

# 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3月2日 木曜日

## OpenAIのChatGPTのAPIを呼び出すAPEXアプリを作る

新しい記事で更新されています。

**OpenAIのChat Completions APIを呼び出すAPEXアプリを作成する**

<https://apexugj.blogspot.com/2024/04/chat-with-generative-ai-sample-app-0.html>

以下は過去の内容です。

OpenAIからChatGPTのAPIの提供が開始され、すでにアプリケーションに組み込んでいる所もあるようです。ChatGPTのAPIの費用をこちらのページから確認すると\$0.002 / 1K tokens tokensとなっています。

OpenAIにアカウントを登録すると最初はFree trialになり、3ヶ月を期限として\$18分だけ利用できるようです。実際にアカウントのアップグレードをせずに、ChatGPTのAPIを呼び出すことができました。

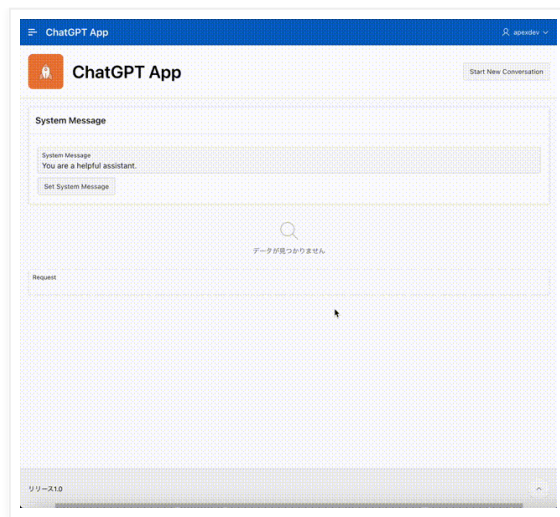
<https://help.openai.com/en/articles/4936830-what-happens-after-i-use-my-free-tokens-or-the-3-months-is-up-in-the-free-trial>

お金が絡む話ですので、試される場合はOpenAIのPricingのページをご自身で確認されるようお願いします。

作成したAPEXアプリケーションは以下の動作をします。

「去年の日本の野球の日本シリーズで優勝した球団は？」のChatGPTの回答は「去年（2020年）の日本シリーズで優勝した球団は、福岡ソフトバンクホークスです。」でした。ChatGPTが学習で使ったデータは2021年までというのは確かなようです。

続く質問「一昨年は？」の回答は「一昨年（2019年）の日本シリーズで優勝した球団は、**埼玉西武ライオンズ**です。」でした。。。



この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/chatgpt-app.zip>

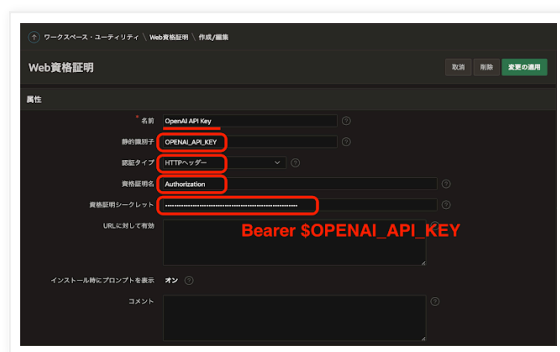
今回はアプリケーションの作成手順については省き、実装についての説明だけを行います。

OpenAIのAPIを呼び出すにあたって、**Web資格証明**を作成します。

**ワークスペース・ユーティリティ**の**Web資格証明**を開きます。ここで作成するWeb資格証明を指定して、APIを呼び出します。

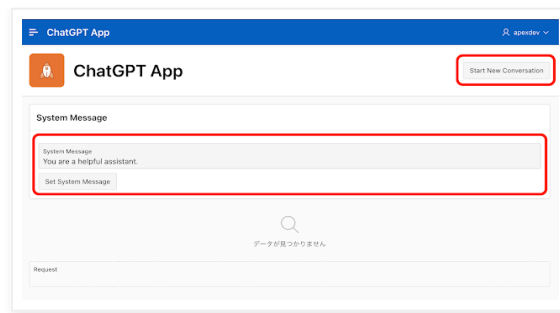
**Web資格証明**の名前は**OpenAI API Key**としています。

静的識別子として**OPENAI\_API\_KEY**、認証タイプは**HTTPヘッダー**、資格証明名は**Authorization**とします。**資格証明シークレット**として**Bearer**で始めて空白で区切った後、**OpenAI**のAPIキーを続けた文字列を設定します。



アプリケーションの実装の説明です。

ChatGPTの会話はroleがsystemのメッセージから始めます（省略も可ですが、このアプリでは省略できません）。systemのメッセージは**The system message helps set the behavior of the assistant.**のことです。



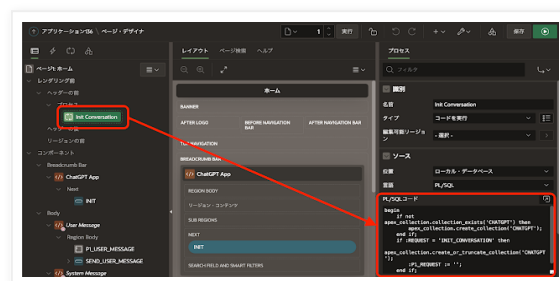
ボタン**Start New Conversation**は、ChatGPTへ送信する会話を初期化します。一連の会話はAPEXコレクション**CHATGPT**に保存しています。

会話の初期化は**レンダリング前**のプロセス**Init Conversation**として実装しています。

```
begin
  if not apex_collection.collection_exists('CHATGPT') then
    apex_collection.create_collection('CHATGPT');
  end if;
  if :REQUEST = 'INIT_CONVERSATION' then
    apex_collection.create_or_truncate_collection('CHATGPT');
    :P1_REQUEST := '';
  end if;
end;
```

init-collection.sql hosted with ❤ by GitHub

[view raw](#)



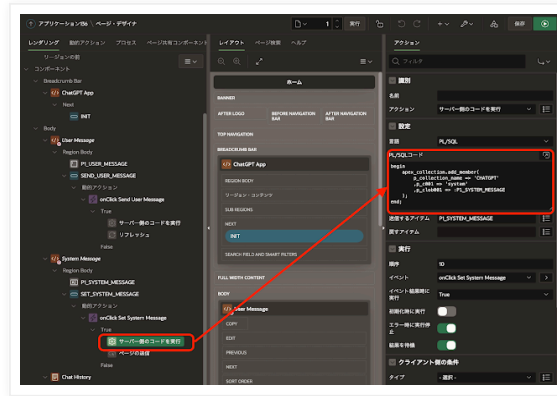
ボタン**Set System Message**は**動的アクション**で実装しています。System Messageとして入力した文字列をAPEXコレクションに追加しています。

```
begin
  apex_collection.add_member(
    p_collection_name => 'CHATGPT'
    ,p_c001 => 'system'
    ,p_clob001 => :P1_SYSTEM_MESSAGE
  );
end;
```

set-system-message.sql hosted with ❤ by GitHub

[view raw](#)

**System Message**を設定するリージョンは、APEXコレクションCHATGPTが空のときに表示されるよう、**サーバー側の条件**を設定しています。



会話履歴がある場合（systemのメッセージのみを含む）、ユーザーのメッセージを送信するリジョンが現れます。**User Message**として質問を入力し、ボタン**Send User Message**をクリックすると一連の会話をリクエストとして、ChatGPTのAPIを呼び出します。

```
declare
    l_request  json_object_t;
    l_request_clob clob;
    l_messages json_array_t;
    l_message  json_object_t;
    /*
    * OpenAIのドキュメントより、APIの応答例を拝借。
    * 参考: https://platform.openai.com/docs/guides/chat
    */
    l_response_clob clob := q'~{
'id': 'chatcmpl-6p9XYPYSTTRi0xEviKjjilqrWU2Ve',
'object': 'chat.completion',
'created': 1677649420,
'model': 'gpt-3.5-turbo',
'usage': {'prompt_tokens': 56, 'completion_tokens': 31, 'total_tokens': 87},
'choices': [
{
'message': {
'role': 'assistant',
'content': 'The 2020 World Series was played in Arlington, Texas at the Globe Life Field,
'finish_reason': 'stop',
'index': 0
}
}
]
}~';
    l_response json_object_t;
    l_choices  json_array_t;
    l_role     varchar2(80);
    l_content  clob;
    l_usage    json_object_t;
begin
    /*
```

```

* ユーザーによるメッセージをコレクションに追記する。
*/
apex_collection.add_member(
    p_collection_name => 'CHATGPT'
    ,p_c001 => 'user'
    ,p_clob001 => :P1_USER_MESSAGE
);
/*
* ChatGPTのAPIに送信するメッセージを作成する。
*/
/*
* APEXコレクションCHATGPTより、作成時刻の昇順でメッセージの配列にする。
*/
l_messages := json_array_t();
for r in (select c001, clob001 from apex_collections where collection_name = 'CHATGPT' orde
loop
    l_message := json_object_t();
    l_message.put('role' ,r.c001);
    l_message.put('content',r.clob001);
    l_messages.append(l_message);
end loop;
l_request := json_object_t();
l_request.put('model','gpt-3.5-turbo');
l_request.put('messages', l_messages);
/*
* temprature, top_pなどのパラメータを設定するとしたら、ここでputする。
*/
l_request_clob := l_request.to_clob();
/* デバッグのため、送信するメッセージをP1_REQUESTに書き込む。 */
:P1_REQUEST := l_request_clob;
/*
* OpenAIのChatGPTのAPIを呼び出す。今の所、コメントアウト。
*
* 参照: https://openai.com/blog/introducing-chatgpt-and-whisper-apis
* API Ref: https://platform.openai.com/docs/api-reference/chat
*/
apex_web_service.clear_request_headers;
apex_web_service.set_request_headers('Content-Type','application/json',p_reset => false);
l_response_clob := apex_web_service.make_rest_request(
    p_url => 'https://api.openai.com/v1/chat/completions'
    ,p_http_method => 'POST'
    ,p_body => l_request_clob
    ,p_credential_static_id => 'OPENAI_API_KEY'
);
/* ドキュメントに記載されているレスポンスで処理を継続する。 */
l_response := json_object_t(l_response_clob);
l_choices := l_response.get_array('choices');

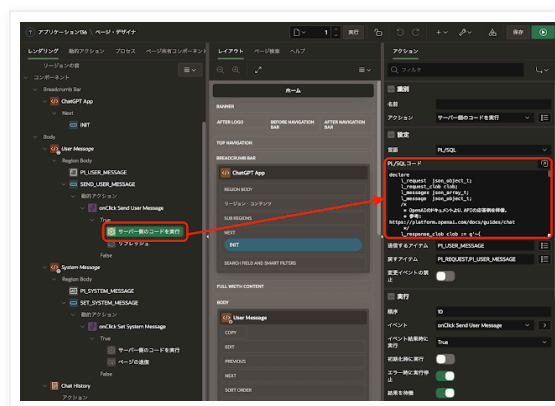
```

```

l_message := treat(l_choices.get(0) as json_object_t).get_object('message');
l_role    := l_message.get_string('role');
l_content := l_message.get_clob('content');
/* usageも取り出す。*/
l_usage := l_response.get_object('usage');
/*
 * ChatGPTからの応答をAPEXコレクションに追記する。
 */
apex_collection.add_member(
    p_collection_name => 'CHATGPT'
    ,p_c001 => l_role
    ,p_clob001 => l_content
    ,p_n001 => l_usage.get_number('prompt_tokens')
    ,p_n002 => l_usage.get_number('completion_tokens')
    ,p_n003 => l_usage.get_number('total_tokens')
);
/* ユーザーのメッセージを消去する */
:P1_USER_MESSAGE := '';
end;
```

call-chatgpi-api.sql hosted with ❤ by GitHub

[view raw](#)

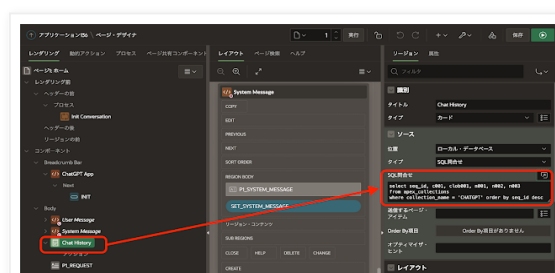


コードの説明は以上です。

ChatGPTとの会話はAPEXコレクションに保存しており、その表示にカード・リージョンを使っています。ソースのSQLは以下です。

```

select seq_id, c001, clob001, n001, n002, n003
from apex_collections
where collection_name = 'CHATGPT' order by seq_id desc
```

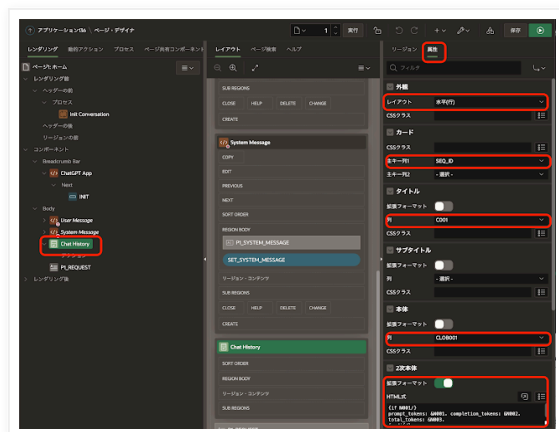


カードの表示について、**属性**で以下のように設定しています。

外観のレイアウトは**水平(行)**を選んでいます。カードの主キー列1は**SEQ\_ID**です。

タイトルの列は**C001**、本体の列は**CLOB001**、2次本体の拡張フォーマットを**ON**にし、**HTML式**として以下を記述しています。

```
{if N001/}  
prompt_tokens: &N001. completion_tokens: &N002. total_tokens: &N003.  
{endif/}
```



作成したアプリケーションの説明は以上になります。

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: 18:19

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.