# 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。

2023年5月9日火曜日

## knockout.jsのチュートリアルをOracle APEXで実装してみる

ドイツ在住のMaik Michelさんが彼のブログにOracle APEXでknockout.jsを使用する方法を紹介しています。

Improve UX by using Knockout - show only functions that actually make sense

興味深い内容だったので、knockout.jsのサイトで紹介されているチュートリアルのIntroductionをOracle APEXで実装してみました。

チュートリアルのページは以下から始まります。

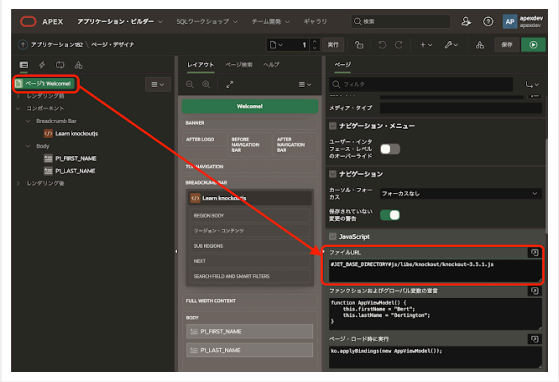
https://learn.knockoutjs.com/#/?tutorial=intro

Step1から5までありますが、最後のStep 5はまとめで実際の実装サンプルは１から４までです。

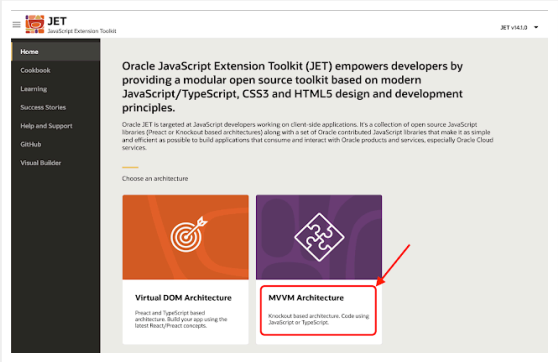
### 準備

Oracle APEXでknockout.jsを使用するために、**ページ・プロパティ**の**JavaScript**の**ファイルURL**に以下を記述します。バージョンについては、将来のAPEXのバージョンで変更される可能性があります。

**#JET_BASE_DIRECTORY#js/libs/knockout/knockout-3.5.1.js**



Oracle JETのサイトで紹介されているように、Oracle JETはMVVM Architectureを採用しており、knockoutを含んでいます。Oracle APEXは主にチャートの表示にOracle JETを採用しているため、Oracle APEXにもknockoutは含まれます。




## ステップ1：Welcome!の実装

最初の例ではHTML要素のテキスト・ノードの更新を行っています。

knockout.jsのチュートリアルでは、以下のHTMLを記述しています。

```
<p>First name: <strong data-bind="text: firstName"></strong></p>
<p>Last name: <strong data-bind="text: lastName"></strong></p>
```

Oracle APEXでは、これらの要素を**タイプ**が**表示のみ**の**ページ・アイテム**として作成することにします。

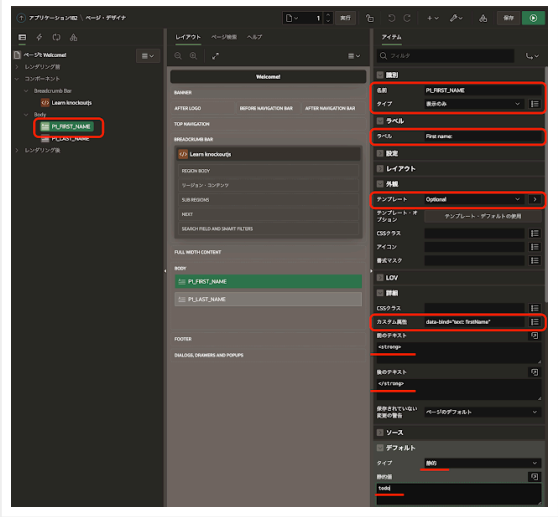
**First name**の要素は、**識別**の**名前**を**P1_FIRST_NAME**、**タイプ**を**表示のみ**、**ラベル**は**First name:**とします。ラベルと値が横並びになるように、**外観**の**テンプレート**に**Optional**を選択します。

**詳細**の**カスタム属性**に、knockout.jsが使用するカスタム属性**data-bind**を指定します。

**data-bind="text: firstName"**

**前のテキスト**に****、**後のテキスト**に****を記述します。これはknockout.jsのチュートリアルに寄せた設定で、Oracle APEXでは、ユニバーサル・テーマで定義されている強調表示のクラス**u-bold**を**詳細**の**CSSクラス**に設定する方が一般的です。

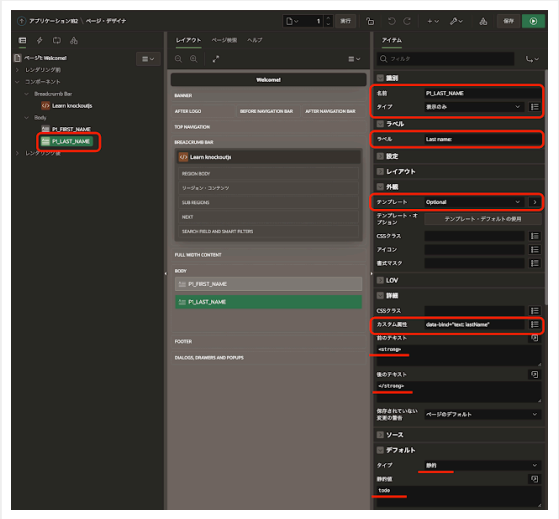
**デフォルト**の**タイプ**に**静的**を選択し、**静的値**として**todo**を設定します。これもknockout.jsのチュートリアルに寄せた設定です。



**Last name**の要素も同様に作成します。

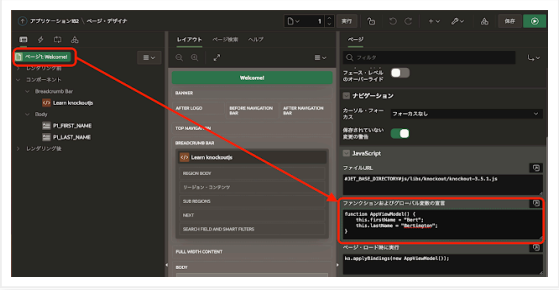
**識別**の**名前**は**P1_LAST_NAME**、**ラベル**は**Last name:**です。**詳細**の**カスタム属性**は以下になります。

**data-bind="text: lastName"**



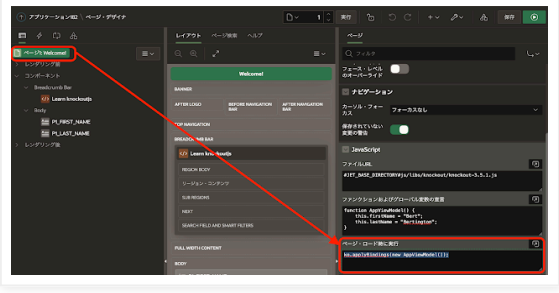
**ページ・プロパティ**の**JavaScript**の**ファンクションおよびグローバル変数の宣言**にて、ビューモデルを定義します。

```
function AppViewModel() {
    this.firstName = "Bert";
    this.lastName = "Bertington";
}
```



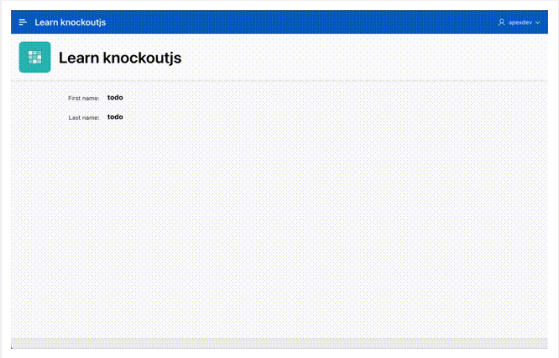
DOM要素にビューモデルをバインドするコードを、**ページ・ロード時に実行**に記述します。

```
ko.applyBindings(new AppViewModel());
```



以上の設定を行いページを実行します。

ページのロード時にページ・アイテム**P1_FIRST_NAME**、**P1_LAST_NAME**の値が、それぞれビュー・モデルで定義されている**Bert**、**Berington**に変更されています。




## Step 2：Making the data editableの実装

先に作成したWelcome!のページをコピーし、Step 2の実装を始めます。

チュートリアルでは、以下のインプット・フィールドを追加しています。

```
<p>First name: <input data-bind="value: firstName" /></p>
<p>Last name: <input data-bind="value: lastName" /></p>
```

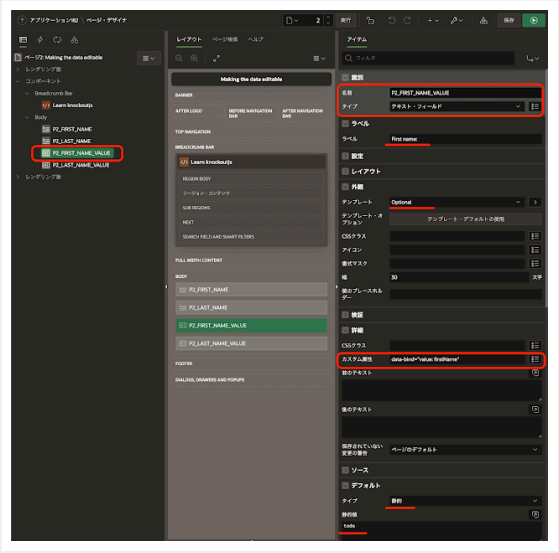
これらの要素を**タイプ**が**テキスト・フィールド**の**ページ・アイテム**として作成することにします。

**First name**の要素は、**識別**の**名前**を**P2_FIRST_NAME_VALUE**、**タイプ**を**テキスト・フィールド**とします。

**詳細**の**カスタム属性**のdata-bindでは、**text**の代わりに**value**を指定します。

**data-bind="value: firstName"**

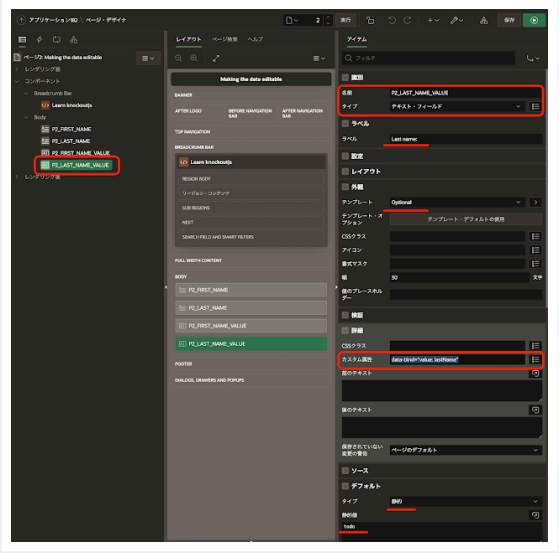
**詳細**の**前のテキスト**、**後のテキスト**の****は削除します。それ以外の設定は**表示のみ**のページ・アイテム**P2_FIRST_NAME**と同じです。



**Last name**の要素も同様に作成します。

**属性**の**名前**は**P2_LAST_NAME_VALUE**、**詳細**の**カスタム属性**は以下になります。

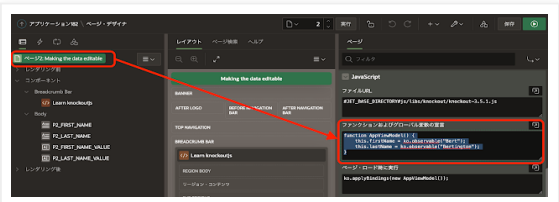
**data-bind="value: lastName"**



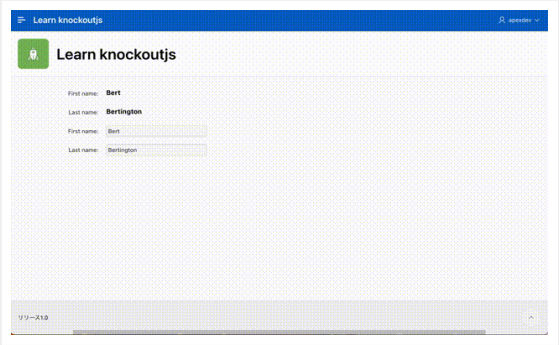
**ページ・プロパティ**の**JavaScript**の**ファンクションおよびグローバル変数の宣言**にて定義されているビューモデルを変更します。

**firstName**および**lastName**に**ko.observable**を指定し、自動的に変更を通知するようにしています。

```
function AppViewModel() {
    this.firstName = ko.observable("Bert");
    this.lastName = ko.observable("Bertington");
}
```



以上の設定を行いページを実行すると、ページ・アイテム**P2_FIRST_NAME_VALUE**、**P2_LAST_NAME_VALUE**が変更されると即座に**P2_FIRST_NAME**、**P2_LAST_NAME**に反映されることが確認できます。




今回の例ではビューモデルの定義をknockout.jsのチュートリアルに寄せましたが、Oracle APEXの一般的な実装では、ko.observableの引数として与える初期値はページ・アイテムの値となるでしょう。

コード例としては以下になります。

```
function AppViewModel() {
    this.firstName = ko.observable(apex.items.P2_FIRST_NAME_VALUE.value);
    this.lastName = ko.observable(apex.items.P2_LAST_NAME_VALUE.value);
}
```

## Step 3：Defining computed valuesの実装

Step 2のページをコピーし、Step 3の実装を始めます。

チュートリアルでは、以下の氏名を表示するフィールドを追加しています。

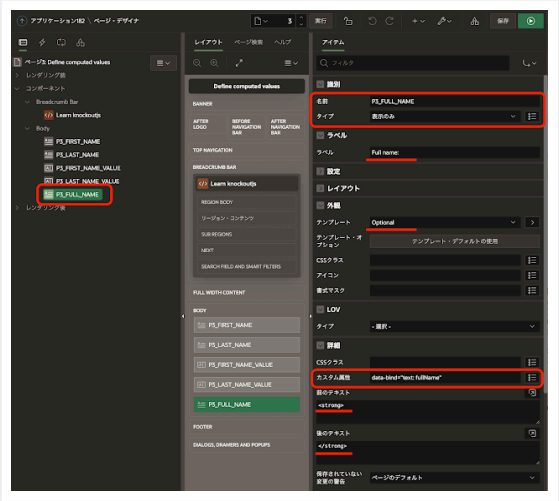
```
<p>Full name: <strong data-bind="text: fullName"></strong></p>
```

この要素を**タイプ**が**表示のみ**の**ページ・アイテム**として作成することにします。

**Full name**の要素は、**識別**の**名前**を**P3_FULL_NAME**、**タイプ**を**表示のみ**、**ラベル**を**Full name:**とします。

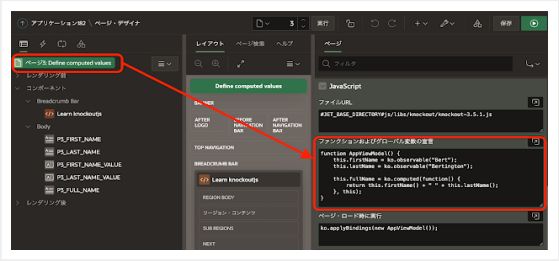
**詳細**の**カスタム属性**に以下を設定します。

**data-bind="text: fullName"**

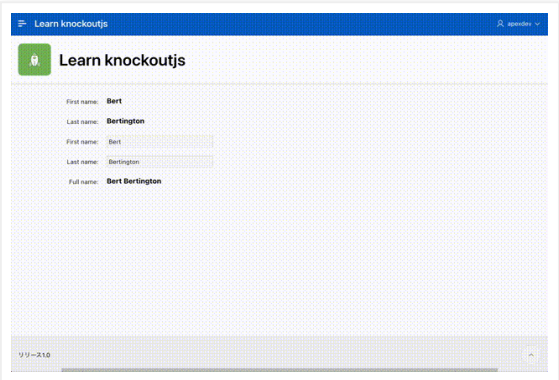


firstNameとlastNameを結合しfullNameを生成するように、ビューモデルを更新します。

```
function AppViewModel() {
    this.firstName = ko.observable("Bert");
    this.lastName = ko.observable("Bertington");

    this.fullName = ko.computed(function() {
        return this.firstName() + " " + this.lastName();
    }, this);
}
```



以上でページを実行すると、**First name**および**Last name**を変更した時点で、**Full name**が更新されることが確認できます。




## Step 4：Add more behaviorの実装

Step 3のページをコピーし、Step 4の実装を始めます。

チュートリアルでは、Last nameを大文字に変更するボタンを作成します。

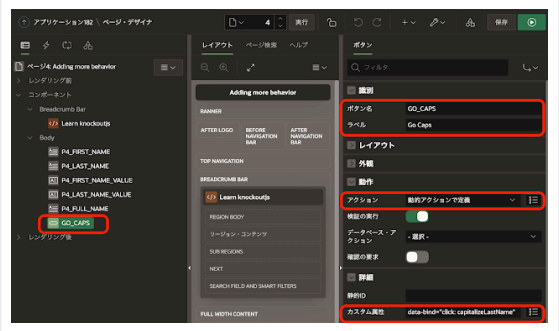
```
<button data-bind="click: capitalizeLastName">Go caps</button>
```

Oracle APEXでは上記を**ボタン**として作成します。

**識別**の**ボタン名**を**GO_CAPS**、**ラベル**は**Go Caps**とします。**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。動的アクションは定義しないので、実質的にはHTTPのGETやPOSTのリクエスト発行しない、何もしないボタンになります。

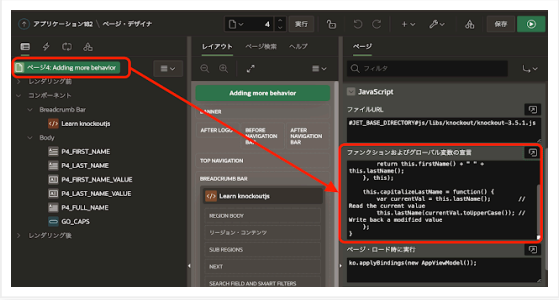
**詳細**の**カスタム属性**として以下を記述することにより、ボタンをクリックするとビューモデルで定義された処理が実行されます。

**data-bind="click: capitalizeLastName"**



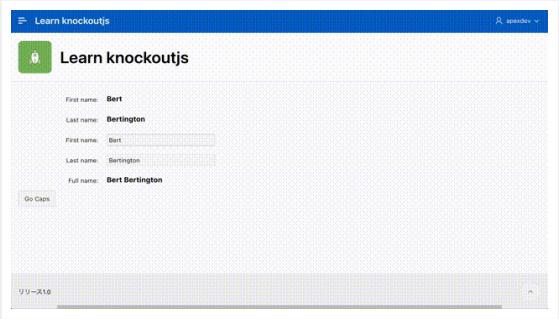
ビューモデルを更新します。Last nameを大文字に変更する処理を含めます。

```
function AppViewModel() {
    this.firstName = ko.observable("Bert");
    this.lastName = ko.observable("Bertington");

    this.fullName = ko.computed(function() {
        return this.firstName() + " " + this.lastName();
    }, this);

    this.capitalizeLastName = function() {
        var currentVal = this.lastName();        // Read the current value
        this.lastName(currentVal.toUpperCase()); // Write back a modified value
    };
}
```



以上の設定を行いページを実行します。

**Go Caps**のボタンをクリックすると、**Last name**が大文字に変更されることが確認できます。また、同時に**Full name**の変更も確認できます。




Oracle APEXでのknockout.jsのチュートリアルの実装は以上になります。

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/learn-knockoutjs.zip

本チュートリアルではknockout.jsのtext、value、clickの３つのバインディングを使用していますが、knockout.jsのドキュメントではこれ以外にも多数のバインディングが説明されています。

Oracle APEXでは機能として動的アクションが提供されていますが、knockout.jsの利用も良い選択肢となりそうです。

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: 14:51

共有



**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