

# 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5月31日 火曜日

## モーダル・ダイアログを重ねてモーダル・ダイアログを開く

モーダル・ダイアログとして開いているページから、さらにモーダル・ダイアログを開く方法について質問を受けたので実装してみました。大まかに2通りの実装方法があります。また、標準のポップアップLOVのテンプレートを変更することで、要件を満たす場合もあるでしょう。

1. **インライン・ダイアログ**を使う。ダイアログは同じページに実装します。
2. **モーダル・ダイアログ**のページを使う。ダイアログは別のページとして実装します。
3. **ポップアップLOV**のテンプレートを変更する。

サンプル・データセットの表EMPを使った対話モード・レポートとフォームのページを作成し、上記の3通りの実装を紹介します。

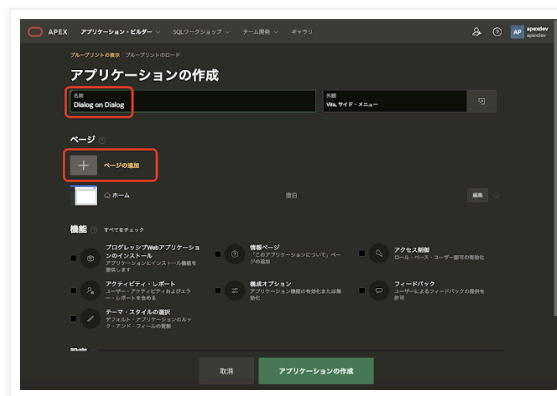
作成したアプリケーションは、以下のように動作します。

Employee Name	Job	Manager	Hire Date	Salary	Commission	Department
中島 登希子	社長		1981/11/17	5,000		会計
伊藤 明子	マネージャー	中島 登希子	1981/05/01	2,850		セールス
佐野 美穂	社員	伊藤 明子	1981/12/03	950		セールス
坂本 明	社員	中島 登希子	1981/05/09	2,450		会計
増田 秀樹	マネージャー	佐野 美穂	1981/01/12	1,100		研究開発
渡辺 大介	セールス	伊藤 明子	1981/02/22	1,200	500	セールス
新井 敦子	アナリスト	関口 英	1982/12/09	3,000		研究開発
村上 肇子	セールス	伊藤 明子	1981/02/20	1,600	300	セールス
村田 淳	役員	石橋 英	1982/12/17	800		研究開発
石橋 和夫	役員	坂本 明	1982/01/23	1,300		会計
石橋 英	アナリスト	関口 英	1981/12/03	3,000		研究開発
金子 恵美	セールス	伊藤 明子	1981/06/08	1,500	9	セールス
関口 英	マネージャー	中島 登希子	1981/04/02	2,975		研究開発
渡辺 大輔	セールス	伊藤 明子	1981/08/28	1,350	1400	セールス

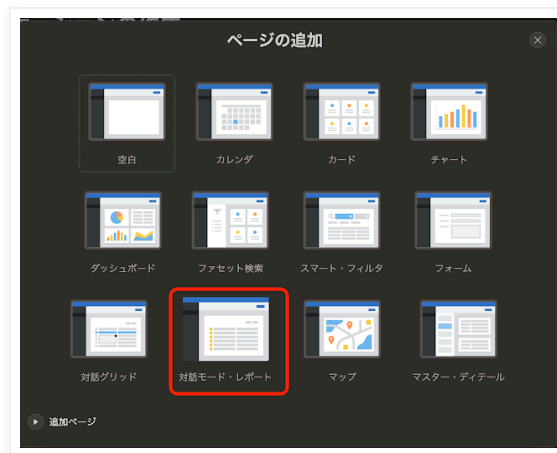
## 準備

アプリケーション作成ウィザードを起動します。アプリケーションの名前はDialog on Dialogとします。

ページの追加をクリックし、対話モード・レポートとフォームのページを追加します。



対話モード・レポートを選択します。



ページ名をインライン・ダイアログとします。表またはビュー、対話モード・レポートを選択し、表またはビューとして表EMPを選択します。フォームを含めるにチェックを入れ、表EMPを編集するフォームのページを作成します。

このフォームのページを、親となるモーダル・ダイアログのページとします。このフォームのページから、さらにモーダル・ダイアログを開く実装を行います。

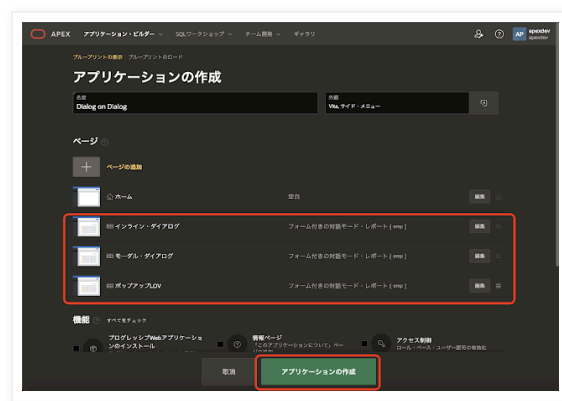
ページの追加を実行します。



ページ名をモーダル・ダイアログとして、同じページを追加します。さらにページ名をポップアップLOVとして、同じページを追加します。

インライン・ダイアログ、モーダル・ダイアログ（別ページ）、ポップアップLOVを使った実装を試すページがそれぞれ追加されました。

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ます。インライン・ダイアログを使った実装はページ番号3、モーダル・ダイアログを使った実装はページ番号5、ポップアップLOVはページ番号7のページに実施します。

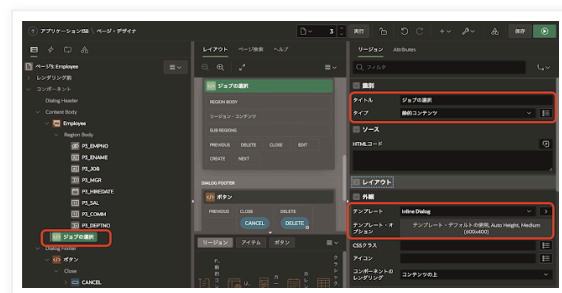


## インライン・ダイアログによる実装

ページ・デザイナーにて、ページ番号3のページを開きます。

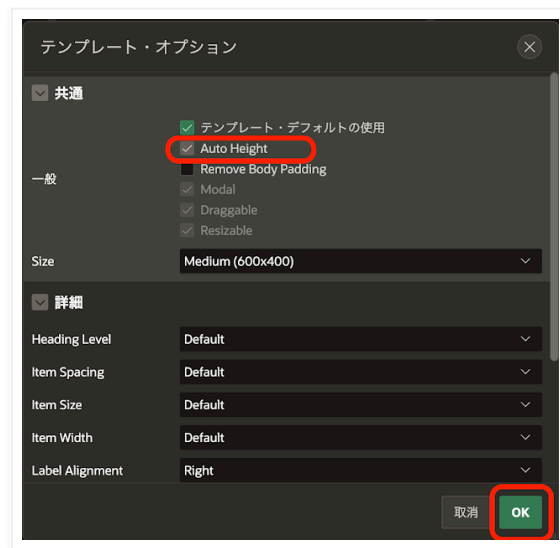
Content Body上でコンテキスト・メニューを開き、リージョンの作成を実行します。このリージョンをダイアログにします。

作成したリージョンの識別のタイトルをジョブの選択とします。タイプは静的コンテンツとします。外観のテンプレートにInline Dialogを選択すると、このリージョンがダイアログとして扱われます。



テンプレート・オプションをクリックし、Auto Heightにチェックを入れます。これ以外にも見かけを調整するオプションがあるので、必要に応じて調整します。

OKをクリックして、変更を確定します。

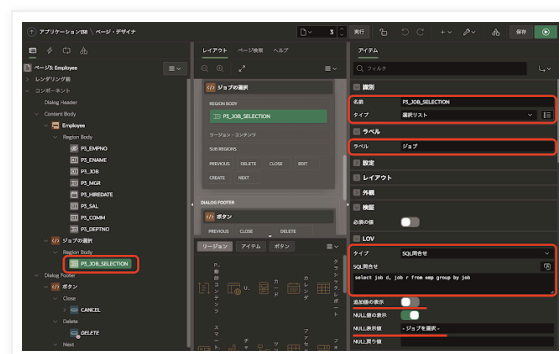


リージョン**ジョブの選択**に、ジョブを選択するページ・アイテムP3\_JOB\_SELECTIONを作成します。

識別の名前をP3\_JOB\_SELECTION、タイプは選択リストとします。ラベルはジョブとします。LOVのSQL問合せとして、以下のSELECT文を記述します。

```
select job d, job r from emp group by job
```

追加値の表示はOFF、NULL値の表示はON、NULL表示値として- ジョブを選択 -を記述します。

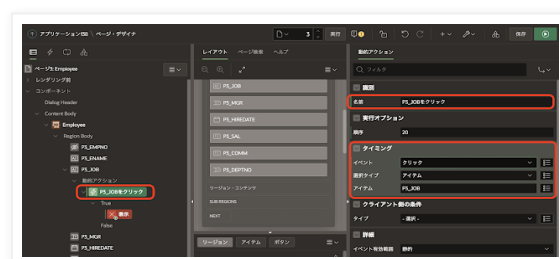


インライン・ダイアログの操作は、すべて動的アクションで実装します。この処理にはインライン・ダイアログのリージョンを開く、閉じる、値の設定が含まれます。

ページ・アイテムP3\_JOBをクリックしたときにリージョン**ジョブの選択**を開きます。ページ・アイテムP3\_JOBの内容を、ダイアログを開くと表示されるページ・アイテムP3\_JOB\_SELECTIONを使って選択できるようにします。

ページ・アイテムP3\_JOBにたいして、動的アクションを作成します。

識別の名前はP3\_JOBをクリックとします。タイミングのイベントはクリック、選択タイプはアイテム、アイテムはP3\_JOBになります。



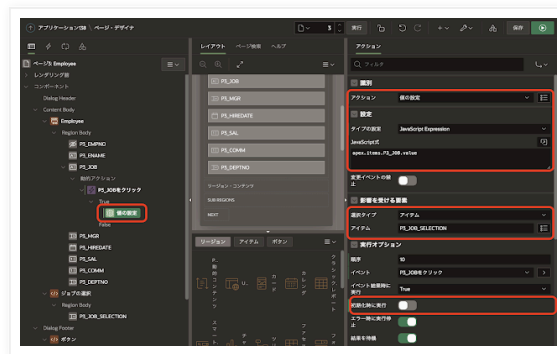
すでにページ・アイテムP3\_JOBに設定されている値を、ページ・アイテムP3\_JOB\_SELECTIONにコピーします。

TRUEアクションの識別のアクションとして値の設定を選択します。設定のタイプの設定にJavaScript Expressionを選び、JavaScript式として以下を記述します。ページ・アイテムP3\_JOBの内容になります。

`apex.items.P3_JOB.value`

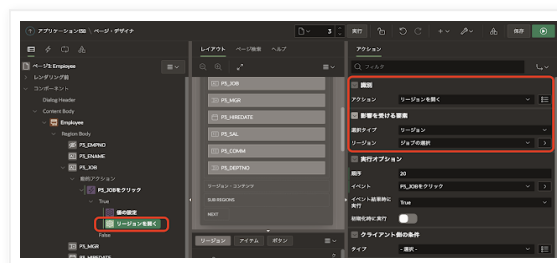
影響を受ける要素の選択タイプにアイテム、アイテムにP3\_JOB\_SELECTIONを選びます。これで、P3\_JOBの値がページ・アイテムP3\_JOB\_SELECTIONに設定されます。

P3\_JOBをクリックしたときにのみ実行させるため、初期化時に実行はOFFにします。



続いて、リージョンジョブの選択を開きます。

TRUEアクションを作成します。識別のアクションに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はリージョン、リージョンとしてジョブの選択を設定します。



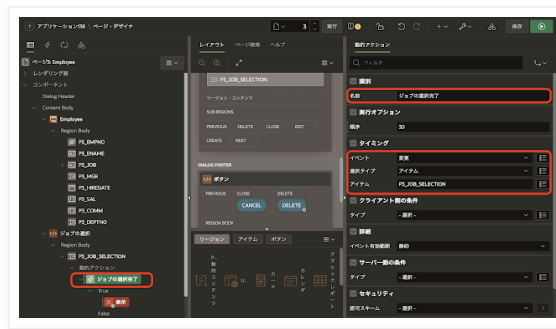
以上で、インライン・ダイアログを開く動的アクションが実装できました。

今度はインライン・ダイアログを閉じる処理を追加します。

ページ・アイテムP3\_JOB\_SELECTIONで値を選択した時点で、インライン・ダイアログを閉じるようにします。

ページ・アイテムP3\_JOB\_SELECTIONにたいして、動的アクションを作成します。

識別の名前をジョブの選択完了とします。タイミングのイベントは変更、選択タイプはアイテム、アイテムはP3\_JOB\_SELECTIONとなります。

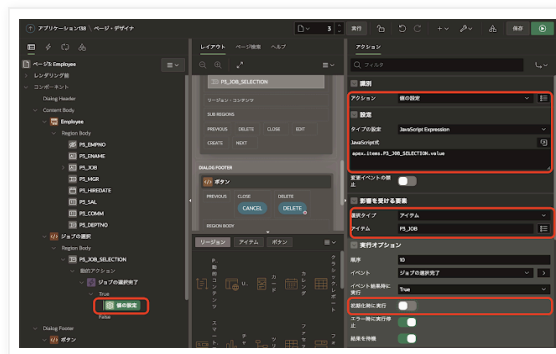


TRUEアクションは、ダイアログを開けた時とは反対に、ページ・アイテムP3\_JOB\_SELECTIONの値をページ・アイテムP3\_JOBに設定します。

識別のアクションとして、値の設定を選択します。設定のタイプの設定はJavaScript Expression、JavaScript式として、以下を記述します。

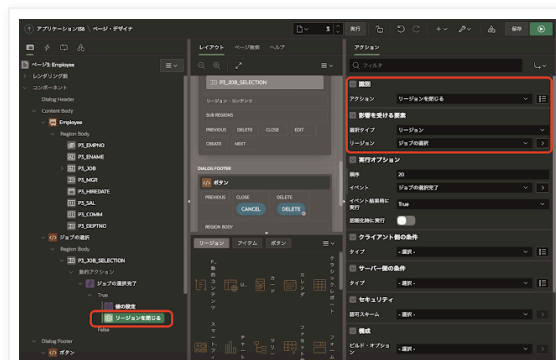
`apex.items.P3_JOB_SELECTION.value`

影響を受ける要素の選択タイプはアイテム、アイテムとしてP3\_JOBを選択します。このTRUEアクションでも、初期化時に実行はOFFにします。



TRUEアクションを作成して、値の設定の次に配置します。

識別のアクションとして、リージョンを閉じるを選択します。影響を受ける要素の選択タイプはリージョン、リージョンとしてジョブの選択を選びます。



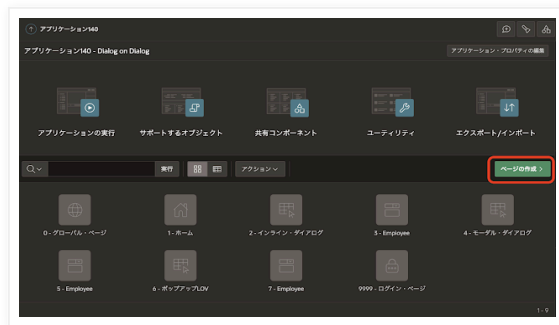
以上で、インライン・ダイアログを使った実装は完了です。

アプリケーションを実行すると、記事の先頭にあるGIF動画のような動作になります。

## モーダル・ダイアログによる実装

ジョブの選択を異なるページに実装します。ページは、モーダル・ダイアログとします。

ページの作成を実行します。

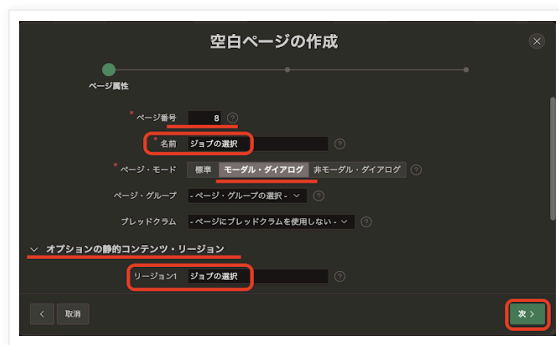


空白ページを選択します。



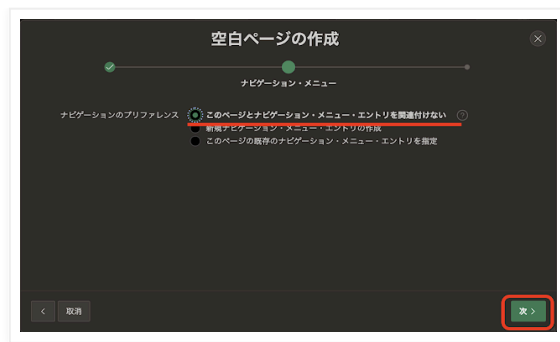
ページ番号は8、名前はジョブの選択、ページ・モードはモーダル・ダイアログとします。オプションの静的コンテンツ・リージョンを開き、リージョン1にジョブの選択と入力します。

次へ進みます。

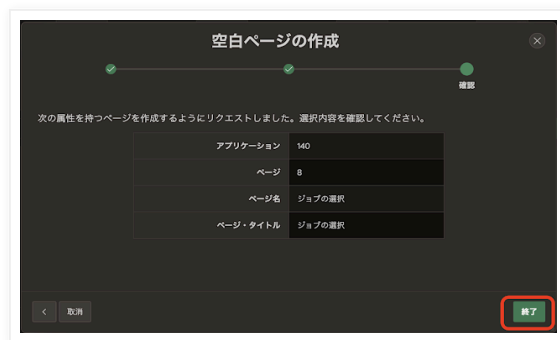


モーダル・ダイアログなので、ナビゲーションのプリファレンスは、このページとナビゲーション・メニュー・エントリを関連付けないから変更しません。

次へ進みます。

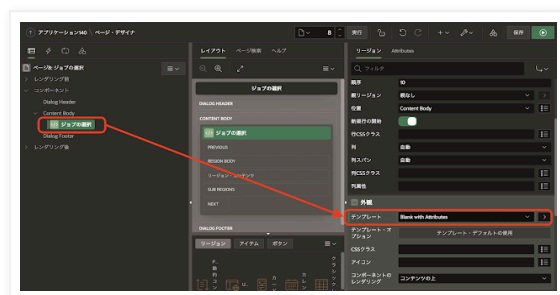


終了をクリックし、ページの作成を完了します。

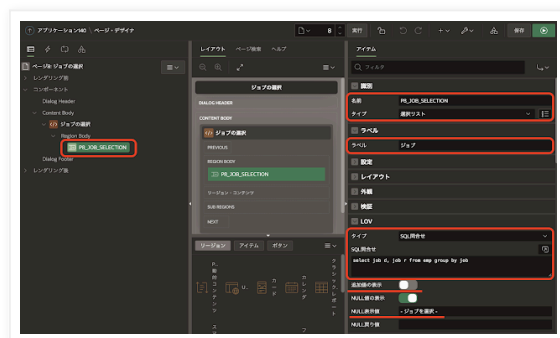


ページが作成されます。

あらかじめ作成されているリージョンジョブの選択の、外観のテンプレートをBlank with Attributesへ変更します。外観のテンプレートがStandardの場合、ページのタイトルとリージョンのタイトルの両方が画面に表示されます。Blank with Attributesに変更することにより、リージョンのタイトルは表示されなくなり、ページのタイトルのみが表示されます。



ジョブを選択するページ・アイテムP8\_JOB\_SELECTIONを作成します。これは、インライン・ダイアログの際に作成したP3\_JOB\_SELECTIONとまったく同じ設定です。

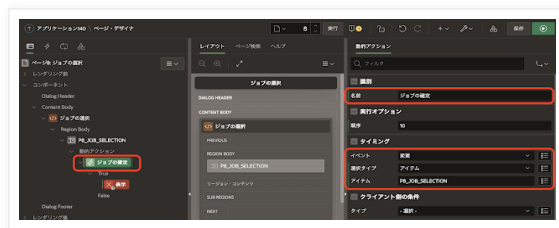


ページ・アイテムP8\_JOB\_SELECTIONの値が変更されたときに、ページの送信を行います。

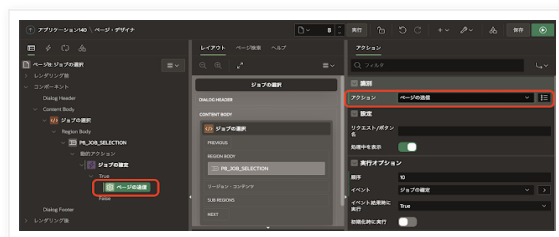
P8\_JOB\_SELECTIONに動的アクションを作成します。



識別の名前をジョブの確定とします。タイミングのイベントは変更、選択タイプはアイテム、アイテムはP8\_JOB\_SELECTIONです。



TRUEアクションの識別のアクションとして、ページの送信を選択します。



動的アクションでページの送信を行なう代わりに、ボタンを作成してページの送信を行なうこともできます。

ページが送信されたら、モーダル・ダイアログを閉じます。

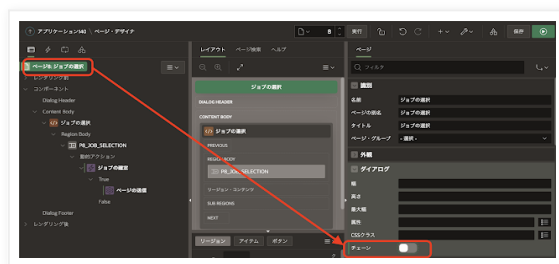
左ペインでプロセス・ビューを開き、プロセスを作成します。

識別の名前はダイアログを閉じる、タイプとしてダイアログを閉じるを選択します。設定の戻すアイテムにP8\_JOB\_SELECTIONを指定します。



ページ・プロパティのダイアログのチェーンをOFFにします。(レポートと同時にフォームを作成したときはチェーンのデフォルトはOFFですが、単体のページでモーダル・ダイアログを作成した場合はONがデフォルトのようです。)

チェーンがONの場合、このモーダル・ダイアログのページは親のモーダル・ダイアログの画面を置き換えます。(同じiframeを使用する) チェーンがOFFの場合は、親のモーダル・ダイアログの中で、モーダル・ダイアログが開かれます。



子となるモーダル・ダイアログは、以上で出来上がりです。

親のページから呼び出す設定を行います。

ページ・デザイナーにて、ページ番号**5**のフォームのページを開きます。

インライン・ダイアログのようにページ・アイテムをクリックしたときに、別のページを開く実装は簡単ではないです(JavaScriptで[apex.navigation.dialog](#)を呼び出す)。今回は別ページを開くボタンを作成します。

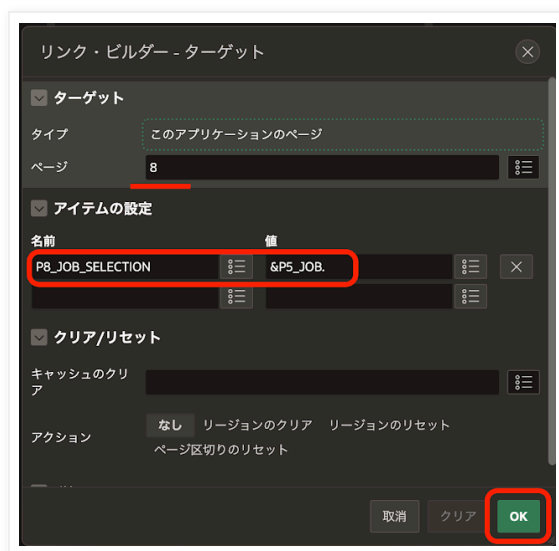
識別のボタン名は**B\_SELECT\_JOB**とします。ラベルは**ジョブの選択**とします。動作のアクションとして、このアプリケーションのページにリダイレクトを選びます。



ターゲットをクリックし、**リンク・ビルダー・ターゲット**を開きます。

ターゲットのページは**8**、アイテムの設定の名前に**P8\_JOB\_SELECTION**、値には**&P5\_JOB.**を指定します。値に**&P5\_JOB.**が設定されている影響について、後述しています。

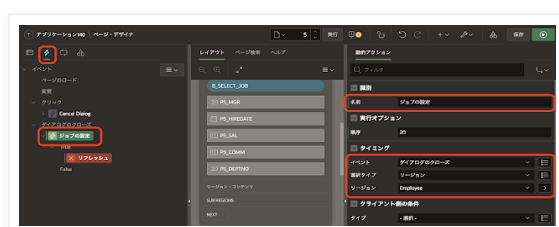
OKをクリックして、変更を保存します。



ダイアログがクローズしたときに返される値（P8\_JOB\_SELECTIONの値）を、ページ・アイテムP5\_JOBに設定する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、**ダイアログのクローズ**で動的アクションを作成します。

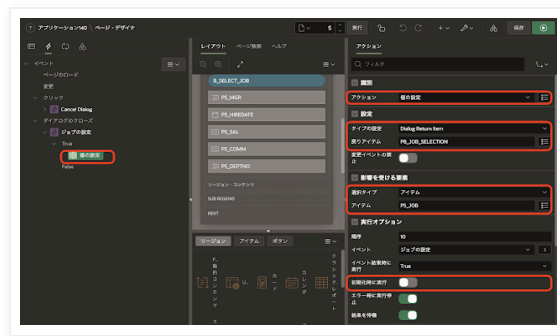
識別の名前を**ジョブの設定**とします。タイミングのイベントは**ダイアログのクローズ**、選択タイプは**リージョン**、リージョンとして**Employees**を選びます。



TRUEアクションの識別のアクションとして、値の設定を選択します。設定のタイプの設定はDialog Retrun Item、戻りアイテムはP8\_JOB\_SELECTIONになります。戻りアイテムは異なるページに存在するため、選択リストから選ぶことはできません。キーボードから入力します。影響を受ける要素の選択タイプはアイテム、アイテムとしてP5\_JOBを指定します。

ページ番号8のモーダル・ダイアログが閉じるときに戻すアイテムとして指定したP8\_JOB\_SELECTIONの値を受け取って、ページ・アイテムP5\_JOBに設定しています。

このアクションも、初期化時に実行はOFFです。



以上で、別ページのモーダル・ダイアログによる実装は完了です。

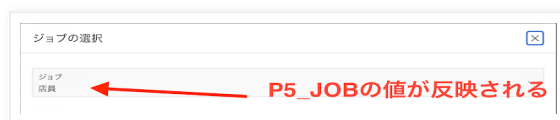
インライン・ダイアログの実装とあまり変わらないように見えますが、ボタン**ジョブの選択**をクリックしたときに、HTTPのGETリクエストが発行されています。インライン・ダイアログはページ上のJavaScriptの処理で、HTTPのリクエストは発生していません。

わかりやすい違いとして、以下に例を挙げます。

親のモーダル・ダイアログが開いた直後に**ジョブの選択**をクリックします。



開いたモーダル・ダイアログの値は**店員**です。

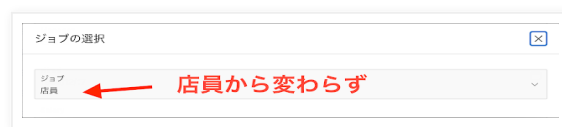


ページ・アイテムP8\_JOB\_SELECTIONに**セールス**を選択しダイアログを閉じると、ジョブはセールスに変わります。

再度、**ジョブの選択**をクリックします。



ページ・アイテムP8\_JOB\_SELECTIONの初期値は店員のまま、変わりません。



ボタン**ジョブの選択**のターゲットとして設定した引数**&P5\_JOB.**の値は、親のモーダル・ダイアログのページが開いたときに初期化されます。それ以降にページ・アイテムP5\_JOBの値が変更されても、ボタンのターゲットには反映されません。

## ポップアップLOVのテンプレート変更

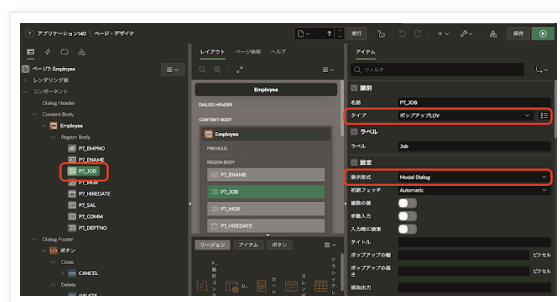
ポップアップLOVのテンプレートを変更する方法については、Louis Moreauxさんの以下の記事を参考にしています。

### Popup LOV custom template

<https://lmoreaux.hashnode.dev/popup-lov-custom-template>

ページ・デザイナーでページ番号7のページを開きます。

ジョブのページ・アイテムP7\_JOBのタイプをポップアップLOVに変更し、設定の表示形式をModal Dialogにします。



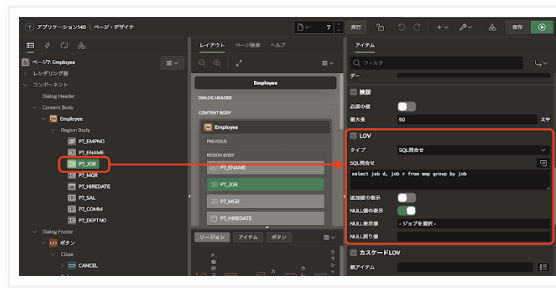
ページ・アイテムP7\_JOBのLOVの設定は、今までにP3\_JOB\_SELECTIONやP8\_JOB\_SELECTIONに設定した内容と同じです。

タイプはSQL問合せ、SQL問合せとして以下を記述します。

```
select job d, job r from emp group by job
```

検索された列名はDとRです。これはテンプレート内の置換文字列として使用します。

追加値の表示はOFF、NULL値の表示はON、NULL表示値として- ジョブを選択 -を記述します。



ポップアップLOVに一覧されるジョブを、太字、赤、センタリングして表示するよう、テンプレートを変更します。

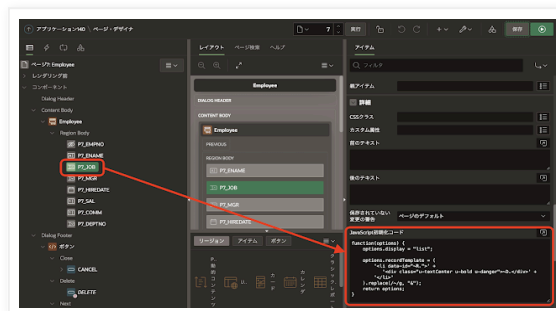
以下のJavaScriptコードを、ページ・アイテムP7\_JOBの詳細のJavaScript初期化コードに記述します。

```
function(options) {
    options.display = "list";

    options.recordTemplate = (
        '<li data-id="~R.">' +
        '<div class="u-textCenter u-bold u-danger">~D.</div>' +
        '</li>'
    ).replace(/~/g, "&");
    return options;
}
```

popup-lov-custom-template.js hosted with ❤ by GitHub

[view raw](#)



ポップアップLOVへのカスタム・テンプレートの適用は以上です。

アプリケーションを実行して、ページ・アイテムP7\_JOBをポップアップさせると、以下のように表示されます。

元記事では、画像を含んだもっと見栄えのよいカスタム・テンプレートを用意しています。より詳しく知りたい場合は、英語ですが元記事の参照をお勧めします。

以上になります。

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/dialog-on-dialog-sample.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: 17:11

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.