



**HCL Volt MX**

# コントラクトなしの コンポーネント

HVMX-BC-500 Volt MX Iris Collaboration

# コンポーネント入門

- コンポーネントは、自己充足的で再利用可能なエンティティです。
- コンポーネントは、複雑なアプリケーションを迅速に構築するための強力な方法を提供します。
- コンポーネントをアプリケーションにドラッグ＆ドロップすることで、すべてのコードを記述することなく、洗練されたフル機能のアプリケーションを作成できます。
- コンポーネントには以下が含まれます。
  - コードモジュール
  - Foundryサービス
  - 使用方法を簡素化するためのコントラクト
  - 同じプロジェクト内やワークスペース内の複数のプロジェクトでの再利用性
  - ローカルでの共有とマーケットプレイスを介した幅広い層との共有

# コンポーネント紹介

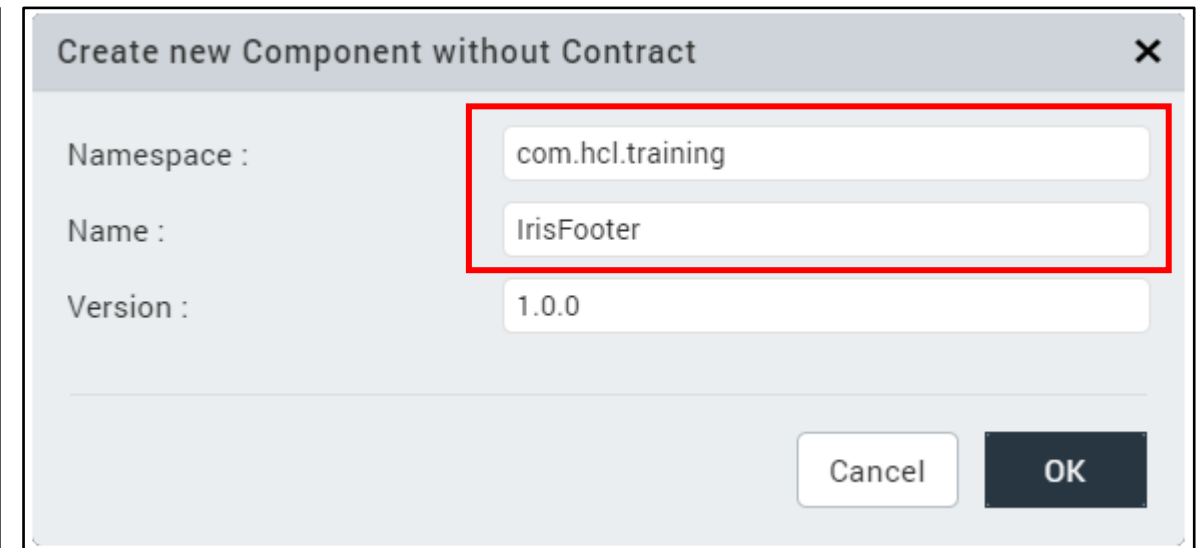
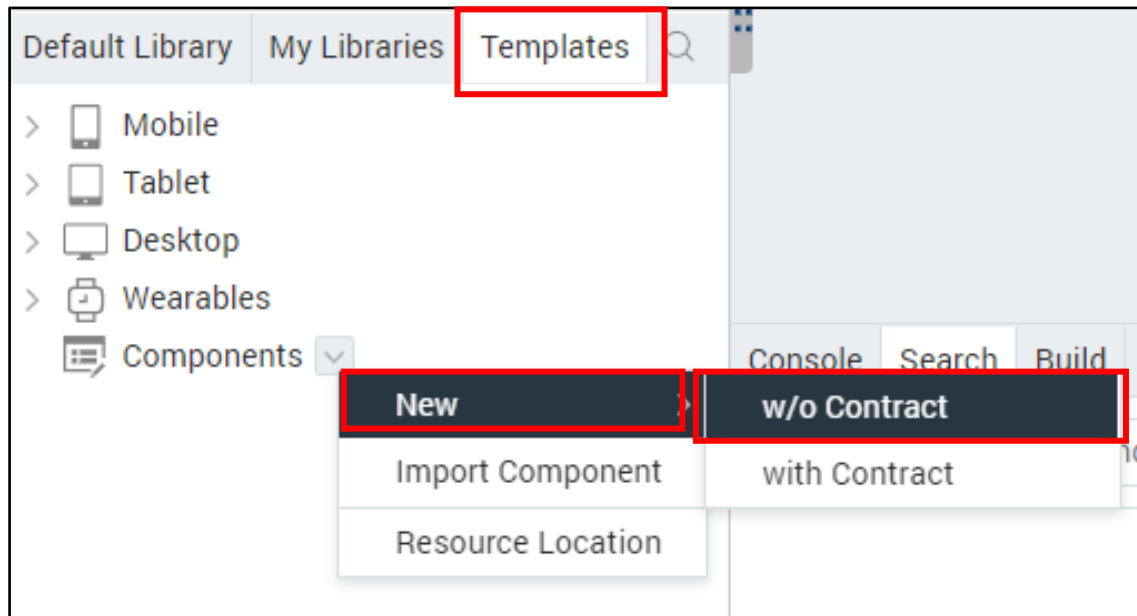
- コンポーネントには、以下の2種類があります。
  - コントラクトのないコンポーネント
    - 自己充足的で再利用可能なUIエンティティ
  - コントラクトを持つコンポーネント
    - このタイプのコンポーネントは、コンシューマにコントラクトを公開することで、内部の詳細を抽象化します。
    - コンポーネントの消費者は、コントラクトを利用してコンポーネントのUIや振る舞いをカスタマイズできます。

# コントラクトなしのコンポーネント - 特長

- コントラクトを持たないコンポーネントは、以下のような特徴を持ちます。
  - 独自の名前空間とクラス名
  - UIはビューファイルで、ビジネスロジックはコントローラファイルで、ビジネスロジックはコントローラファイルに記述
- UI階層全体とビジネスロジックは、親コンテナからアクセス可能です。
  - `this.view.parent` は、コントローラ内の親ビューへのアクセスを提供しませんが、親コントローラの `widget.parent` は、その親ビューを提供します。
- コントラクトを持たないコンポーネントは、以下のことが可能です。
  - あるプロジェクトからエクスポートして、別のプロジェクトにインポートできます。
  - リファレンスアーキテクチャとフリーフォームタイプの両方のプロジェクトで使用されます。

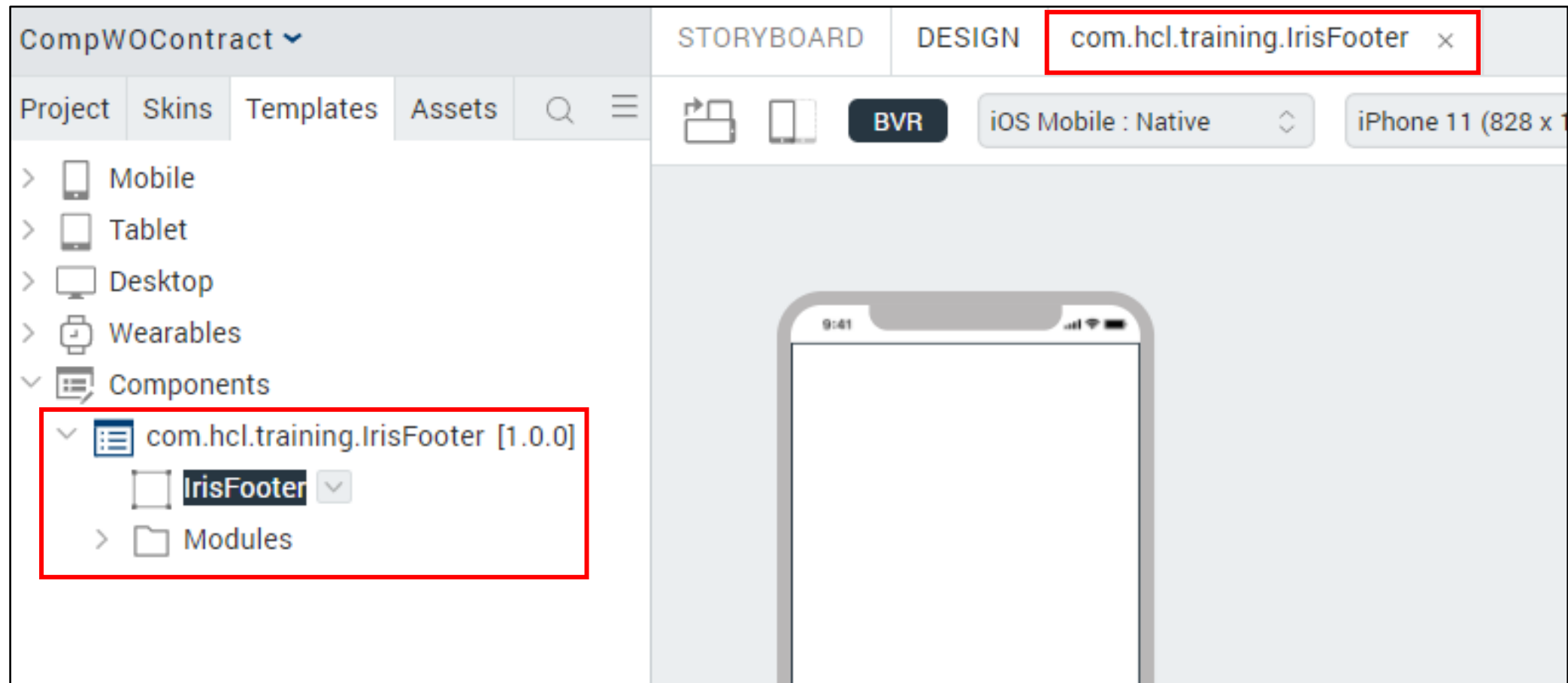
# ユースケースの実装

- アプリケーションで使用するフッターを、コントラクトのないComponentを作成して作成します。
- フッターには、標準的なボタンセットを配置し、ナビゲーションアクションを適用します。
- 各フォームのボタンを制御できるので、”自分自身に戻る”ことはありません。



# コンポーネントの作成

- 以下は、コンポーネントがどのように作成されるかを示しています。
  - デフォルトでは、コンポーネントを作成すると、フレックスコンテナ、コントローラファイル、およびコントローラアクションファイルが作成されます。



# コンポーネントUIの作成

- IrisFooterフレックスコンテナの更新
  - 寸法はHeightを'10%」、Widthを'100%」に設定します。
  - 位置は、Leftを'0dp」、Bottomを'0dp」とします。
  - 注意:Bottomを更新した後、Topの値を削除してください。
- フレックスコンテナ 'flexfooter' のレイアウトタイプを 'Flow Horizontal' に設定します。

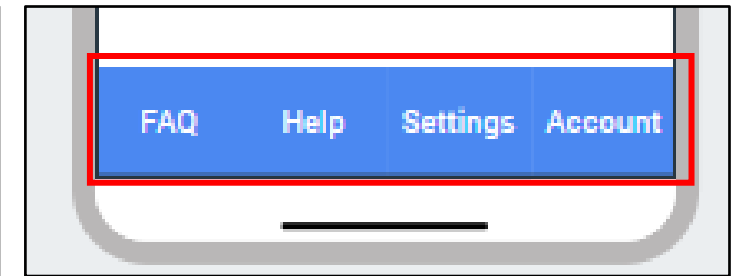
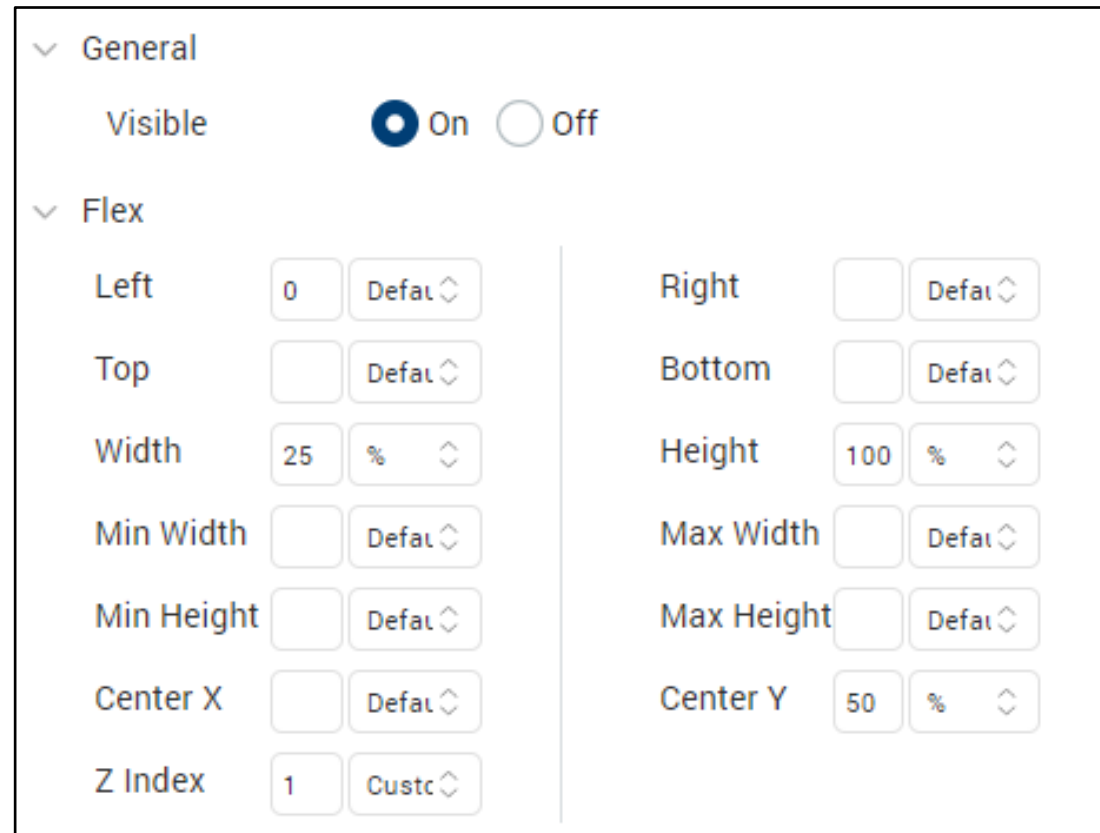
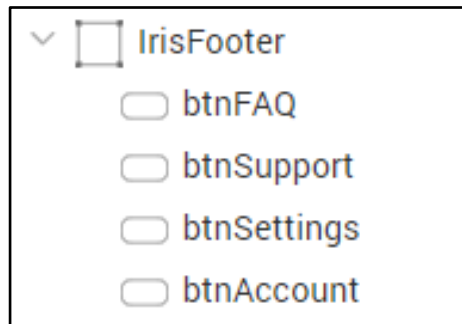
The screenshot displays the configuration interface for the 'IrisFooter' component. The 'General' tab is active, showing the component's ID as 'IrisFooter' and its visibility status as 'On'. The 'Render' button is visible. The 'Flex' section is highlighted with a red box, showing the following settings:

Property	Value	Unit
Left	0	Dp
Top	Default	
Width	100	%
Right	Default	
Bottom	0	Dp
Height	10	%
Min Width	Default	
Max Width	Default	
Min Height	Default	
Max Height	Default	
Center X	Default	
Center Y	Default	
Z Index	1	Custc

Below the 'Flex' section, the 'Look' tab is selected, showing the 'FlexContainer' settings. The 'General' sub-tab is active, displaying 'Clip Bounds' as 'On' and 'Layout Type' as 'Free Form'.

# コンポーネントUIの作成

- 4つのボタンを追加します。
  - 各ボタンの位置は、'Left' を '0dp'、'CenterY' を '50%' とします。
  - 各ボタンの寸法は、'Height' が '100%'、'Width' が '25%' とします。



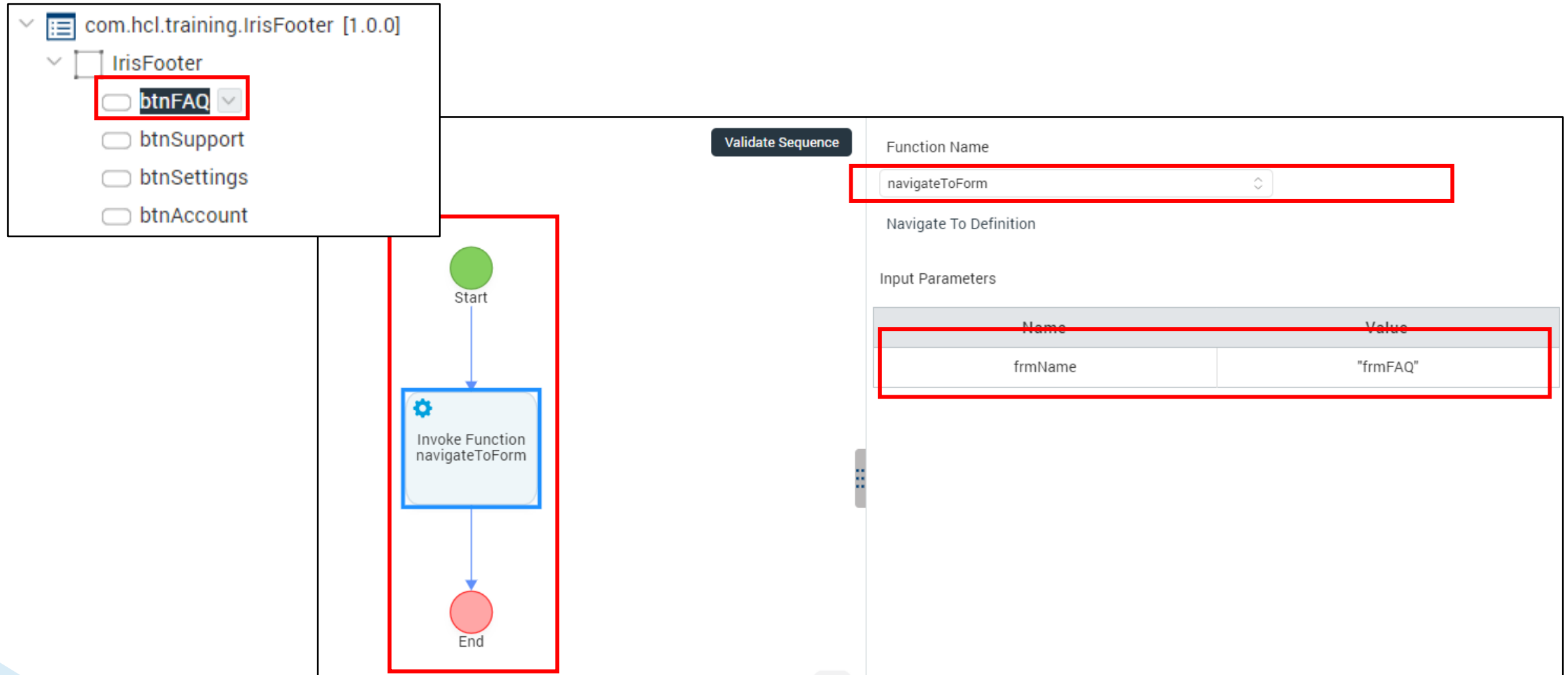
# コンポーネントロジックの実装

- コンポーネントロジックの実装
  - コンポーネントに関連するロジックは、controllers JavaScript ファイルに実装する必要があります。
  - controller と controllerActions ファイルは RequireJS 形式です。
  - controllerActions には、コンポーネント内のウィジェットに関連するアクション シーケンス用の生成コードが含まれます。
- 実装するロジックは次のとおりです。
  - 各ボタンをクリックすると、適切なフォームに移動します。
  - これをサポートするメソッドを作成します。
- コントローラファイルを更新し、以下を追加します（そして、必ずファイルを保存してください）。

```
define(function() {  
    return {  
        navigateToForm(frmName) {  
            var navigationObject = new voltmx.mvc.Navigation(frmName);  
            navigationObject.navigate();  
        }  
    };  
});
```

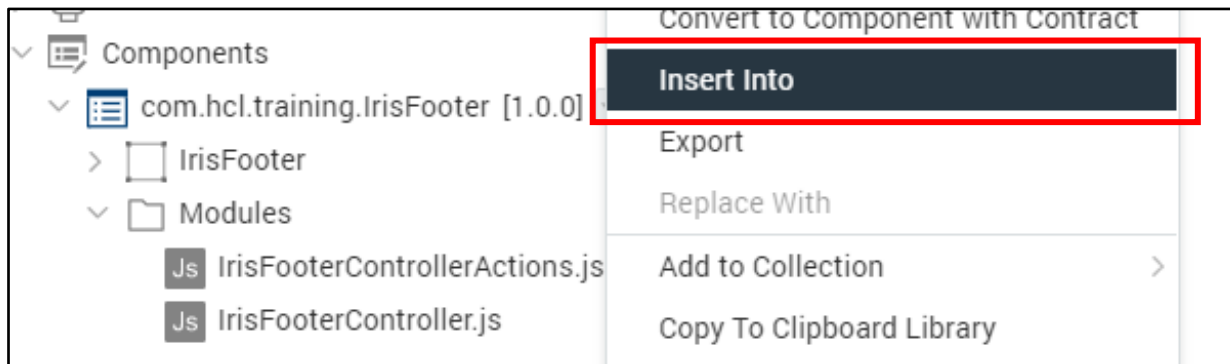
# コンポーネントロジックの実装（続き）

- btnFAQのコードを追加して、フッターの他のウィジェットでも同じことを繰り返します。



# フォームでのコンポーネントの使用

- フォームでのコンポーネントの使用
  - フォームを開きます。
  - Template タブに移動します。
  - コンポーネントを右クリックし（またはコンテキストメニューから）、"insert into" オプションを選択します。
  - コンポーネントがフォームに追加されます。
  - コンポーネントをフォームにドラッグすることもできます。
- 以下は、FAQフォームにフッターを追加した例です。
  - 他のフォームでコンポーネントを使用するには、以下の手順に従います。



# フォームでのコンポーネントの使用（つづき）

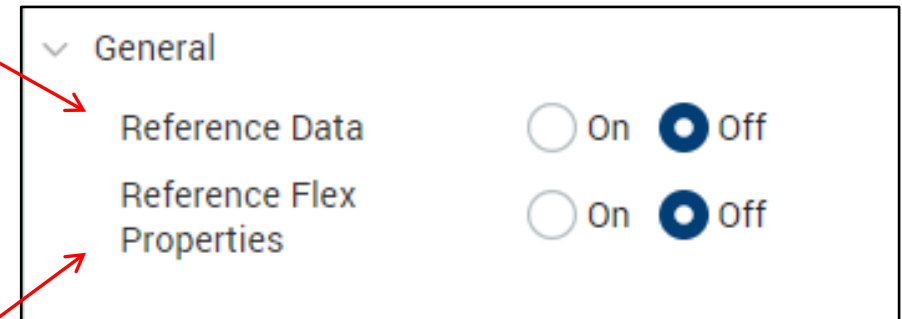
- フォーム内のコントローラのFlex Containerタブを選択すると、ウィジェットのデータを編集したり、ウィジェットの位置を変更するためのオプションが表示されます。

## リファレンスデータ

- Off(デフォルト): フォームのデータを編集できますが、グローバルには表示されません。
- On: データを変更することはできません

## 参照先 Flex プロパティ

- Off (デフォルト): Lookタブでコンポーネントの場合、ポジションの編集が可能です
- On: 位置を変更することはできません



General

Reference Data ☐ On ☒ Off

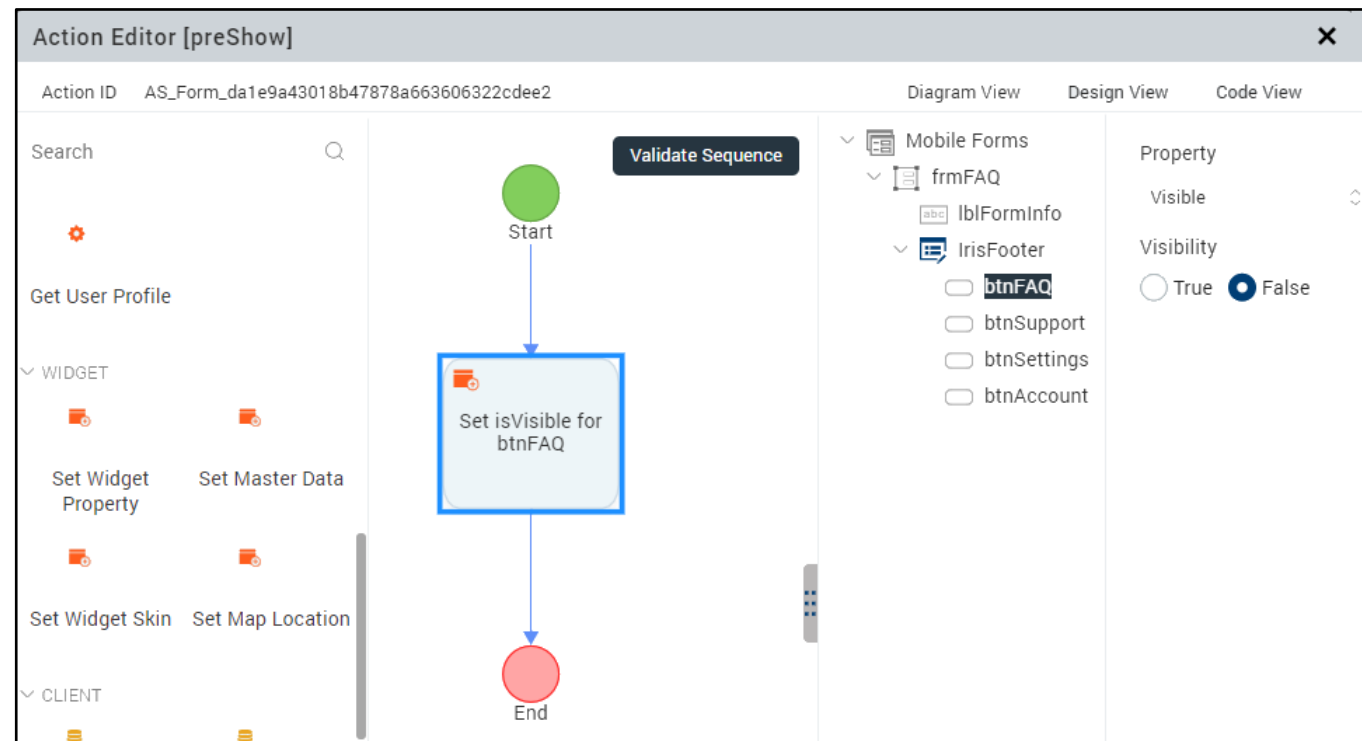
Reference Flex Properties ☐ On ☒ Off

Red arrows from the text boxes point to the 'Reference Data' and 'Reference Flex Properties' labels.

# フォーム内のコンポーネントの編集

- コンポーネントのすべてが、その親から直接利用できます。
- コンポーネントのUIは、そのまま使用することも、アプリの要件に合わせてカスタマイズすることもできます。
- コンポーネントをフォームに追加した後、フォームのコントローラから機能を追加できます。
- 新しく追加した関数は”this.view.functionName()” として呼び出せます。

- また、アクションエディタを使用してプロパティを編集することもできます
- 以下は、ボタンを非表示にする例です。



# 練習

- このレッスンで説明した手順に従ってください。
  - コントラクトのないコンポーネントを作成する。
  - そのコンポーネントをフォームで使用する。
  - フォーム上で使用されるコンポーネントをカスタマイズする。

***HCL***

[www.hcltech.com](http://www.hcltech.com)

\$10 BILLION | 159,000+ IDEAPRENEURS | 50 COUNTRIES