2024年6月24日月曜日

Oracle APEXでTurbo Framesを使用する

先日、Oracle APEXでhtmxを使用するサンプル・アプリケーションを作成しました。同じアプリケーションをHotwireのTurbo Framesで置き換えてみます。

htmxを使用するアプリケーションの作り方は、こちらの記事で紹介しています。本記事ではTurbo Framesを使用するための変更点を紹介します。

今回作成したアプリケーションは以下のように動作します。見かけはhtmxを使ったものと、ほぼ同じです。

以下にアプリケーションの変更点を紹介します。

アプリケーションを実装しているホーム・ページのページ・プロパティの変更から始めます。

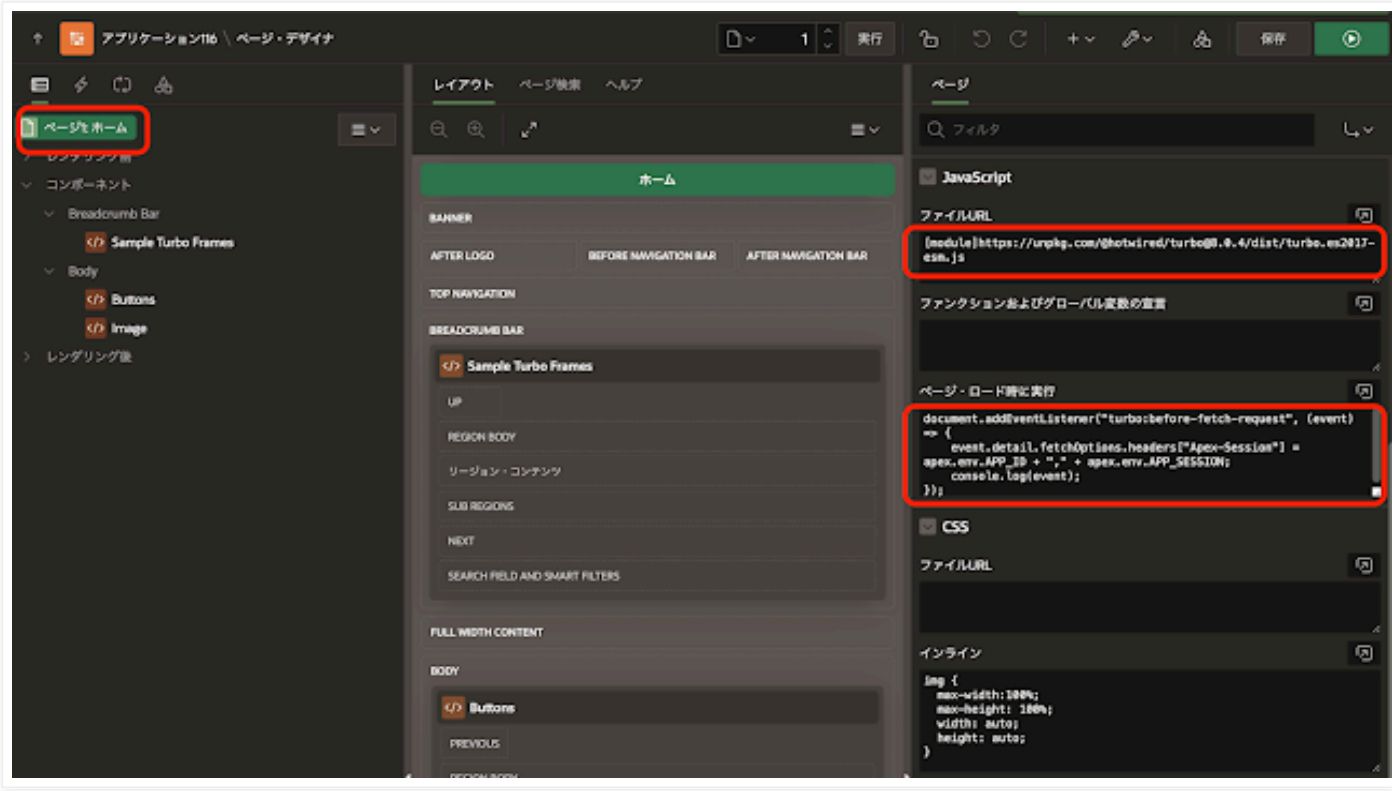
JavaScriptのファイルURLの指定は以下になります。TurboのESモジュールをロードするため、<script type="module"となるように**[module]**を先頭に付与します。

[module]https://unpkg.com/@hotwired/turbo@8.0.4/dist/turbo.es2017-esm.js

本番環境などでは、このファイルをアプリケーションまたはワークスペースに静的ファイルとして保存し、ミニファイされたファイルをロードする方が良いでしょう。

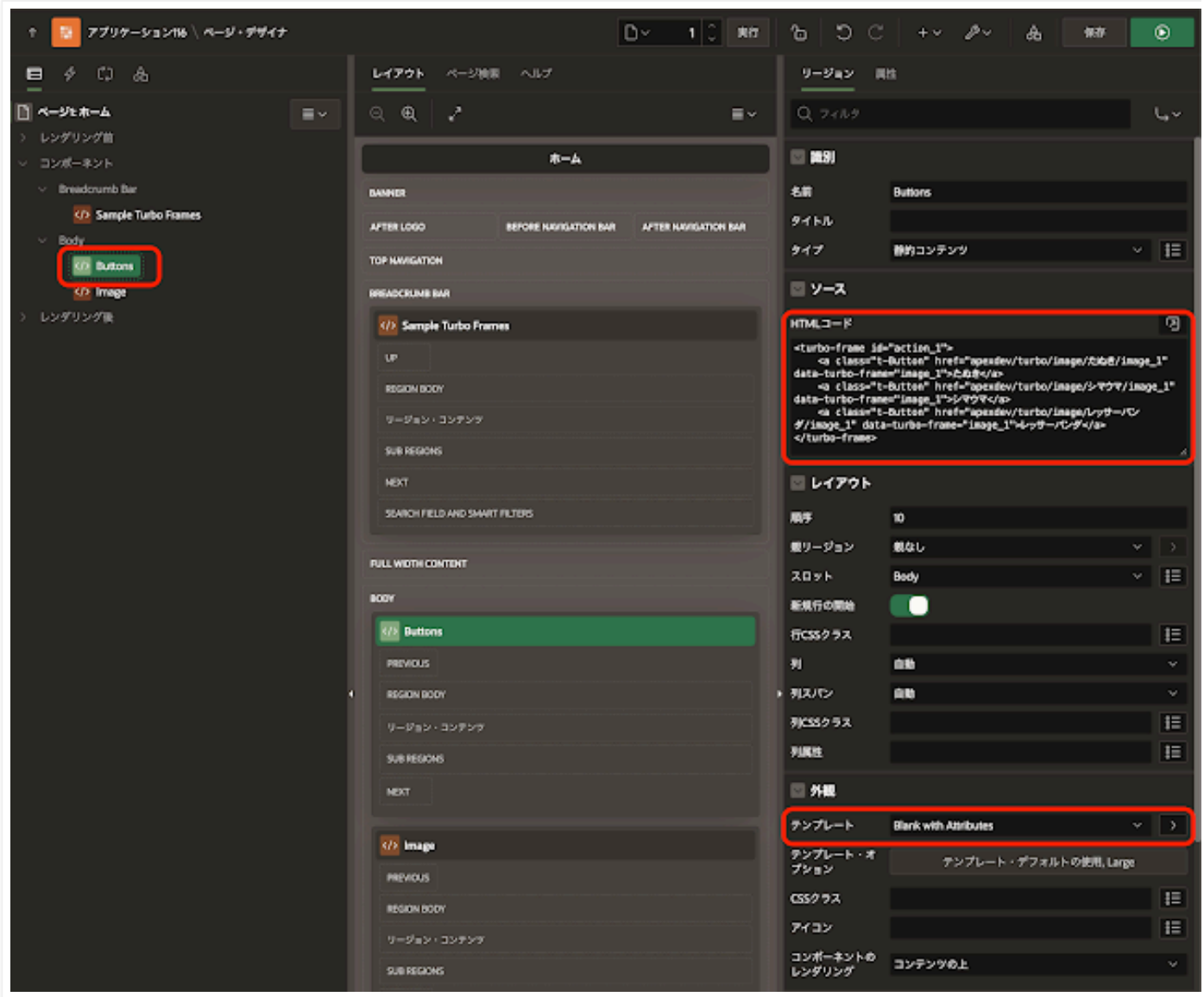
ページ・ロード時に実行に以下を記述します。画像を取得するRESTサービスのリクエストに、**Apex-Session**ヘッダーを含めます。Turboの場合は**turbo:before-fetch-request**のイベントを受けて、ヘッダーの追加を行います。

```
document.addEventListener("turbo:before-fetch-request", (event) => {
   event.detail.fetchOptions.headers["Apex-Session"] = apex.env.APP_ID + "," + apex.env.APP_SESSION;
   // console.log(event);
});
```



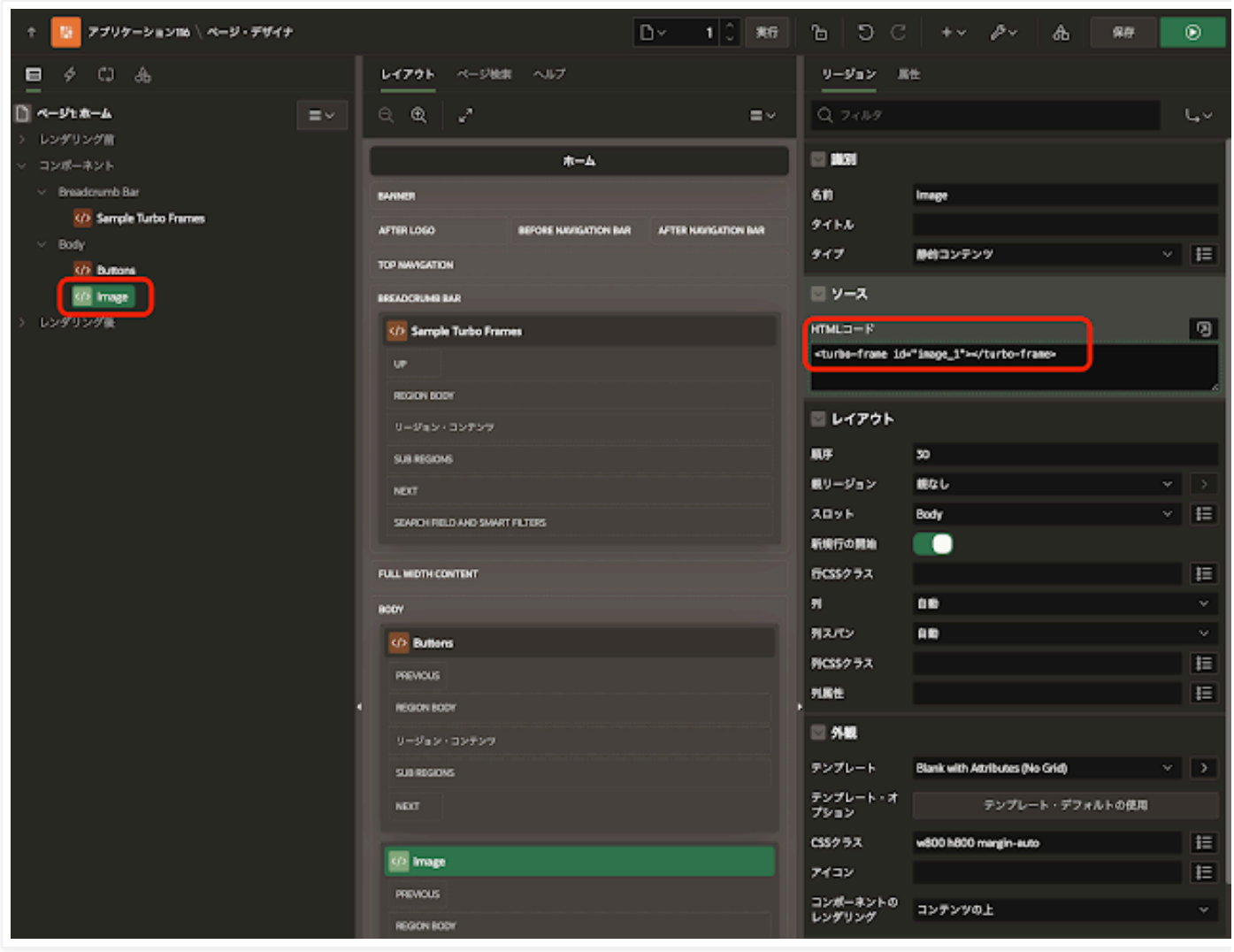
ボタンはすべて、**turbo-frame**要素の中のA要素として作成します。**data-turbo-frame**として**image_1**を指定することにより、**id**が**image_1**の**turbo-frame**要素を、**href**の呼び出しで受け取ったHTML（今回は画像）で更新します。

```
<turbo-frame id="action_1">
   <a class="t-Button" href="apexdev/turbo/image/たぬき/image_1" data-turbo-frame="image_1">たぬき</a>
   <a class="t-Button" href="apexdev/turbo/image/シマウマ/image_1" data-turbo-frame="image_1">シマウマ</a>
   <a class="t-Button" href="apexdev/turbo/image/レッサーパンダ/image_1" data-turbo-frame="image_1">レッサーパンダ</a>
</turbo-frame>
```



画像を描画する領域の**HTMLソース**として、**turbo-frame**要素を記述します。**id**は**image_1**です。

```
<turbo-frame id="image_1"></turbo-frame>
```



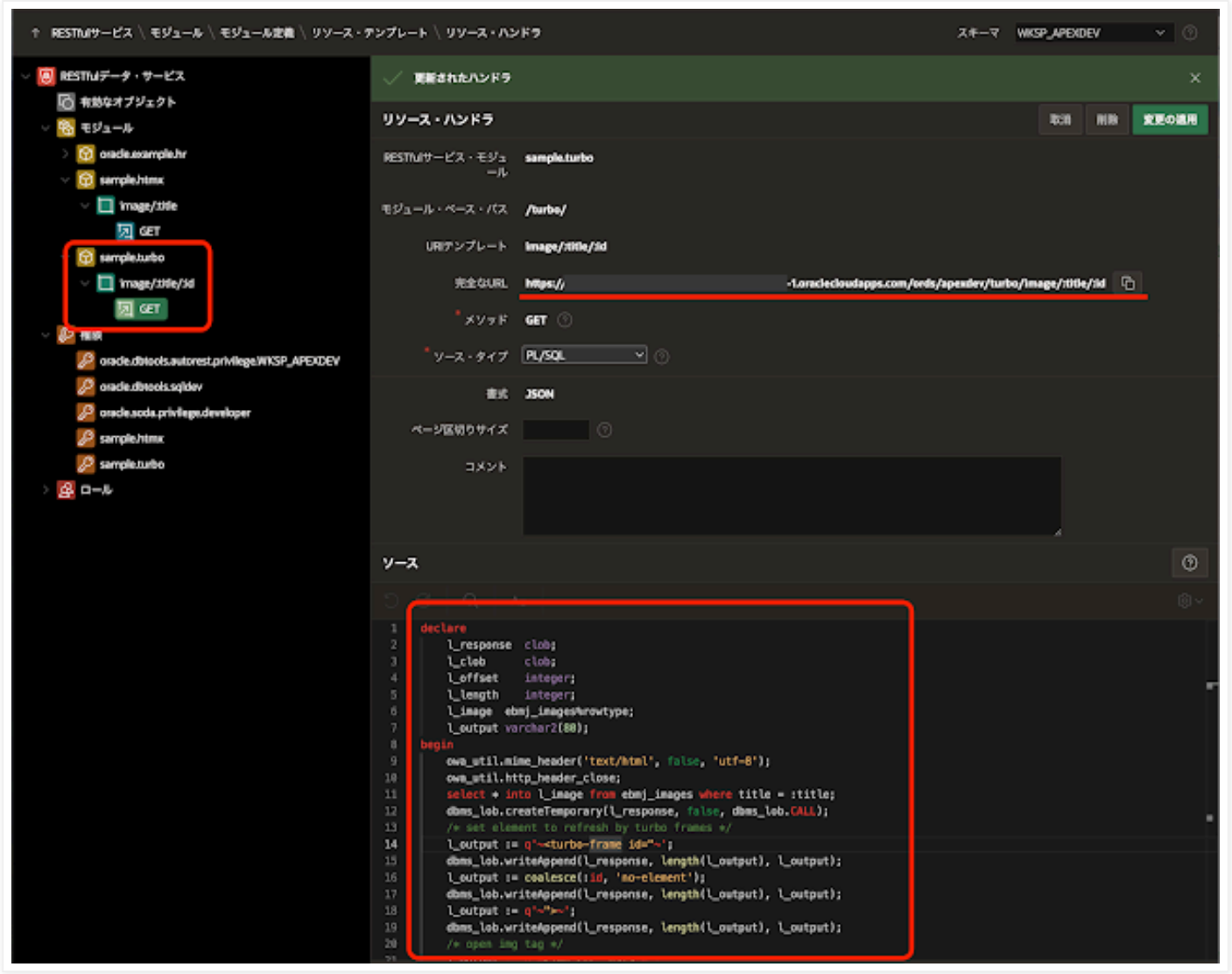
画像を返すRESTサービスでは、更新対象を示すturbo-frame要素にIMG要素を含めて、HTMLを返すようにコードを変更します。

モジュール・パスは**/turbo/**、**テンプレート**は**image/:title/:id**とし、更新対象とするturbo-frame要素のidも引数に含めました。

```
declare
    l_response  clob;
    l_clob      clob;
    l_offset    integer;
    l_length    integer;
    l_image  ebmj_images%rowtype;
    l_output varchar2(80);
begin
    owa_util.mime_header('text/html', false, 'utf-8');
    owa_util.http_header_close;
    select * into l_image from ebmj_images where title = :title;
    dbms_lob.createTemporary(l_response, false, dbms_lob.CALL);
    /* set element to refresh by turbo frames */
    l_output := q'~<turbo-frame id="~';
    dbms_lob.writeAppend(l_response, length(l_output), l_output);
    l_output := coalesce(:id, 'no-element');
    dbms_lob.writeAppend(l_response, length(l_output), l_output);
    l_output := q'~">~';
```

```
    dbms_lob.writeAppend(l_response, length(l_output), l_output);
    /* open img tag */
    l_output := q'~<img src="data:~';
    dbms_lob.writeAppend(l_response, length(l_output), l_output);
    l_output := l_image.content_mimetype;
    dbms_lob.writeAppend(l_response, length(l_output), l_output);
    l_output := q'~;base64,~';
    dbms_lob.writeAppend(l_response, length(l_output), l_output);
    /* base64 encoded image */
    l_clob := apex_web_service.blob2clobbase64(l_image.content, 'N');
    l_length := dbms_lob.getlength(l_clob);
    l_offset := dbms_lob.getlength(l_response) + 1;
    dbms_lob.copy(
        dest_lob => l_response
        ,src_lob => l_clob
        ,amount =>  l_length
        ,dest_offset => l_offset
        ,src_offset => 1
    );
    /* close img tag */
    l_output := q'~">~';
    dbms_lob.writeAppend(l_response, length(l_output), l_output);
    l_output := q'~</turbo-frame>~';
    dbms_lob.writeAppend(l_response, length(l_output), l_output);
    /* return img tag */
    apex_util.prn(l_response, false);
    dbms_lob.freeTemporary(l_response);
exception
    when no_data_found then
        :status_code := 204;
        htp.p('<div>no data found</div>');
end;
```

rest-image-get-turbo.sql hosted with ❤️ by GitHub view raw



Turbo Framesを使うために実施した変更は以上になります。

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/sample-turbo-frames.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: 14:05

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.