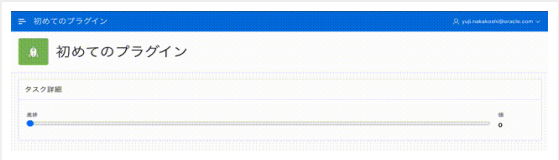
日日是Oracle APEX

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。

2021年3月13日土曜日

APEX Instant Tips #16のCreate an APEX plugin from scratch!を見てプラグインを作成する

カナダのInsumというAPEXのパートナーが配信しているAPEX Instant Tipsの16回目でStephan Dobreさんが５分でアイテム・プラグインを作っていました。



そのビデオに沿って作業を行った記録です。

APEX Instant Tips #16 Create an APEX plugin from scratch!

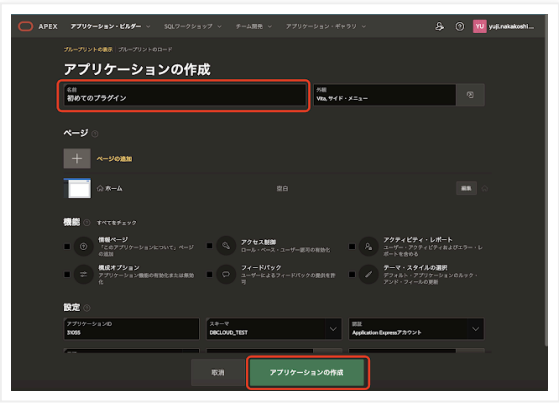
https://www.youtube.com/watch?v=jLA5NGnhSu0

一番最初にブラウザにオーストリアのFOEX GmbHという、こちらもAPEXのパートナーが作成したプラグインをブラウザに導入します。

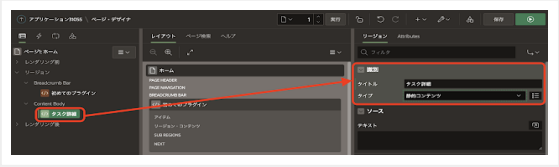
https://www.foex.at/apex-builder-extension/

このプラグインをブラウザに導入することにより、ビデオの最初で紹介しているBoilerplate Code(プラグインのサンプル・コード）を参照できるようになります。とはいえ、この記事にはプラグインのコードを貼り付けるので、プラグインを入れなくても作業を進めることはできるでしょう。

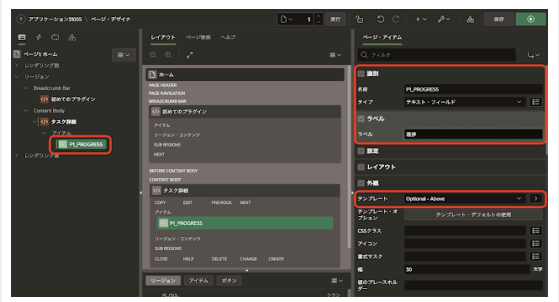
続く準備作業として、プラグインを作る前に、それを組み込むアプリケーションを作成します。この部分も５分間のビデオには含まれていません。アプリケーション作成ウィザードを起動し、**名前**を**初めてのプラグイン**として、**アプリケーションの作成**を実行します。



アプリケーションが作成されたら、**ホーム**・ページを開き、**タイプ**が**静的コンテンツ**のリージョンを追加します。**名前**は**タスク詳細**とします。

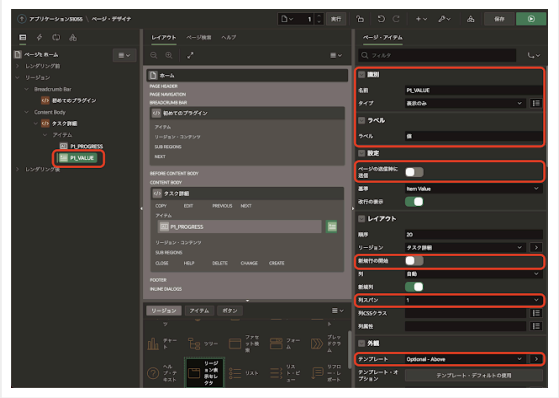


作成したリージョンにページ・アイテムを２つ作成します。最初に進捗を入力するページ・アイテム**P1_PROGRESS**を作成します。こちらのページ・アイテムを、これから作成するプラグインにより、数値ではなくゲージによって値を設定できるようにします。最初は**タイプ**を**テキスト・アイテム**とします。**ラベル**には**進捗**を設定します。**外観**の**テンプレート**として、**Optional - Above**を設定します。



もうひとつ、ページ・アイテムP1_PROGRESSに設定された値を表示するページ・アイテム**P1_VALUE**を作成します。**タイプ**は**表示のみ**とします。**ラベル**は**値**と設定します。

これ以外の設定は、アプリケーションの動作には直接関係しませんが、少しは見た目も良くなるように次の設定をおこないます。**設定**の**ページの送信時に送信**は**OFF**です。表示される元の値はP1_PROGRESSなので、P1_VALUEを送信に含める必要はありません。（ただ、どちらにしてもページの送信は行わないので影響はありません。）**レイアウト**の**新規行の開始**は**OFF**とし、P1_PROGRESSと同じ行に配置します。**列スパン**は**1**として、アイテムとしては最小の幅となるようにします。最後に**外観**の**テンプレート**として、P1_PROGRESSと同じ**Option - Above**を選択します。



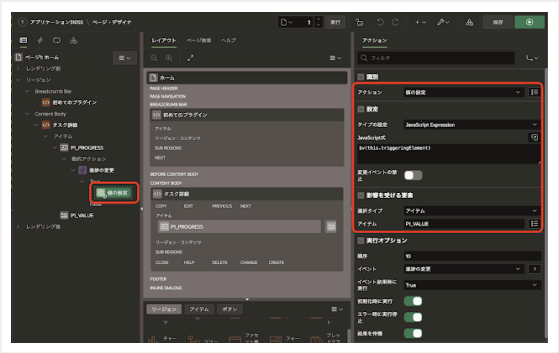
ページ・アイテムP1_PROGRESSの値が変更されたときに、P1_VALUEへ反映させる動的アクションを作成します。P1_PROGRESSの上でコンテキスト・メニューを表示し、**動的アクションの作成**を実行します。

動的アクションの**名前**を**進捗の変更**とします。**タイミング**はデフォルトで、**イベント**が**変更**、**選択タイプ**は**アイテム**、**アイテム**が**P1_PROGRESS**となっているので、そのまま使用します。

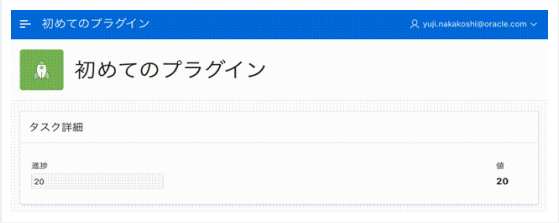
True**アクション**として**値の設定**を選択し、**設定**の**タイプの設定**をJavaScript Expression、**JavaScript式**として

$v(this.triggeringElement)

を指定します。**影響を受ける要素**として、**選択タイプ**を**アイテム**、**アイテム**は**P1_VALUE**を選択します。動的アクションはページ・アイテムP1_PROGRESSの変更をイベントとしているので、this.triggeringElementはP1_PROGRESSです。$vファンクションにより値を取得して、影響を受ける要素であるP1_VALUEへ、その値を設定します。



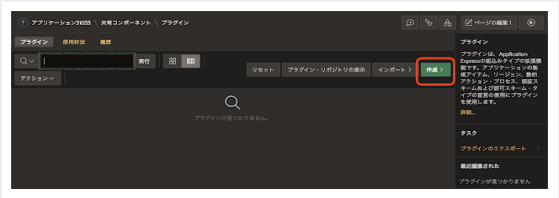
以上でプラグインを作成する前準備が完了しました。アプリケーションを実行して動作を確認してみます。



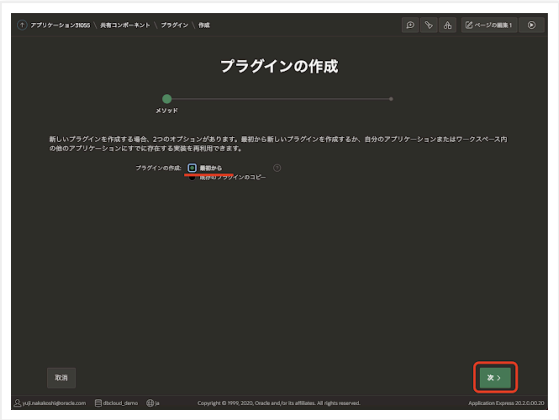
これから本題のプラグイン作成に入ります。Stephan Dobreさんは５分で作っていました。

共有コンポーネントから**プラグイン**を開きます。

作成済みのプラグインの一覧より、**作成**を実行します。



プラグインの作成は**最初**からを選び、**次**に進みます。



名前をInstant Slider、**内部名**は**INSTANT_SLIDER**、**タイプ**を**アイテム**とします。プラグインの実体となるPL/SQLコードをこれから書いていきますが、一から記述するのは大変です。

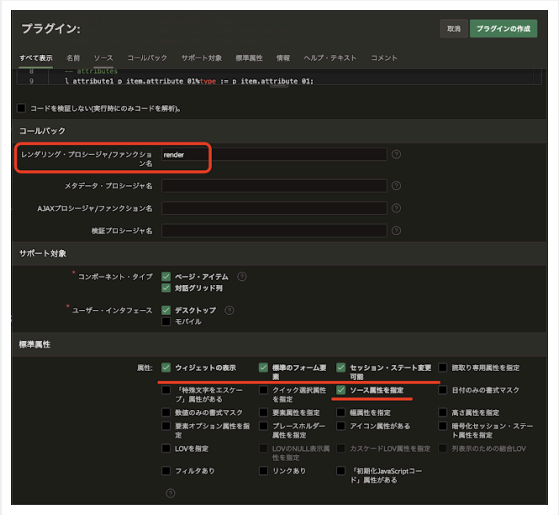
最初に説明したFOEXのプラグインが入っていると右ペインに**Boilerplate Code**のリンクが現れます。これをクリックします。

私のブラウザの問題なのか、Plug-in Boilerplate Codeのダイアログがきちんと表示されていませんが、**Plugin Type**として**Item**を選んで**Callback**として**Render**をクリックすると、**PL/SQL**のプロシージャとして**render**が表示されます。これを**PL/SQLコード**にコピペします。

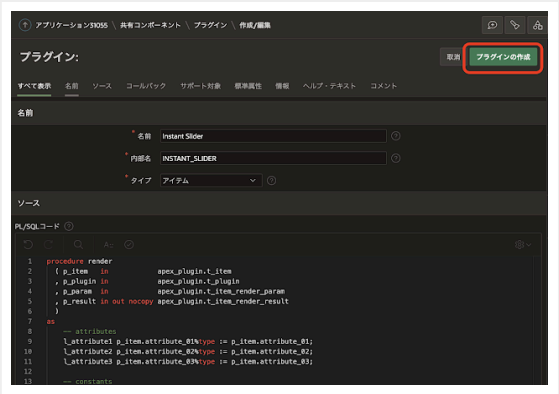

```
procedure render
  ( p_item   in           apex_plugin.t_item
  , p_plugin in           apex_plugin.t_plugin
  , p_param  in           apex_plugin.t_item_render_param
  , p_result in out nocopy apex_plugin.t_item_render_result
  )
as
    -- attributes
    l_attribute1 p_item.attribute_01%type := p_item.attribute_01;
    l_attribute2 p_item.attribute_02%type := p_item.attribute_02;
    l_attribute3 p_item.attribute_03%type := p_item.attribute_03;

    -- constants
    c_escaped_value constant varchar2(32767) := apex_escape.html(p_param.value);
    c_escaped_name  constant varchar2(32767) := apex_escape.html(p_item.name);
begin

    --debug
    if apex_application.g_debug
    then
        apex_plugin_util.debug_item_render
          ( p_plugin => p_plugin
          , p_item   => p_item
          , p_param  => p_param
          );
    end if;

    htp.p('<input
          class=""
          style="width:100%;"
          id="'    || c_escaped_name  || '"
          name="'  || c_escaped_name  || '"
          value="' || c_escaped_value || '"
          type="text"
          >');

end render;
```

コールバックのセクションの**レンダリング・プロシージャ/ファンクション名**として、PL/SQLコードに記載したプロシージャ名**render**を指定します。

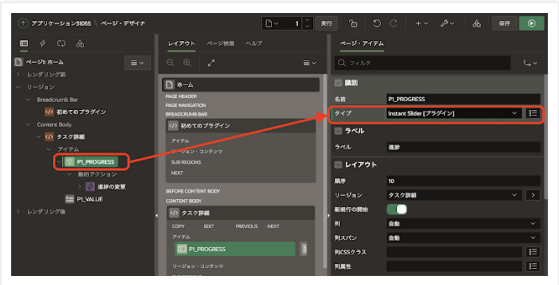
また、**標準属性**のセクションの**属性**の**ウィジェットの表示**、**標準のフォーム要素**、**セッション・ステート変更可能**、**ソース属性を指定**にチェックを入れます。



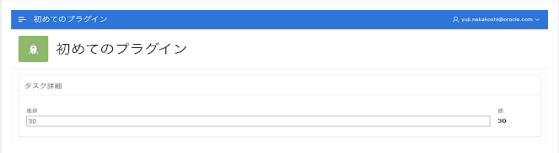
以上を設定して、**プラグインの作成**をクリックします。



プラグインが作成されたらページ・デザイナに戻り、ページ・アイテムP1_PROGRESSの**タイプ**を新規に作成したプラグインである**Instant Slider**に変更します。



アプリケーションを実行し、変更を確認します。まだボイラープレートのコードのままなので、ゲージの表示にはなっていません。



プラグインInstant Sliderを開いて、PL/SQLコードを修正します。

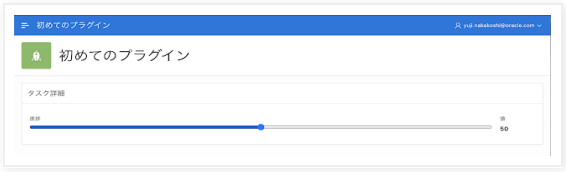
ボイラープレートのこの部分を

```
htp.p('<input
        class=""
        style="width:100%;"
        id="'    || c_escaped_name  || '"
        name="'  || c_escaped_name  || '"
        value="' || c_escaped_value || '"
        type="text"
        >');
```

typeをrangeに変更し、属性として**min**、**max**、**step**をそれぞれ**0**、**100**、**10**とします。

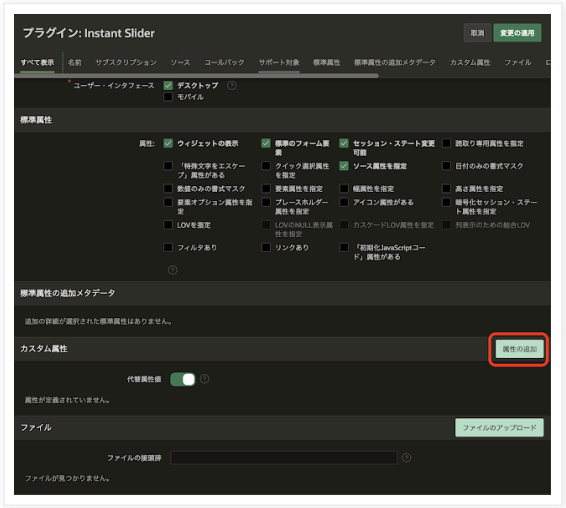
```
htp.p('<input
        class=""
        style="width:100%;"
        id="'    || c_escaped_name  || '"
        name="'  || c_escaped_name  || '"
        value="' || c_escaped_value || '"
        type="range"
        min="0"
        max="100"
        step="10"
        >');
```

ページを実行して、変更を確認します。

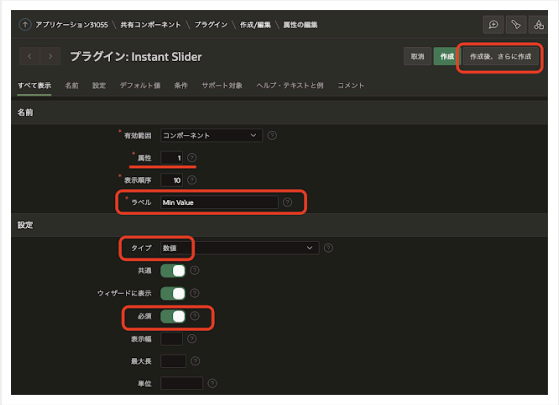


これで大体は完成です。しかし、プライグインのコード中にmin、max、stepが0、100、10とハードコードされています。これをアイテムのプロパティとして設定できるようにします。

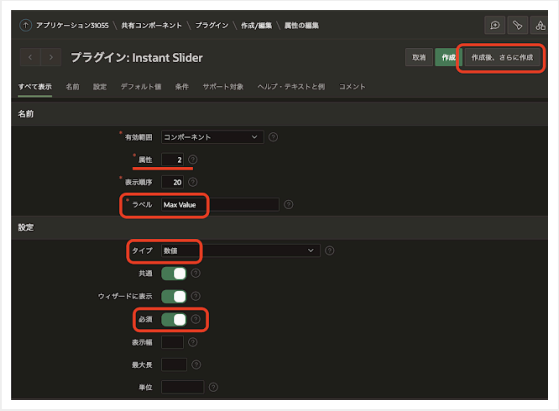
プラグインの編集ページの下の方に**カスタム属性**というセクションがあります。このカスタム属性として**属性の追加**を実行します。



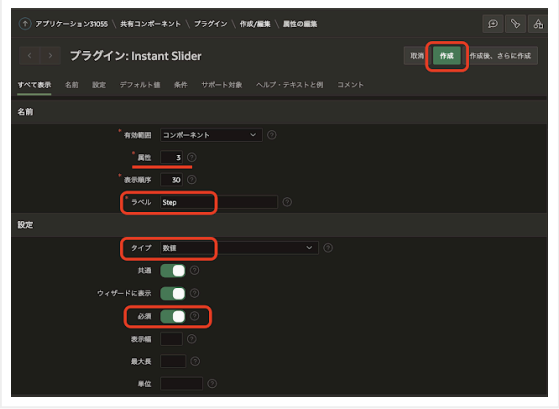
最初は**属性**の１番を、**ラベル**が**Min Value**となるように登録します。**設定**の**タイプ**は**数値**、**必須**を**ON**にします。**作成後、さらに作成**をクリックします。



次に**属性**の２番を、**ラベル**が**Max Value**となるように登録します。**設定**の**タイプ**は**数値**、**必須**を**ON**にします。**作成後、さらに作成**をクリックします。



最後に**属性**の３番を、**ラベル**が**Step**となるように登録します。**設定**の**タイプ**は**数値**、**必須**を**ON**にします。**作成**をクリックします。



これでカスタム属性はすべて設定しました。これらのカスタム属性でハードコードされた数値を置き換えるように、プロシージャrenderを修正します。

カスタム属性を参照するためのコードは、最初からボイラープレートに含まれています。

```
l_attribute1 p_item.attribute_01%type := p_item.attribute_01;
l_attribute2 p_item.attribute_02%type := p_item.attribute_02;
l_attribute3 p_item.attribute_03%type := p_item.attribute_03;
```

しかし属性番号で定義された変数では、コード中で扱いにくいため、l_min、l_max、l_stepへ変数定義を変更します。クロス・サイト・スクリプティングから守るためapex_escape.htmlファンクションも使用します。

```
   l_min p_item.attribute_01%type := apex_escape.html(p_item.attribute_01);
   l_max p_item.attribute_02%type := apex_escape.html(p_item.attribute_02);
   l_step p_item.attribute_03%type := apex_escape.html(p_item.attribute_03);
```

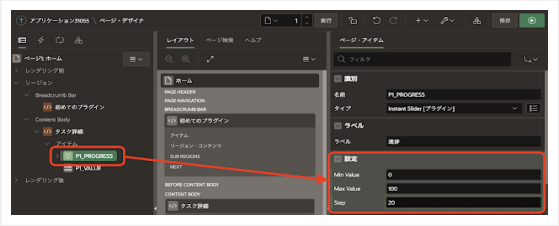
続いて、このカスタム属性を使用してハードコードの部分を書き換えます。

```
   htp.p('<input
         class=""
         style="width:100%;"
         id="'    || c_escaped_name  || '"
         name="'  || c_escaped_name  || '"
         value="' || c_escaped_value || '"
         type="range"
         min="'  || l_min  || '"
         max="'  || l_max  || '"
         step="' || l_step ||'"
         >');
```

コード全体は次のように変更されます。

```
procedure render
  ( p_item   in           apex_plugin.t_item
  , p_plugin in           apex_plugin.t_plugin
  , p_param  in            apex_plugin.t_item_render_param
  , p_result in out nocopy apex_plugin.t_item_render_result
  )
as
   -- attributes
   l_min p_item.attribute_01%type := apex_escape.html(p_item.attribute_01);
   l_max p_item.attribute_02%type := apex_escape.html(p_item.attribute_02);
   l_step p_item.attribute_03%type := apex_escape.html(p_item.attribute_03);

   -- constants
   c_escaped_value constant varchar2(32767) := apex_escape.html(p_param.value);
   c_escaped_name  constant varchar2(32767) := apex_escape.html(p_item.name);
begin

   --debug
   if apex_application.g_debug
   then
      apex_plugin_util.debug_item_render
        ( p_plugin => p_plugin
        , p_item   => p_item
        , p_param  => p_param
        );
   end if;

   htp.p('<input
         class=""
         style="width:100%;"
         id="'    || c_escaped_name  || '"
         name="'  || c_escaped_name  || '"
         value="' || c_escaped_value || '"
         type="range"
         min="'  || l_min  || '"
         max="'  || l_max  || '"
         step="' || l_step ||'"
         >');

end render;
```

プラグインの変更を保存して、ページ・アイテムP1_PROGRESSのプロパティを確認します。**設定**に**Min Value**、**Max Value**、**Step**が増えているので、それぞれ**0**、**100**、**20**を設定します。



以上でプラグインと、それを使ったアプリケーションは完成です。

変更を保存してページを実行すると、最初のGIF動画のような動作を確認することができます。

今回作成したアプリケーションのエクスポートをこちらに置きました。

https://github.com/ujnak/apexapps/blob/master/exports/instantslider.sql

Oracle APEXのアプリケーション作成の一助になれば幸いです。

完

Yuji N. 時刻: 19:29

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