```

```
procedure speech(
    p_input         in clob
    ,p_model        in varchar2 default C_TTS_MODEL
    ,p_voice        in varchar2 default C_TTS_VOICE_NOVA
    ,p_mimetype     in varchar2 default 'audio/mp3'
    ,p_audio        out blob
    ,p_credential_static_id in varchar2
);
```

```
end utl_openai_vision;
```

```
/
```

```
create or replace package body utl_openai_vision
as
```

```
procedure chat_completions(
    p_text          in varchar2
    ,p_image        in blob
    ,p_mimetype     in varchar2
    ,p_max_tokens   in number default 300
    ,p_role         out varchar2
    ,p_content      out clob
    ,p_usage        out clob
    ,p_response     out clob
    ,p_credential_static_id in varchar2
```

```
)
```

```
as
```

```

l_request json_object_t;
l_request_clob clob;
l_messages json_array_t;
l_message json_object_t;
l_content json_array_t;
l_prompt json_object_t;
l_image json_object_t;
l_response clob;
l_response_json json_object_t;
e_call_api_failed exception;
l_choices json_array_t;
l_choice json_object_t;
l_usage json_object_t;

```

begin

```

/* プロンプトの準備 */
l_prompt := json_object_t();
l_prompt.put('type', 'text');
l_prompt.put('text', p_text);
/* イメージの準備 - base64で送信する */
l_image := json_object_t();
l_image.put('type', 'image_url');
l_image.put('image_url', 'data:' || p_mimetype || ';base64,'
|| apex_web_service.blob2clobbase64(p_image, 'N', 'N')
);
/* contentの作成 */
l_content := json_array_t();
l_content.append(l_prompt);
l_content.append(l_image);
/* messageの作成 */
l_message := json_object_t();
l_message.put('role', 'user');
l_message.put('content', l_content);
/* messagesの作成 */
l_messages := json_array_t();
l_messages.append(l_message);
/* requestの作成 */
l_request := json_object_t();
l_request.put('model', C_MODEL);
l_request.put('messages', l_messages);
l_request.put('max_tokens', p_max_tokens);
/* call OpenAI chat completions api */
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 => C_REQUEST_URL
    ,p_http_method => 'POST'

```

```

        ,p_body => l_request_clob
        ,p_credential_static_id => p_credential_static_id
    );
    /* process response */
    p_response := l_response;
    if apex_web_service.g_status_code <> 200 then
        raise e_call_api_failed;
    end if;
    l_response_json := json_object_t(l_response);
    l_choices := l_response_json.get_array('choices');
    for i in 1..l_choices.get_size()
    loop
        l_choice := treat(l_choices.get(i-1) as json_object_t);
        l_message := l_choice.get_object('message');
        p_role     := l_message.get_string('role');
        p_content  := p_content || l_message.get_clob('content');
    end loop;
    l_usage := l_response_json.get_object('usage');
    p_usage := l_usage.to_clob();
end chat_completions;

procedure speech(
    p_input      in clob
    ,p_model     in varchar2
    ,p_voice     in varchar2
    ,p_mimetype  in varchar2
    ,p_audio     out blob
    ,p_credential_static_id in varchar2
)
as
    l_request json_object_t;
    l_request_clob clob;
    e_call_api_failed exception;
begin
    /* requestの作成 */
    l_request := json_object_t();
    l_request.put('model', p_model);
    l_request.put('input', p_input);
    l_request.put('voice', p_voice);
    /* call OpenAI text to speech api */
    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);
    apex_web_service.set_request_headers('Accept', p_mimetype, p_reset => false);
    p_audio := apex_web_service.make_rest_request_b(
        p_url => C_TTS_URL
        ,p_http_method => 'POST'
    );
end;

```

```

        ,p_body => l_request_clob
        ,p_credential_static_id => p_credential_static_id
    );
    if apex_web_service.g_status_code <> 200 then
        raise e_call_api_failed;
    end if;
end speech;

end utl_openai_vision;
/

```

call-openai-gpt-4v.sql hosted with ❤ by GitHub

[view raw](#)

ORDS REST APIのPOSTハンドラのコードを以下に置き換えます。以下のコードでは、OpenAIのAPIを呼び出す**Web資格証明**が**OPENAI_API_KEY**として登録されていることを前提にしています。

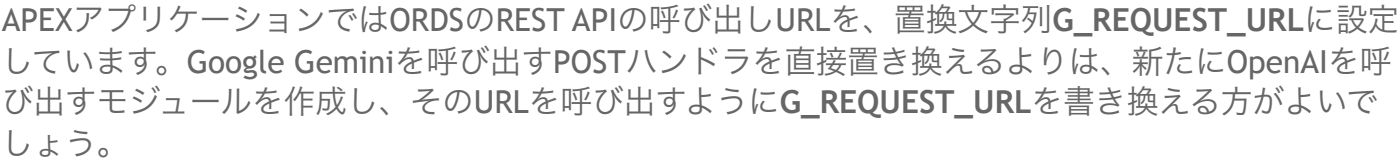
```

declare
    l_response clob;
    l_role varchar2(16);
    l_content clob;
    l_usage clob;
    l_prompt clob;
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。';
    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'
    );
    :status := 200;
    http.p(l_content);
exception
    when others then
        http.p(l_response);
        :status := 400;
end;

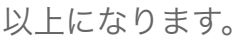
```

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

[view raw](#)



OpenAI GPT-4Vの場合、プロンプトに回答は小学生にわかるような文章でお願いします。を追加したら、それなりに柔らかい回答になりました。



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: 19:36

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](#)