

# 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。

2022年3月24日 木曜日

## メンテナンスなどの通知をアプリケーションに表示する

APEXアプリケーションの画面にメンテナンスの告知などの通知を表示させるには、という質問を見かけたので、試しにいくつか実装してみました。

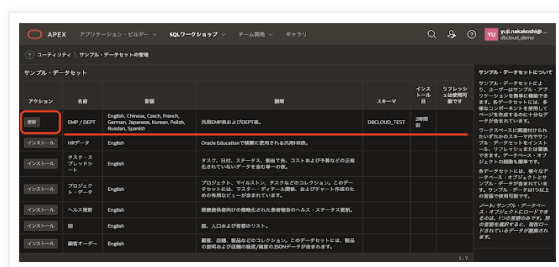
APEX標準の機能としては、**グローバル通知**があります。それ以外にも、**グローバル・ページ**にリージョンを作成することにより、通知を実装できます。

以下の4つを試してみました。

1. アプリケーション定義のグローバル通知を使う。
2. グローバル・ページに静的コンテンツのリージョンを作成する - 表示はブラウザ側のJavaScriptで制御する。
3. グローバル・ページに静的コンテンツのリージョンを作成する - 表示はAPEXのユーザー・プリファレンスを使って、サーバー側で制御する。
4. グローバル・ページにインライン・ダイアログのリージョンを作成する。

通知の実装には、**サンプル・データセット**の**EMP/DEPT**をインストールすると作成できるアプリケーション**デモ - 従業員 / 部門**を使用します。アプリケーションの内容に依存する作業はないため、どのようなAPEXアプリケーションを元にしても、通知の実装を試すことができます。

**SQLワークショップ**のユーティリティの**サンプル・データセット**から、**EMP/DEPT**のデータセットを**インストール**します。すでにインストール済みの場合は、アプリケーションを作成するために**更新**を実行します。



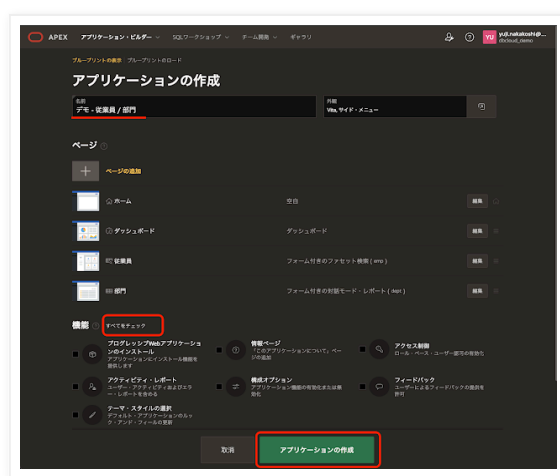
画面の案内に従って**データセットのインストール**または**既存のデータセットのリフレッシュ**を実行すると、**アプリケーションの作成**を呼び出す画面に移ります。

**アプリケーションの作成**をクリックします。



アプリケーション作成ウィザードが起動したら、機能は使わないので、**すべてをチェック**をクリックし、**チェックを全部外して**アプリケーションに組み込まれないようにします。

アプリケーションの作成をクリックすると、今回の実装の元になるアプリケーションが作成されます。



## アプリケーション定義のグローバル通知を使う

以下のように通知が表示される実装を行います。



アプリケーション定義のグローバル通知として、以下を記述します。

```
<div class="global-notification-area" align="center">
```

```
<pre class="banner-content">
```

```
2022年4月17日午前9:00より10:00まで、メンテナンスのため本アプリケーションは利用できません。
```

```
詳しくは<a href="http://...../notice.html">http://...../notice.html</a>を参照してください。
```

```
</pre>
```

```
</div>
```

変更の適用をクリックします。



ページを実行すると、以下の表示になります。

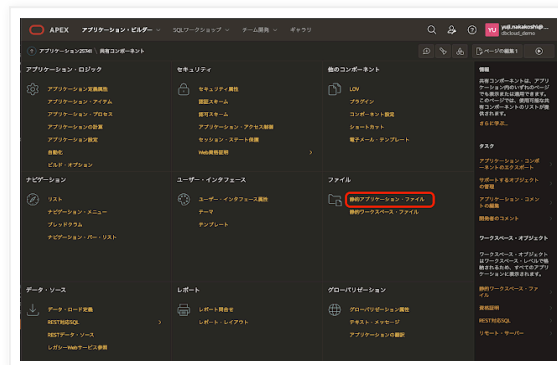


グローバル通知の説明にあるように、ここで記述している文字列はページ・テンプレートに含まれる置換文字列#GLOBAL\_NOTIFICATION#を置き換えます。それだけの処理なので、このままだと見栄えが今ひとつです。

通知文にはCSSのクラス**banner-content**を指定しています。このクラスによって、通知の表示スタイルを調整します。

CSSの定義はアプリケーションに含まれるすべてのページから参照できるよう、静的アプリケーション・ファイルに記述します。複数のアプリケーションから参照させる場合は、静的ワークスペース・ファイルとして作成します。

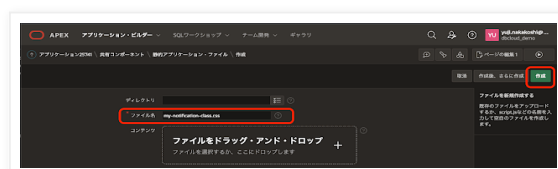
共有コンポーネントの静的アプリケーション・ファイルを開きます。



ファイルの作成を実行します。



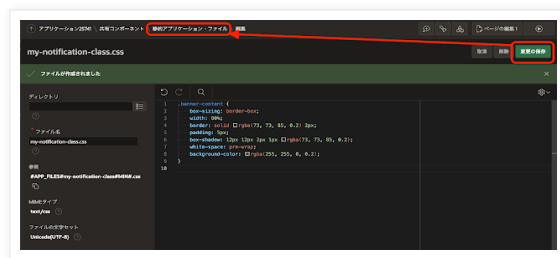
ファイル名としてmy-notification-class.cssを入力し、作成をクリックします。



ファイルの内容として以下を記述します。

```
.banner-content {
  box-sizing: border-box;
  width: 80%;
  border: solid rgba(73, 73, 85, 0.2) 2px;
  padding: 5px;
  box-shadow: 12px 12px 2px 1px rgba(73, 73, 85, 0.2);
  white-space: pre-wrap;
  background-color: rgba(255, 255, 0, 0.2);
}
```

**変更の保存**をクリックします。変更が保存されたら、**静的アプリケーション・ファイル**の一覧へ戻ります。

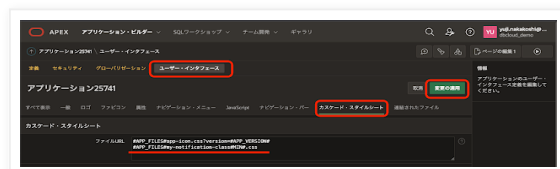


CSSファイルであれば、自動的にMinifyされます。MinifyされたCSSファイルの、**参照**として表示されているパス**#APP\_FILES#my-notification-class#MIN#.css**をクリップボードにコピーします。



**アプリケーション定義のユーザー・インターフェース**を開きます。**カスケード・スタイルシート**のセクションの**ファイルURL**として、**静的アプリケーション・ファイル**として作成したCSSファイルの**参照のパス**を貼り付けます。

**変更の適用**をクリックします。



以上で、最初の画像のように通知が表示されます。

グローバル通知はアプリケーションのすべての画面で表示されます。以下のようにログイン画面も含まれます。



イベントのページのロート上でコンテキスト・メニューを表示させ、動的アクションの作成を実行します。



```
apex.env.APP_PAGE_ID == 9999
```

The screenshot shows the AEM console interface. On the left, the breadcrumb navigation is 'ページ > ロード時に追加されるアクション'. The 'myPageLoadGlobal' action is highlighted in the list. The main panel displays the configuration for this action, including the '名前' (Name) field set to 'myPageLoadGlobal', the 'タイプ' (Type) set to 'JavaScript', and the 'スクリプト' (Script) field containing the path 'libs/overlay/fragments/js/myPageLoadGlobal.js'. The 'イベント' (Event) field is set to 'ページロード' (Page Load).

The screenshot shows the AWS IAM console interface. On the left, the 'Groups' list table contains one entry: 'システム管理者' (System Administrator) with a status of 'Active'. The 'Details' tab on the right shows the group's configuration, including its name 'システム管理者' and its role 'PowerUserAccess'.

動的アクションには**非表示**もありますが、これはスタイルとしては**visibility: hidden**に対応しています。今回の用途では**display: none**の方が適切ななので、**動的アクション**としては、**スタイルの設定**を選んでいきます。

通知メッセージの一番外側の

以上で、サインイン画面を除いたすべてのページで、メンテナンスに関する通知が表示されるようになりました。

## グローバル・ページに静的コンテンツのリージョンを作成する - 表示はブラウザ側のJavaScriptで制御する

ページの先頭に通知を表示します。また、ボタンをクリックして通知を削除できるようにします。通知の削除は、JavaScriptによりブラウザ側で制御します。

通知はAPEX 21.2から導入されたBannerの位置に表示しています。そのため、APEX 21.2以降が実装の要件になります。



ページ・デザイナーでグローバル・ページを開きます。

コンポーネントまたはBody上でコンテキスト・メニューを開き、リージョンの作成を実行します。

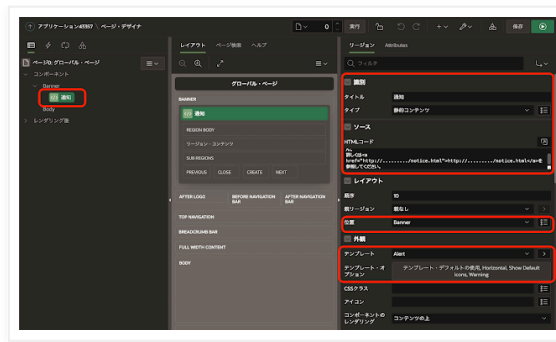


リージョンが作成されます。

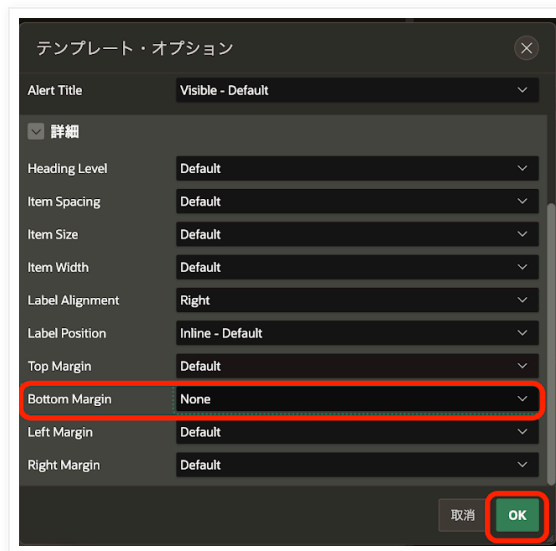
識別の名前を通知、タイプとして静的コンテンツを選択します。ソースのHTMLコードには、通知したい以下の文章を記述します。

2022年4月17日午前9:00より10:00まで、メンテナンスのため本アプリケーションは利用できません。  
詳しくは<http://...../notice.html>を参照してください。

レイアウトの位置にBanner、外観のテンプレートとしてAlertを選択します。

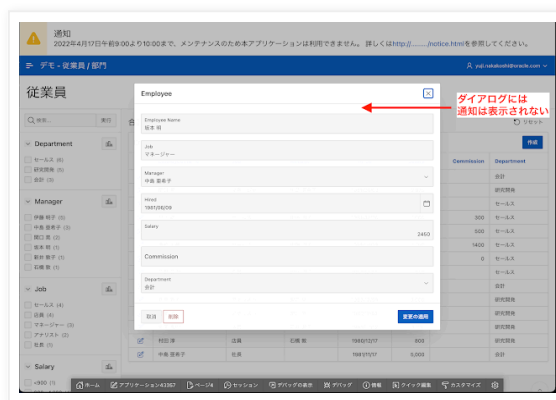


テンプレート・オプションを開き、**Bottom Margin**をNoneに変更します。



以上で、通知がBannerの位置に表示されるようになりました。

グローバル通知とは異なり、Bannerが位置として定義されているページ・テンプレートでのみ、リージョン通知が表示されます。ログイン・ページやダイアログにはBannerの位置が定義されていないため、通知は表示されません。

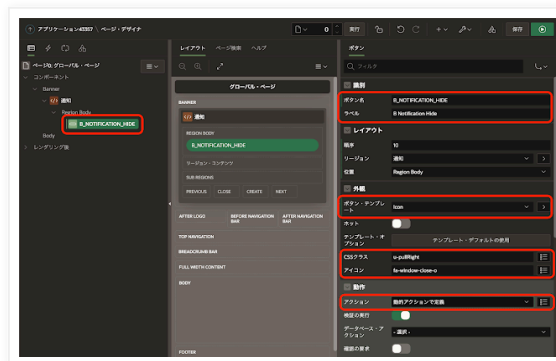


通知の削除を実装します。

リージョン通知にボタンを作成します。

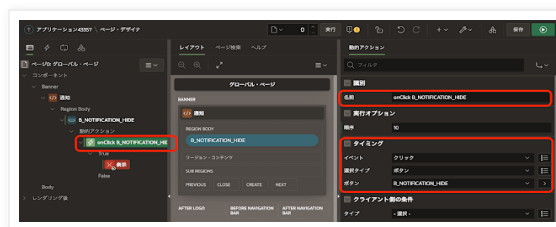
識別のボタン名はB\_NOTIFICATION\_HIDEとします。外観のボタン・テンプレートとしてIconを選択します（そのため、ラベルは表示されないのので特に変更しません）。右端に表示されるようCSSクラスにu-pullRight、アイコンとしてfa-window-close-oを選択します。

動作のアクションとして、動的アクションで定義を選択します。



ボタンに動的アクションを作成します。

識別の名前をonClick B\_NOTIFICATION\_HIDEとします。タイミングはイベントはクリック、選択タイプはボタン、ボタンはB\_NOTIFICATION\_HIDEになります。

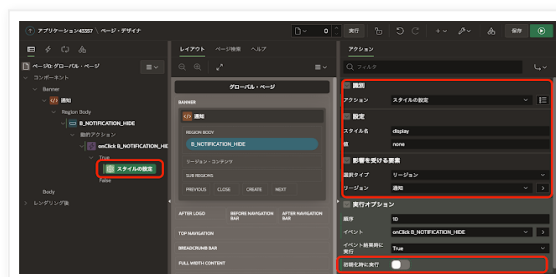


ここまでの実装は、ブラウザ側でもサーバー側でも一緒です（サーバー側での実装を説明するときは、この作業以降から始めます）。

ここから、ブラウザ側のJavaScriptによる実装になります。

作成済みのTRUEアクションを選択します。

識別のアクションとしてスタイルの設定を選択します。設定のスタイル名はdisplay、値はnoneとします。影響を受ける要素の選択タイプとしてリージョンを選び、リージョンは通知とします。実行オプションの初期化時に実行はOFFにします。



追加でTRUEアクションを作成します。

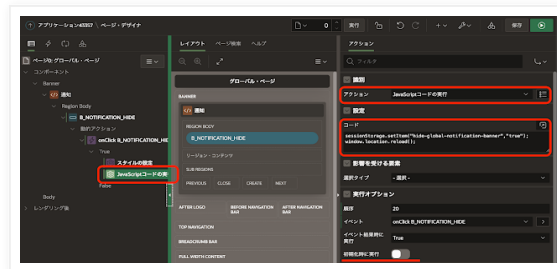
識別のアクションとしてJavaScriptコードの実行を選択します。設定のコードには以下を記述します。

```
sessionStorage.setItem("hide-global-notification-banner", "true");  
window.location.reload();
```



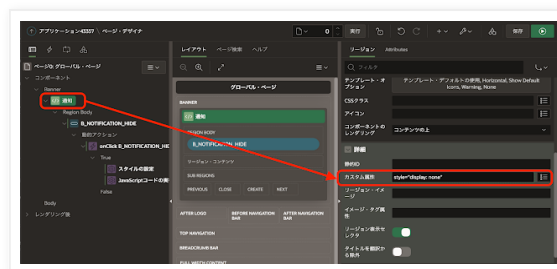
ブラウザのsessionStorageにhide-global-notification-bannerとしてtrueを保存した後、ページ全体をリロードしています。

もっと良いやり方があるはずですが、JavaScriptとCSSが得意ではないのでこれ以上は分かりませんでした。多分、**スタイルの設定**ではなく、もっと凝ったCSSクラスを静的アプリケーション・ファイルに定義し、動的アクションの**アクション**としては**クラスの追加やクラスの削除**を使うことになると思います。



リージョン通知が初期状態で非表示になるよう、詳細の**カスタム属性**に以下を記述します。

`style="display: none"`



ページのロード時にブラウザのsessionStorageのhide-global-notification-bannerを評価して、trueでなければリージョン通知を表示するように、動的アクションを作成します。

右ペインにて**動的アクション・ビュー**を開き、ページのロードで動的アクションの作成を実行します。

作成された動的アクションの識別の名前をonPageLoad Globalとします。クライアント側の条件として以下を記述します。

`sessionStorage.getItem("hide-global-notification-banner") != "true"`



作成済みのTRUEアクションを選択します。

識別の**アクション**として**スタイルの設定**を選択します。設定のスタイル名はdisplay、値はcontentsとします。影響を受ける要素の選択タイプはリージョン、リージョンとして通知を選択します。



今回は、リージョン通知の**カスタム属性**として**style="display: none"**を設定し、**ページのロード時**に（通知を表示する場合は）**display: contents**を設定しています。この設定が逆（HTMLが生成されるときはdisplayの指定が無く、ページのロード時にdisplay: noneを設定して非表示にする）になっていると、ページが表示されるときに一瞬ですがリージョン通知が表示されてしまいます。

以上で通知の実装は完了です。

sessionStorageに通知の表示/非表示の選択を保存しているため、一旦通知を閉じるとブラウザを終了するまで通知が再表示されることはありません。

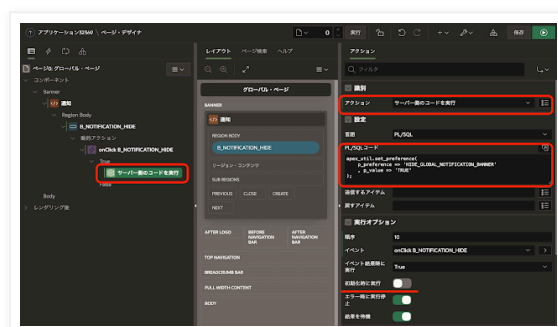
## グローバル・ページに静的コンテンツのリージョンを作成する - 表示はAPEXのユーザー・プリファレンスを使って、サーバー側で制御する

通知を削除するボタンに作成した、TRUEアクションの設定から始めます。

識別の**アクション**として、**サーバー側のコードを実行**を選択します。設定のPL/SQLコードとして以下を記述します。

```
apex_util.set_preference(
  p_preference => 'HIDE_GLOBAL_NOTIFICATION_BANNER'
  , p_value => 'TRUE'
);
```

実行オプションの初期化時に実行はOFFです。

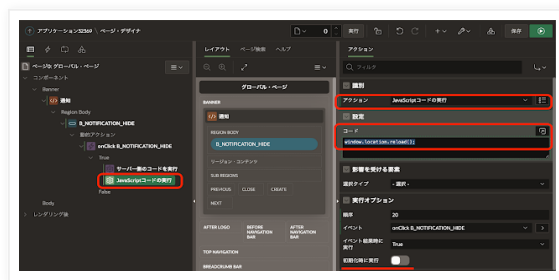


追加でTRUEアクションを作成します。

識別の**アクション**として**JavaScriptコードの実行**を選択します。設定のコードには以下を記述します。

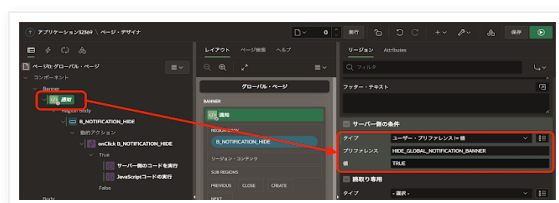
```
window.location.reload();
```

今回はリージョン通知は**サーバー側の条件**で非表示にします。そのため、ページの再読み込みを行う必要があります。



リージョン通知を選択し、**サーバー側の条件**を設定します。

**サーバー側の条件**の**タイプ**として、**ユーザー・プリファレンス != 値**を選択します。**プリファレンス**は**HIDE\_GLOBAL\_NOTIFICATION\_BANNER**とし、**値**には**TRUE**を指定します。



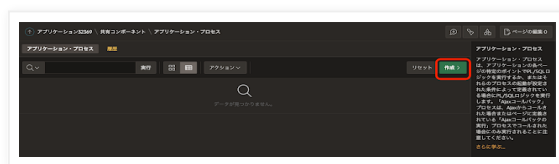
ユーザー・プリファレンスは一旦設定するとずっと維持されます。つまり永遠に通知が表示されなくなります。それだと困るため、サインアウトすると設定をリセットするようにします。

**共有コンポーネントのアプリケーション・プロセス**を開きます。



作成済みの**アプリケーション・プロセス**の一覧が表示されます。

**作成**をクリックします。



名前は**RESET\_HIDE\_GLOBAL\_NOTIFICATION\_BANNER**とします。**ポイント**は**認証後**です。**ユーザー・プリファレンス**は**ユーザー**に対する設定なので、一般ユーザーが操作するためには、**ユーザー認証**が完了している必要があります。

次へ進みます。

アプリケーション・プロセスの作成

名前 RESET\_HIDE\_GLOBL\_NOTIFICATION\_

順序 1

ポイント 画面遷換

「Ajaxコールバック」プロセスは、可能なかぎりページ・レベルで宣言してください。

次 >

コードとして以下を記述します。ユーザー・プリファレンス  
**HIDE\_GLOBAL\_NOTIFICATION\_BANNER**を削除します。

```
apex_util.remove_preference(  
    p_preference => 'HIDE_GLOBAL_NOTIFICATION_BANNER'  
);
```

次へ進みます。

アプリケーション・プロセスの作成

言語 PL/SQL

検証成功

```
1 apex_util.remove_preference(  
2     p_preference => 'HIDE_GLOBAL_NOTIFICATION_BANNER'  
3 );
```

コードを検証しない(実行時にのみコードを解析)。

次 >

条件タイプの設定は不要です。**プロセスの作成**を実行します。

アプリケーション・プロセスの作成

条件タイプ

条件タイプの選択

式 アイテム/関数 アイテム/関数はNULLではない アイテム/関数はNULL 存在

プロセスの作成

通知の非表示設定をリセットするアプリケーション・プロセスが作成されました。



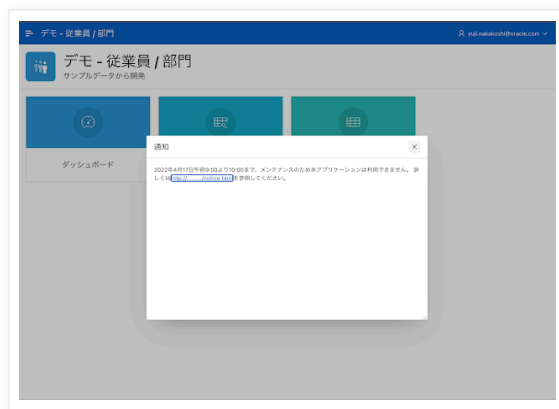
以上で通知の実装は完了です。

おおむねJavaScriptでの実装と同じ動作をしますが、サインインし直すと通知が再度表示されるところが異なります。

この例ではユーザー・プリファレンスを使用して通知の表示を制御していますが、**サーバー側の条件**であれば、他のデータベースに保存されている情報を元にして、通知の表示の有無を決めることができます。

## グローバル・ページにインライン・ダイアログのリージョンを作成する

サインイン直後に通知として、以下のようにダイアログを開きます。



データセットから作成したアプリケーションを元に実装を始めます。

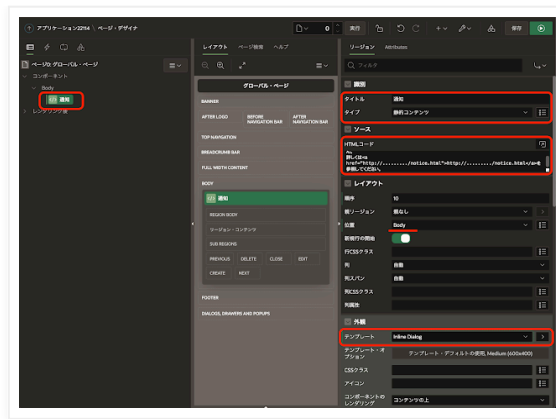
**ページ・デザイナー**にて**グローバル・ページ**を開き、リージョンを作成します。

作成されたリージョンの識別の**タイトル**は**通知**、**タイプ**は**静的コンテンツ**とします。**ソース**の**HTMLコード**に以下を記述します。

2022年4月17日午前9:00より10:00まで、メンテナンスのため本アプリケーションは利用できません。  
詳しくは<http://...../notice.html>を参照してください。

**レイアウト**の**位置**は**Body**のまま、変更しません。

**外観**の**テンプレート**として**Inline Dialog**を選択します。



通知となるインライン・ダイアログが、サインイン後に開かれるように動的アクションを作成します。

左ペインで動的アクション・ビューを開き、ページのロードで動的アクションの作成を実行します。

作成された動的アクションの、識別の名前はonPageLoad Globalとします。



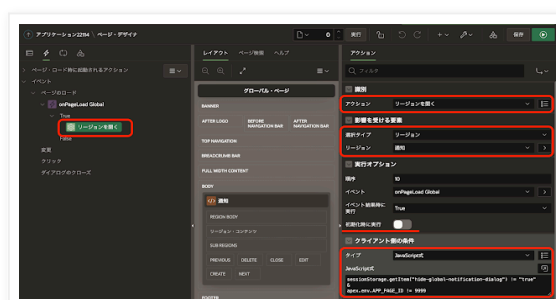
作成済みのTRUEアクションを選択します。

識別のアクションとしてリージョンを開くを選択します。影響を受ける要素の選択タイプにリージョンを選び、リージョンとして通知を指定します。実行オプションの初期化時に実行はOFFです。クライアント側の条件のタイプとして、JavaScript式を選択し、JavaScript式に以下を記述します。

```
sessionStorage.getItem("hide-global-notification-dialog") != "true"
&
apex.env.APP_PAGE_ID != 9999
```

この後に、sessionStorageのhide-global-notification-dialogにtrueを設定するJavaScriptコードを実行するアクションを定義します。

ページ番号9999はログイン・ページです。結果としてログイン・ページではダイアログは表示されず、また、一度ダイアログが表示されるとブラウザを終了するまではダイアログが再表示されることはありません。

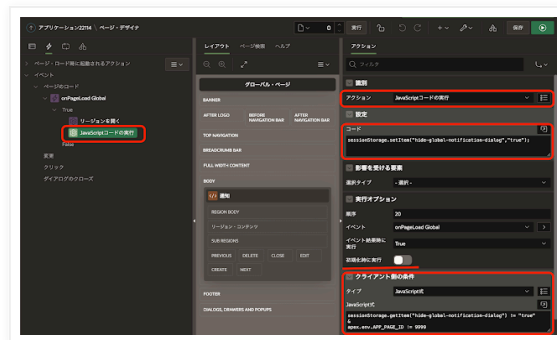


追加でTRUEアクションを作成します。

作成したTRUEアクションの識別のアクションとして、JavaScriptコードの実行を選択します。設定のコードに以下を記述します。

```
sessionStorage.setItem("hide-global-notification-dialog","true");
```

クライアント側の条件は、先行するTRUEアクションリージョンを開くと同じ設定にします。アクションリージョンを開くが実行されていれば、hide-global-notification-dialogにtrueが設定されます。



以上で通知の実装は完了です。

今回作成した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/demo-notification-global.sql>

<https://github.com/ujnak/apexapps/blob/master/exports/demo-notification-javascript.sql>

<https://github.com/ujnak/apexapps/blob/master/exports/demo-notification-plsql.sql>

<https://github.com/ujnak/apexapps/blob/master/exports/demo-notification-dialog.sql>

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: 18:05

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

