

# 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4月26日 金曜日

## Oracle APEXのアプリからいろいろなローカルLLMを呼び出してみる

Apple MシリーズのDocker/Colima上にOracle APEXが動作する環境を作成し（こちらの記事）、Ollamaを使って動かしたローカルLLMを呼び出すAPEXアプリケーションも作成しました（こちらの記事）。

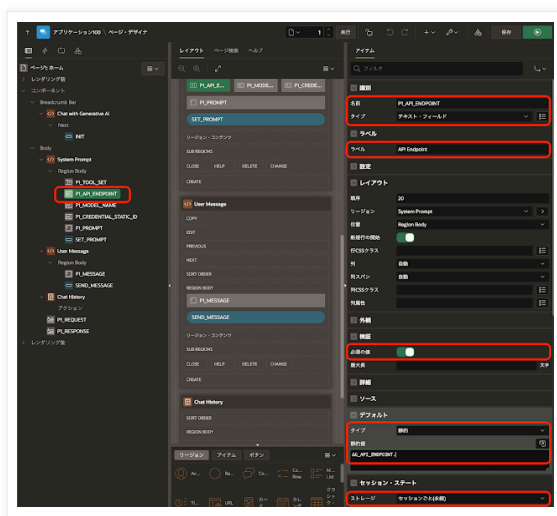
この環境を使って、最近話題のMicrosoftのphi3（Mini, 3.8B, 4bit）、Googleのgemma（7B, 4bit）、Metaのllama3（8B, 4-bit）を動かして、APEXアプリケーションからOpenAI互換のChat Completions APIで呼び出してみます。

呼び出すモデルを変更する度に、アプリケーション・ビルダーを呼び出して置換文字列を変更するのは面倒なので、APIのエンドポイント、モデル名、Web資格証明をアプリケーションから変更できるように、ページ・アイテムを追加します。

ページ・アイテムP1\_TOOL\_SETの下に、APIエンドポイントを設定するページ・アイテムP1\_API\_ENDPOINTを作成します。タイプはテキスト・フィールド、ラベルはAPI Endpointとします。

レイアウトの新規行の開始はオン、検証の必須の値はオン、デフォルトのタイプに静的を選択し、静的値として&G\_API\_ENDPOINT.を指定します。

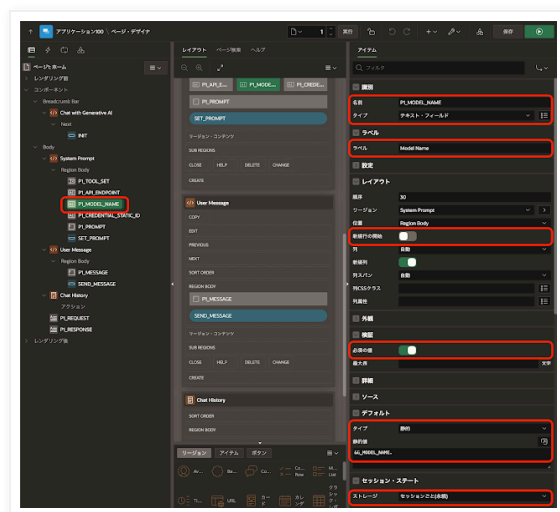
セッション・ステートのストレージはセッションごと(永続)を選択します。



モデル名を設定するページ・アイテムP1\_MODEL\_NAMEを作成します。タイプはテキスト・フィールド、ラベルはModel Nameとします。

レイアウトの新規行の開始はオフとしてページ・アイテムP1\_API\_ENDPOINTの右に配置します。検証の必須の値はオン、デフォルトのタイプに静的を選択し、静的値として&G\_MODEL\_NAME.を指定します。

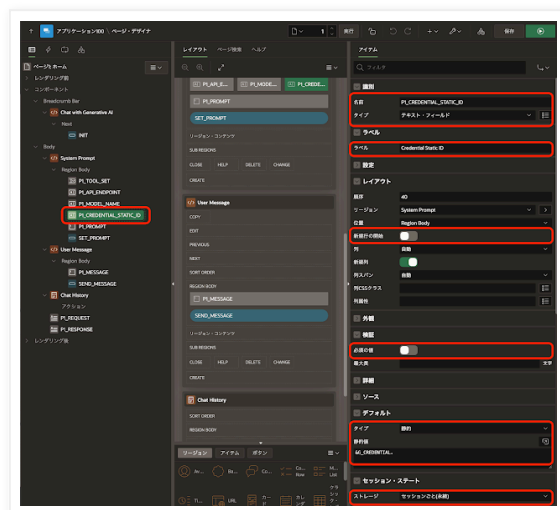
セッション・ステートのストレージはセッションごと(永続)を選択します。



Web資格証明の静的IDを設定するページ・アイテムP1\_CREDENTIAL\_STATIC\_IDを作成します。タイプはテキスト・フィールド、ラベルはCredential Static IDとします。

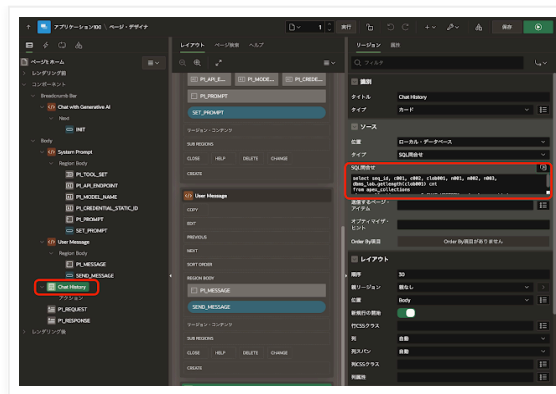
レイアウトの新規行の開始はオフとしてページ・アイテムP1\_MODEL\_NAMEの右に配置します。検証の必須の値はオフ（ローカルLLMでは指定不要）、デフォルトのタイプに静的を選択し、静的値として&G\_CREDENTIAL\_STATIC\_ID.を指定します。

セッション・ステートのストレージはセッションごと(永続)を選択します。



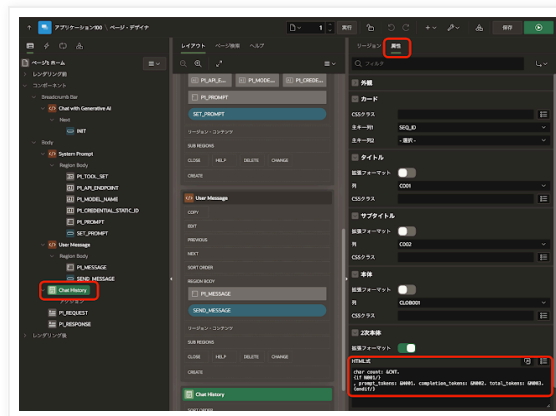
カード・リージョンChat HistoryのソースのSQL問合せを以下に変更し、送受信したメッセージの文字数を検索結果に含めます。

```
select seq_id, c001, c002, clob001, n001, n002, n003, dbms_lob.getlength(clob001) cnt
from apex_collections
where collection_name = :G_CHAT_HISTORY order by seq_id desc
```



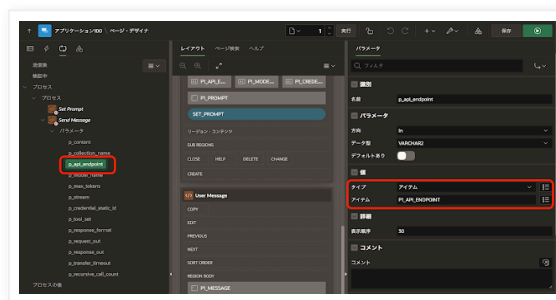
属性の2次本体のHTML式を以下に変更し、メッセージの文字数を表示します。

```
char count: &CNT.
{if N001/}
, prompt_tokens: &N001. completion_tokens: &N002. total_tokens: &N003.
{endif/}
```

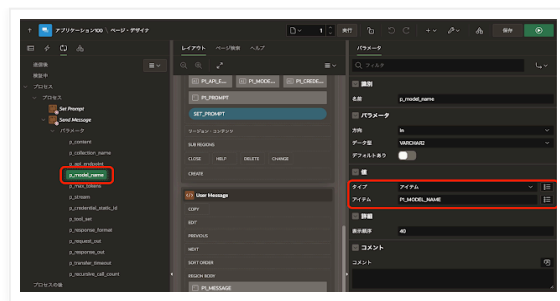


プロセスSend Messageを選択し、パラメータの設定を変更します。

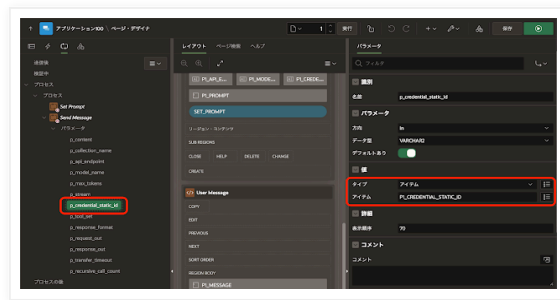
パラメータp\_api\_endpointの値のタイプをアイテムに変更し、アイテムとしてP1\_API\_ENDPOINTを指定します。



パラメータp\_model\_nameも同様に、値のアイテムとしてP1\_MODEL\_NAMEを指定します。



パラメータp\_credential\_static\_idの値のアイテムとしてP1\_CREDENTIAL\_STATIC\_IDを指定します。



以上でAPEXアプリケーションから呼び出すモデルを変更できるようになりました。

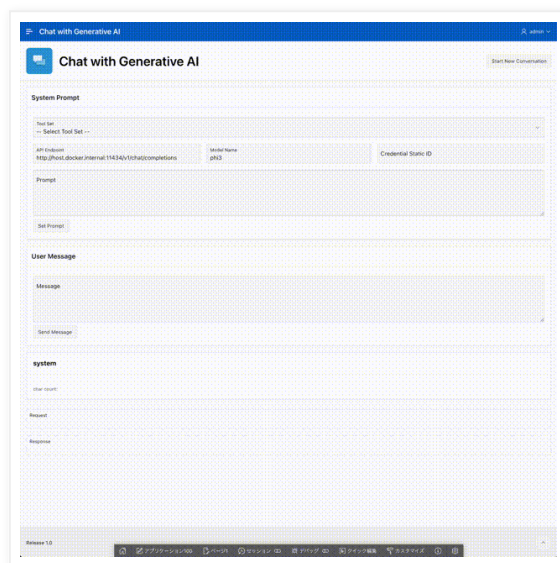
最初にMicrosoftのphi3を呼び出してみます。

```
% ollama run phi3
>>> Send a message (/? for help)
```

API Endpointはhttp://host.docker.internal:11434/v1/chat/completions、Model Nameはphi3です。

以下のメッセージを送信しました。

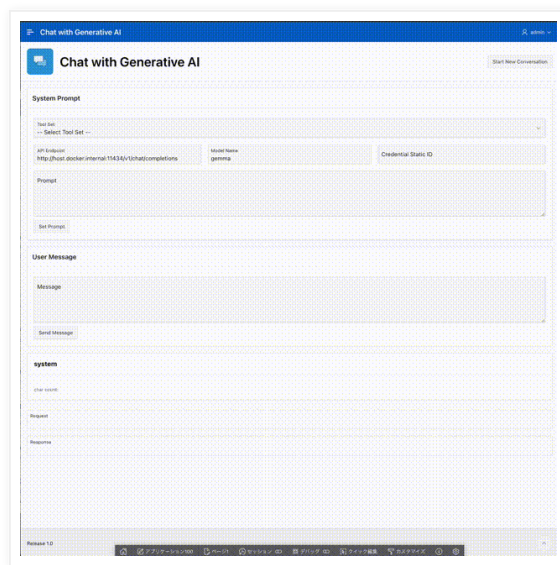
「織田信長が活躍した時代は、一般に何時代と呼ばれていますか？」



同様にGoogle gemmaを呼び出してみます。

```
% ollama run gemma
>>> Send a message (/? for help)
```

**Model Name**に**gemma**を指定します。

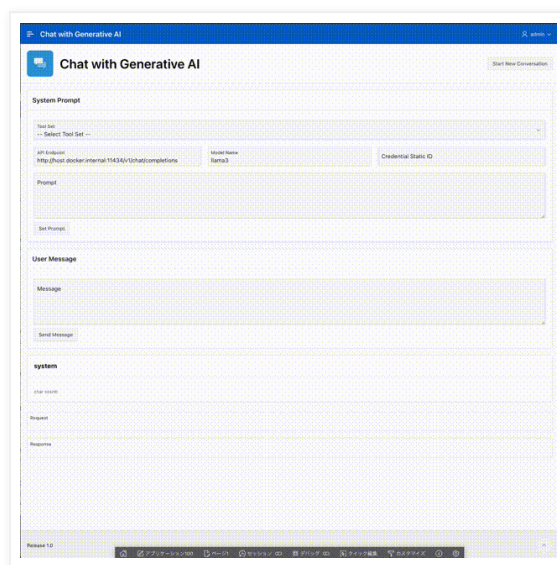


Metaのllama3を呼び出してみます。

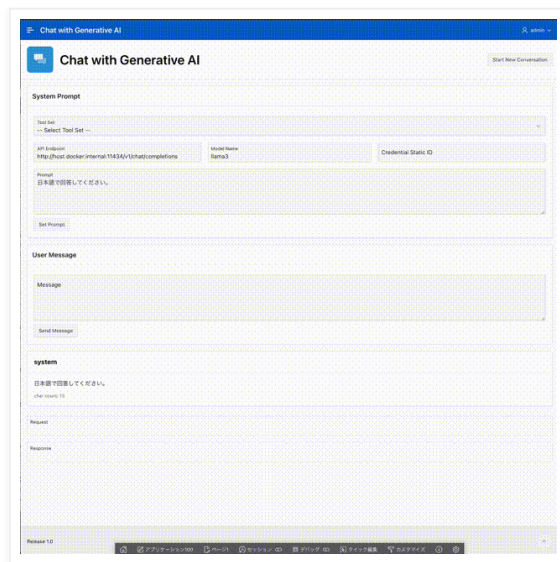
```
% ollama run llama3
>>> Send a message (/? for help)
```

**Model Name**に**llama3**を指定します。

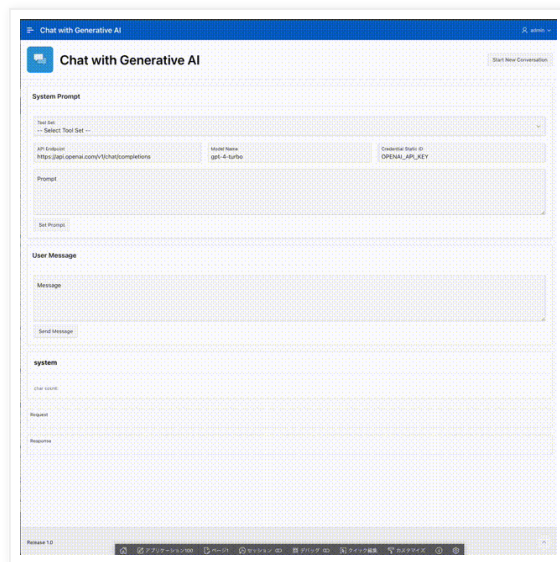
英語で回答されました。



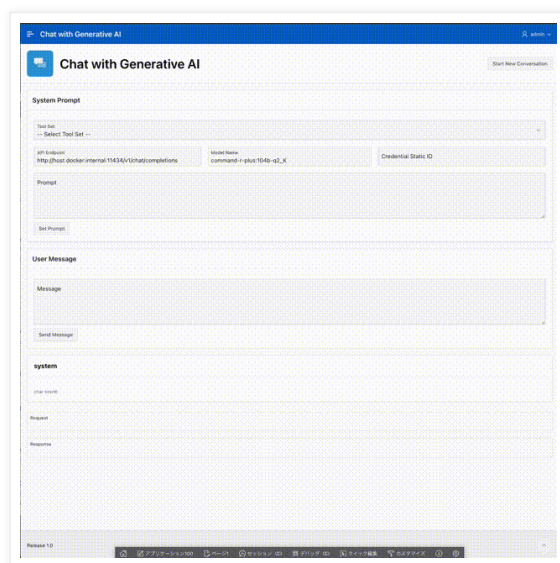
プロンプトとして「日本語で回答してください。」を設定し、同じ質問をしました。



OpenAIの**gpt-4-turbo**を呼び出してみました。

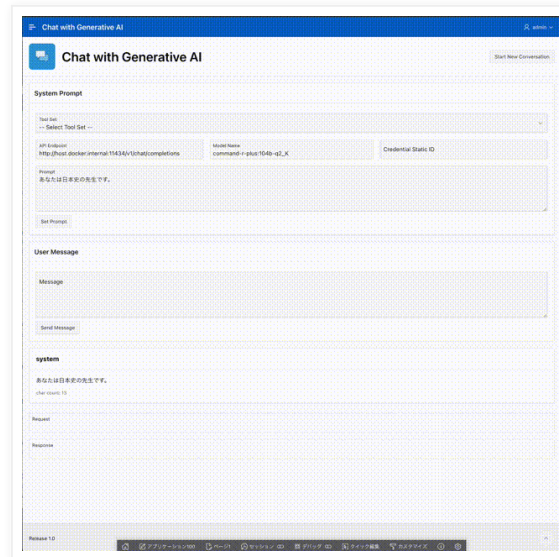


Cohereの**command-r-plus:104b-q2\_K**を呼び出してみました。回答はあっさり「戦国時代」です。



もう少し詳しく回答してもらうために、プロンプトとして「あなたは日本史の先生です。」を設定し、同じ質問をしました。





OpenAI Chat Completions APIを呼び出す形でアプリケーションを作成することにより、本家 OpenAIとOllamaのローカルLLMの切り替えが容易にできるようになりました。

今回機能を追加した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/chat-with-generative-ai-hc.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: 13:33

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.