

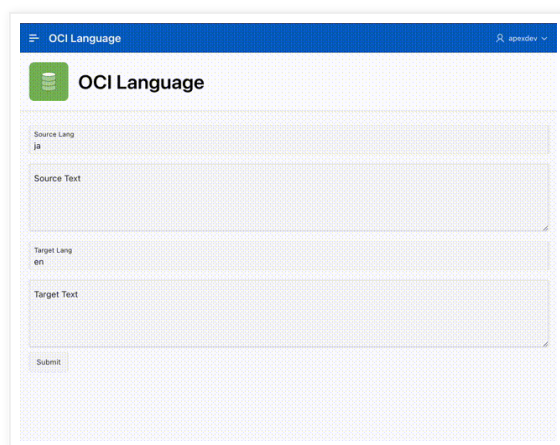
# 日々是Oracle APEX

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2月27日月曜日

## OCI Languageサービスを使用して翻訳を行う

オラクルのOCI Languageに無料枠があったので、APEXでOCI Languageの翻訳サービスを呼び出すアプリケーションを作ってみました。



OCIでのAPIユーザーの作成手順およびAPEXでのWeb資格証明の作成手順については、以前に書いた記事を参考にしています。

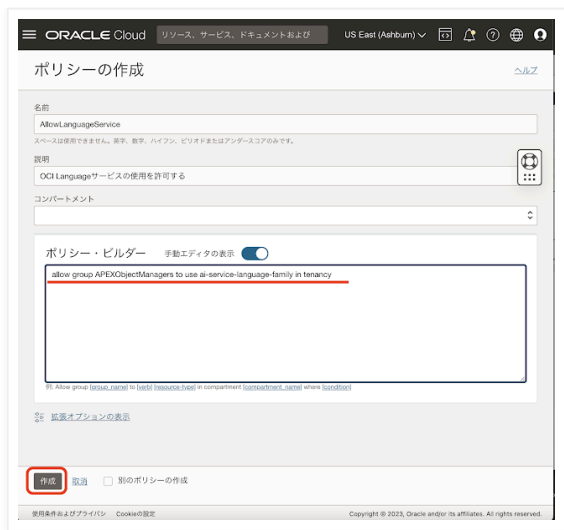
[APEXからOCIオブジェクト・ストレージを操作する\(1\) - APIユーザーの作成](#)

[APEXからOCIオブジェクト・ストレージを操作する\(4\) - Web資格証明の作成](#)

上記の手順に追加して、APIキーを持つユーザーにOCI Languageサービスの呼び出しを許可するため、ポリシーを作成します。

APIユーザーが含まれているグループがAPEXObjectManagersである場合、ポリシー・ステートメントは以下になります。

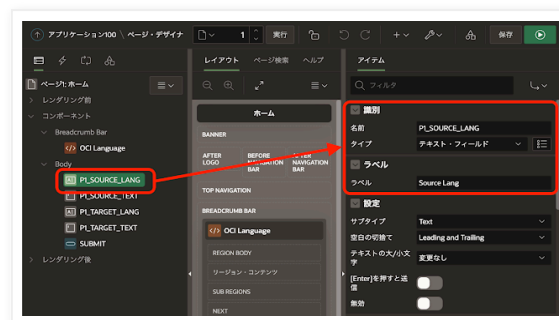
**allow group APEXObjectManagers to use ai-service-language-family in tenancy**



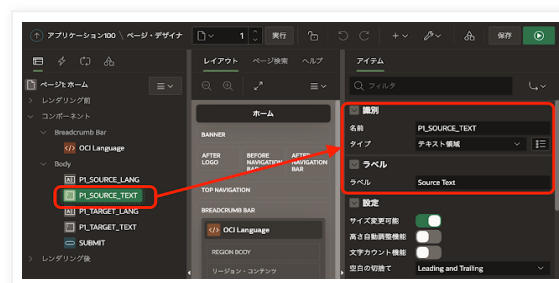
APEX側の作業として、**アプリケーション作成ウィザード**を起動し、空のアプリケーションを作成します。

作成したアプリケーションの**ホーム・ページ**を**ページ・デザイナー**で開き、**ページ・アイテム**と**ボタン**を作成します。

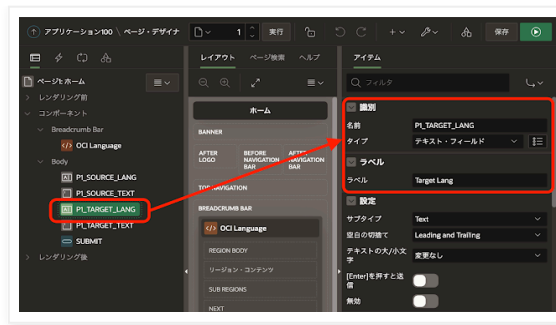
翻訳する文字列の言語を指定する**ページ・アイテムP1\_SOURCE\_LANG**を作成します。**タイプ**は**テキスト・フィールド**、**ラベル**は**Source Lang**とします。**ISO 639-1**のコードを指定します。**タイプ**を選択リストにすると、指定が容易になるでしょう。



翻訳する文字列を指定する**ページ・アイテムP1\_SOURCE\_TEXT**を作成します。**タイプ**は**テキスト領域**、**ラベル**は**Source Text**とします。



同様に**翻訳先の言語**を指定する**ページ・アイテムP1\_TARGET\_LANG**を作成します。

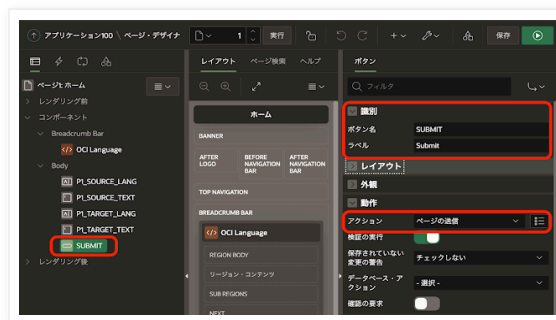


翻訳結果を保持するページ・アイテムP1\_TARGET\_TEXTを作成します。



最後に送信ボタンを作成します。

識別のボタン名はSUBMIT、ラベルはSubmit、動作のアクションはデフォルトのページの送信のままとします。



左ペインでプロセス・ビューを開き、ボタンSUBMITを押した時に実行するプロセスTranslateを作成します。

ソースのPL/SQLコードとして以下を記述します。

```
declare
    /* OCI Languageの翻訳処理のエンドポイント */
    C_ENDPOINT constant varchar2(120) := 'https://language.aiservice.us-ashburn-1.oci.oraclecloud
    l_request json_object_t;
    l_request_clob clob;
    l_response json_object_t;
    l_response_clob clob;
    l_documents json_array_t;
    l_document json_object_t;

begin
    /* 翻訳を依頼するメッセージを作成する。 */
```

```

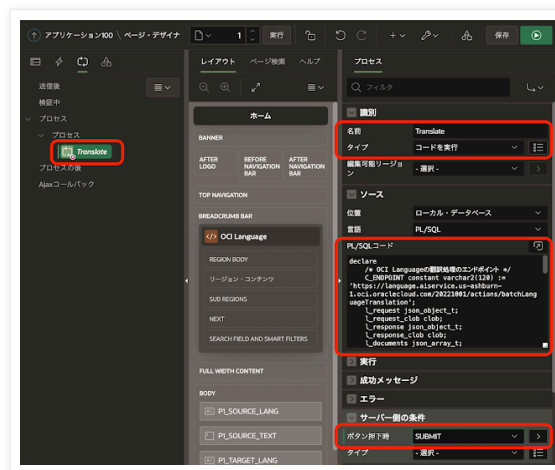
l_request := json_object_t();
l_document := json_object_t();
l_document.put('key', '1');
l_document.put('text', :P1_SOURCE_TEXT);
l_document.put('languageCode', :P1_SOURCE_LANG);
l_documents := json_array_t();
l_documents.append(l_document);
l_request.put('documents', l_documents);
l_request.put('targetLanguageCode', :P1_TARGET_LANG);
l_request_clob := l_request.to_clob();
/* 翻訳リクエストの発行 */
apex_web_service.clear_request_headers();
apex_web_service.set_request_headers('Content-Type','application/json');
l_response_clob := apex_web_service.make_rest_request(
    p_url => C_ENDPOINT
    ,p_http_method => 'POST'
    ,p_body => l_request_clob
    ,p_credential_static_id => 'OCI_API_ACCESS'
);
/* 応答のすべてをJSON形式で確認する。 */
-- :P1_TARGET_TEXT := l_response_clob;
/* 翻訳された文章だけを確認する。 */
l_response := json_object_t(l_response_clob);
l_documents := l_response.get_array('documents');
/* 送信したkeyは1だけなので、翻訳された文章も1つだけ */
l_document := json_object_t(l_documents.get(0));
:P1_TARGET_TEXT := l_document.get_string('translatedText');
end;

```

oci-language-translate.sql hosted with ❤️ by GitHub

[view raw](#)

サーバー側の条件のボタン押下時にSUBMITを選択します。



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。

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/oci-language-translate.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: 14:52

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.

---