



HCL Volt MX

コードでの Segment ウィジェット データの操作

Student Guide



HCL Software Academy for HCL Digital Solutions

Creating a new generation of experts

もくじ

コードでの Segment ウィジェットデータの操作	3
前提条件	3
コードを使って Segment ウィジェットにデータを設定する／Segment ウィジェットからデータを取得する	3

コードでの Segment ウィジェットデータの操作

このレッスンでは、セグメントウィジェットに設定できるデータ構造、セグメントウィジェット API を紹介します。

このドキュメントでは、このレッスンのハンズオン部分の詳細な手順について説明します。

前提条件

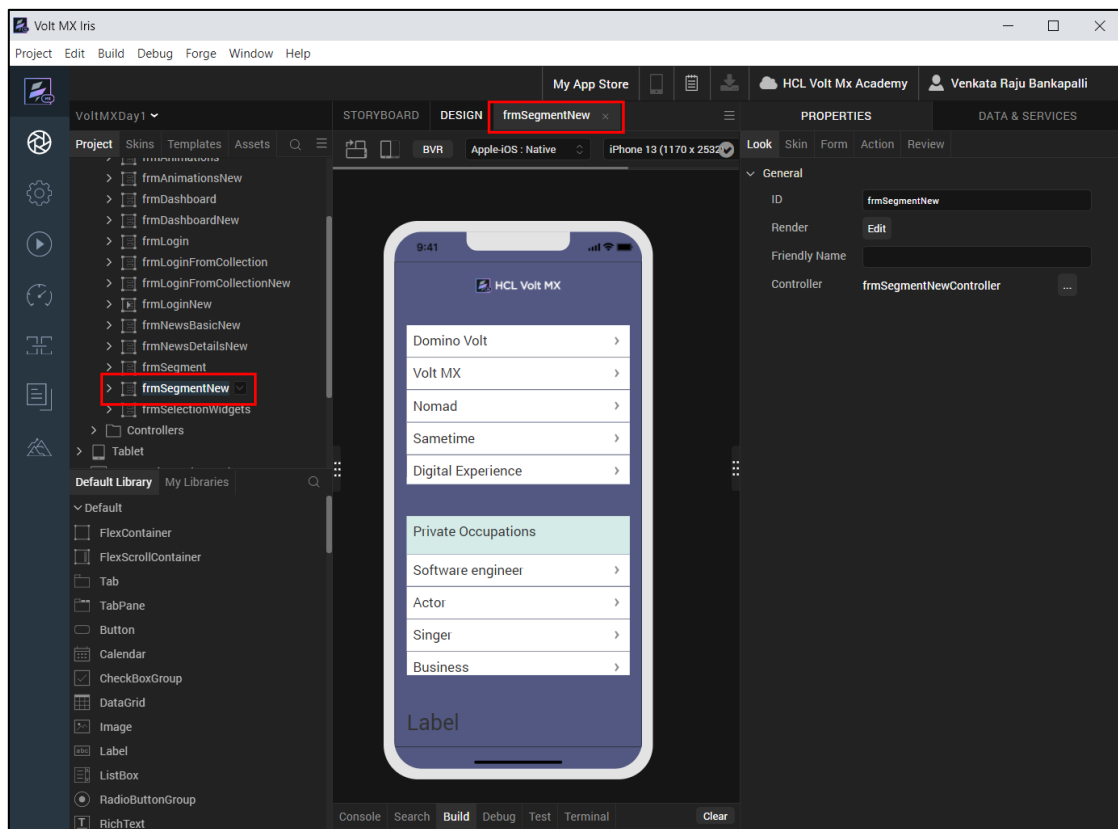
- 前のレッスンのハンズオンの手順を完了していること。

コードを使って Segment ウィジェットにデータを設定する／ Segment ウィジェットからデータを取得する

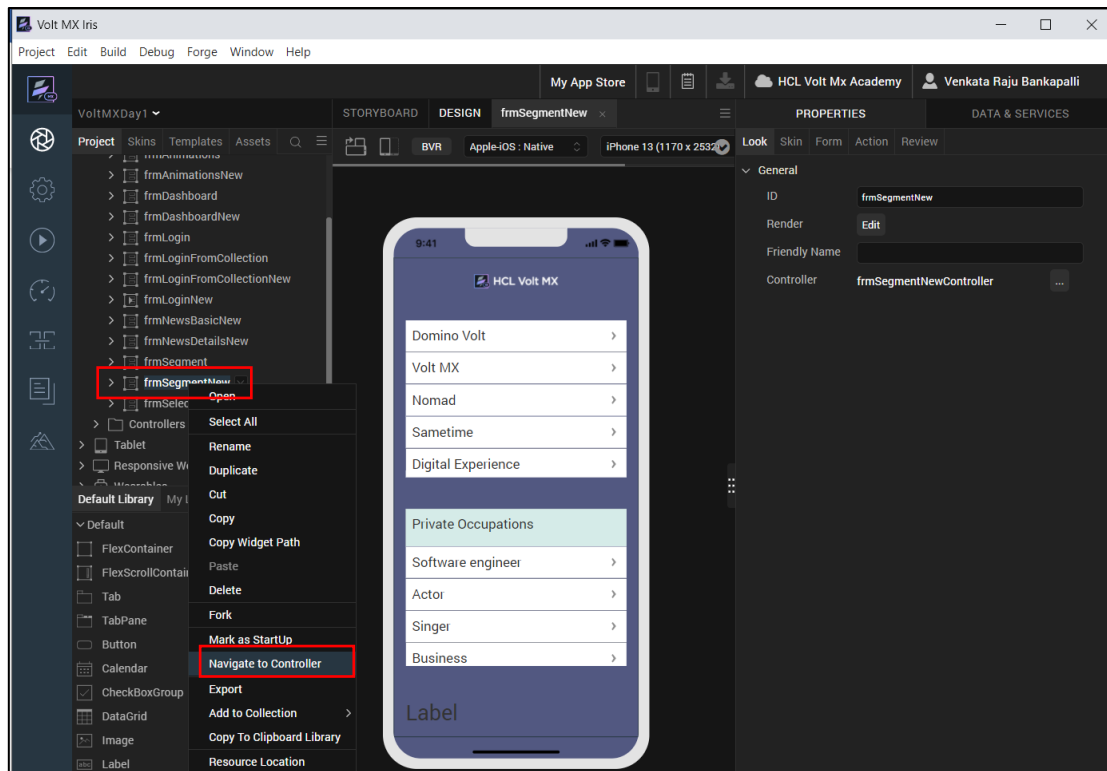
注意事項

- 前のレッスンで作業した Iris プロジェクトで、以下のステップを引き続き実行します。

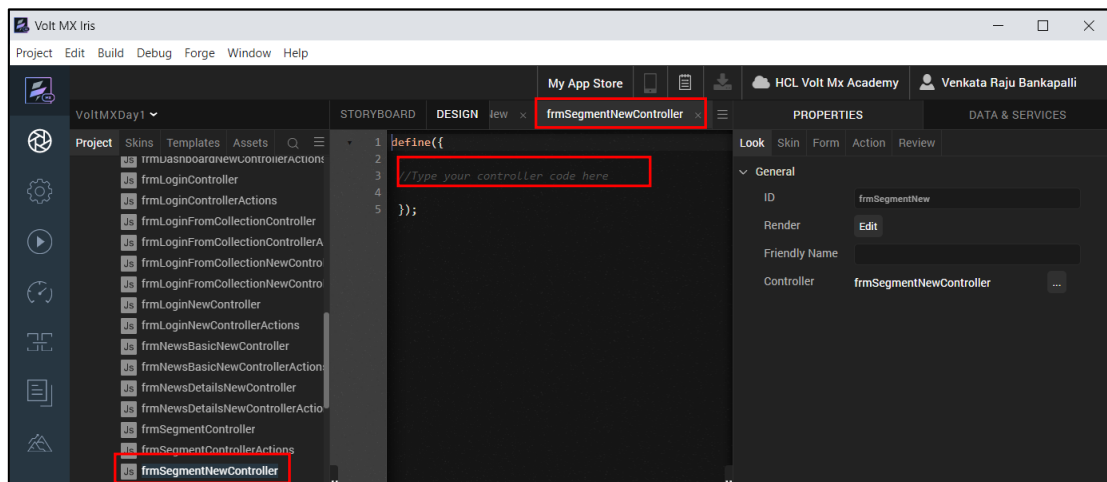
- フォーム **frmSegmentNew** を開きます。



- アウトラインでフォーム **frmSegment** を右クリックし、**Navigate to Controller** オプションをクリックします。このフォームに関連するフォームコントローラが開きます。



- フォーム上の Segment ウィジェット `segBasic` と `segWithTemplates` にデータを設定するために、以下の 2 つの関数を追加します。以下のコードを **// Type your controller code here** にペーストします。



```
/**
 * この関数は、セグメント segBasic にデータを設定する役割を担います。
 * この関数では、Segment ウィジェットの widgetDataMap と setData API を使用しています。
 */
setDataToSegBasic : function () {
    voltmx.print("Entering into setDataToSegBasic");

    var data = [
        {"productName":"Data set through code"},
        {"productName":"Domino Volt"},
    ]
}
```

```

        {"productName":"Volt MX"},
        {"productName":"Nomad"},
        {"productName":"Sametime"},
        {"productName":"Digital Experience"},
        {"productName":"BigFix"},
        {"productName":"Commerce"},
        {"productName":"Unica"},
        {"productName":"Notes & Domino"}
    ];

    this.view.segBasic.widgetDataMap = {"lblProductName":"productName"};
    this.view.segBasic.setData(data);

    voltmx.print("Exiting out of setDataToSegBasic");
},

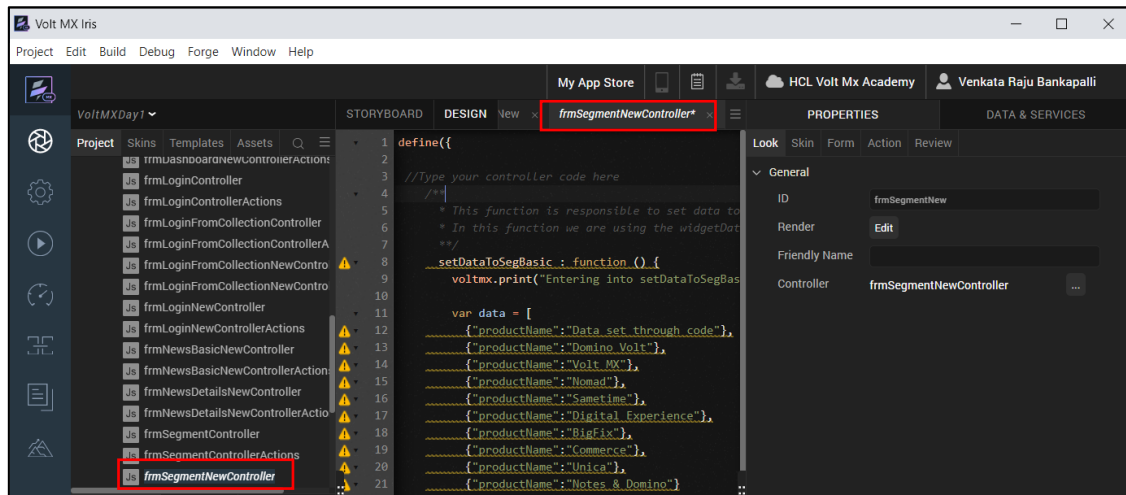
/**
 * この関数は、segWithTemplates にデータを設定します。
 * この関数では、データをセクションで表示するために、どのようにデータを構造化するか、
tempaltes を使って調べます。
 * この関数では、Segment ウィジェットの widgetDataMap と setData API を使用します。
 */
setDataToSegWithTemplates : function () {
    voltmx.print("Entering into setDataToSegWithTemplates");

    var data = [
        [
            {"template":"flxSegSectionHeader", "occupationType":"Data set through code"},
            [
                {"template":"flxSegRow", "occupation":"Data set through code"},
                {"template":"flxSegRow", "occupation":"Actor"},
                {"template":"flxSegRow", "occupation":"Singer"},
                {"template":"flxSegRow", "occupation":"Business"}
            ]
        ],
        [
            {"template":"flxSegSectionHeader", "occupationType":"Public Occupations"},
            [
                {"template":"flxSegRow", "occupation":"Farmer"},
                {"template":"flxSegRow", "occupation":"Doctor"},
                {"template":"flxSegRow", "occupation":"Police"},
                {"template":"flxSegRow", "occupation":"Army"}
            ]
        ]
    ];

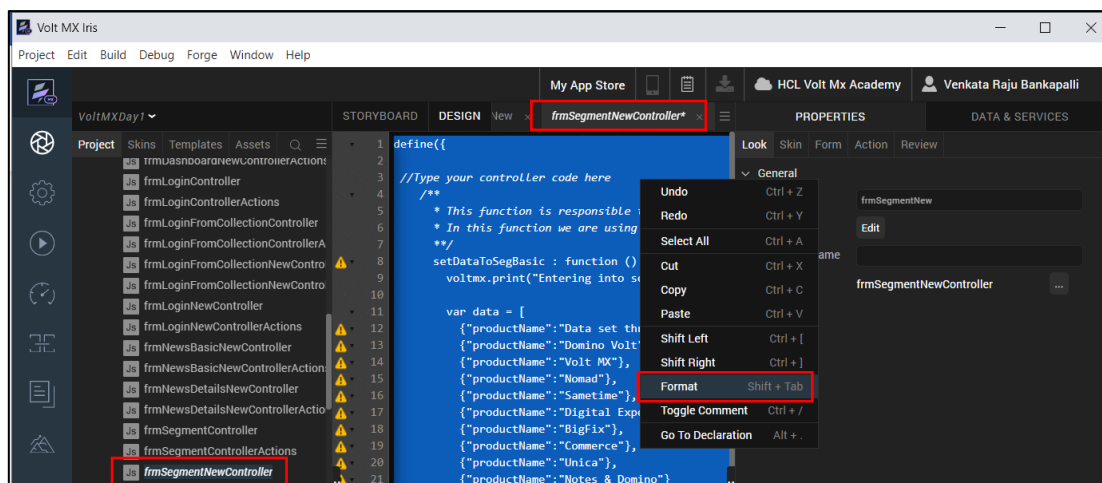
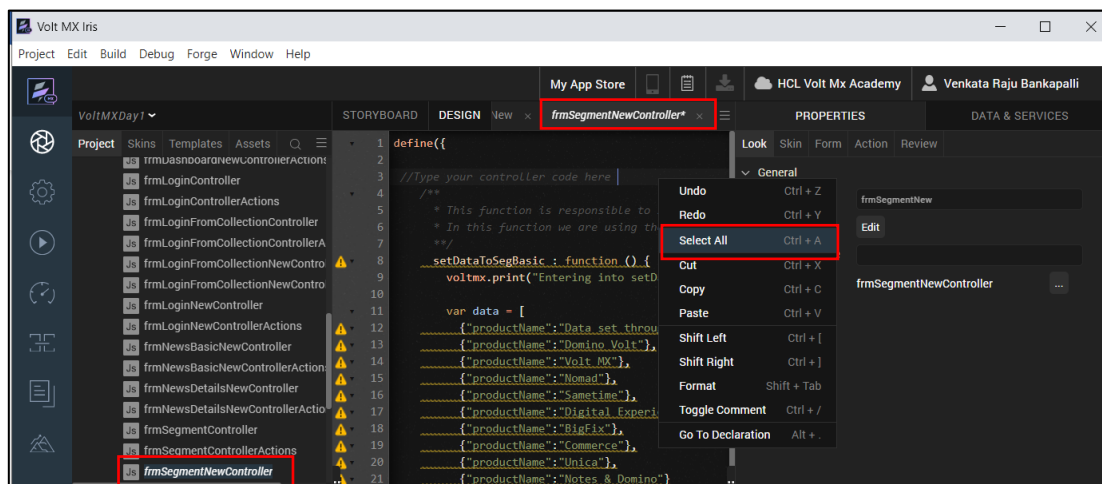
    this.view.segWithTemplates.widgetDataMap = {"lblSegRow":"occupation",
    "lblSegHeader":"occupationType"};
    this.view.segWithTemplates.setData(data);

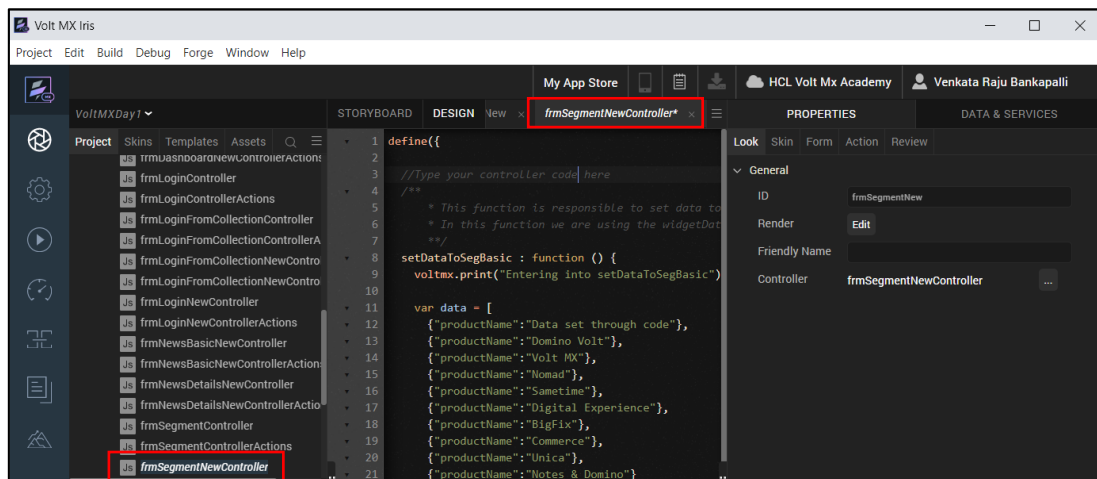
    voltmx.print("Exiting out of setDataToSegWithTemplates");
},

```

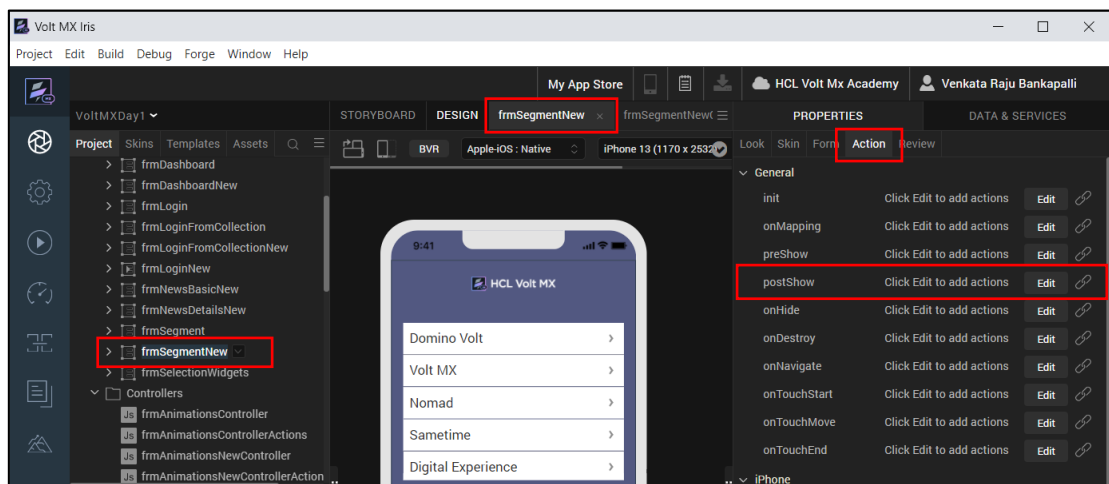


- コードを整形するために、コードエディタ上で右クリックし、**Select All** を選択し、再度右クリックで **Format** をクリックします。

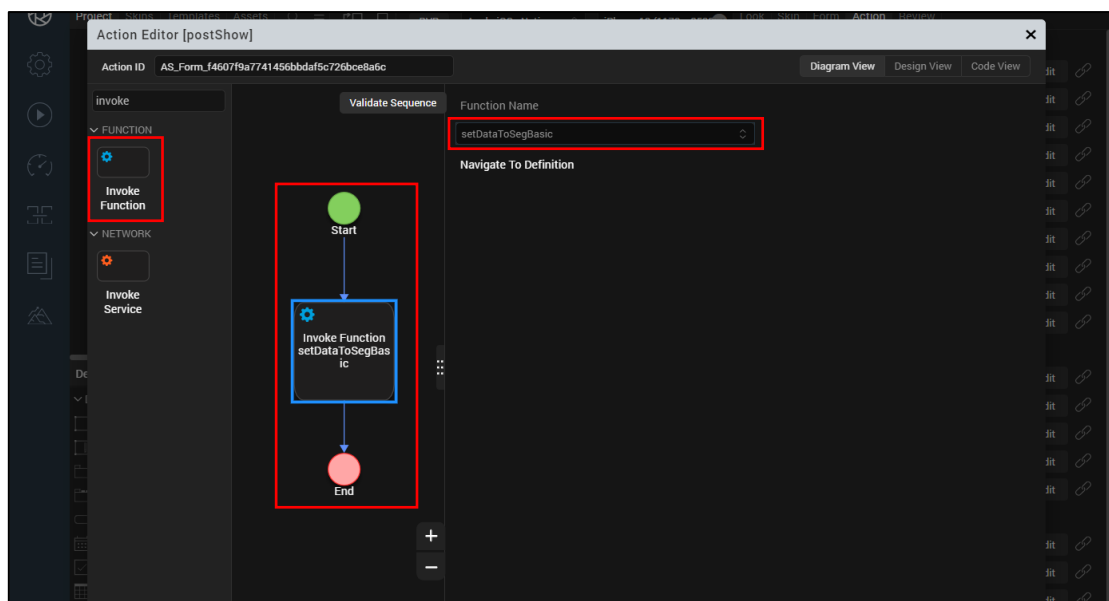




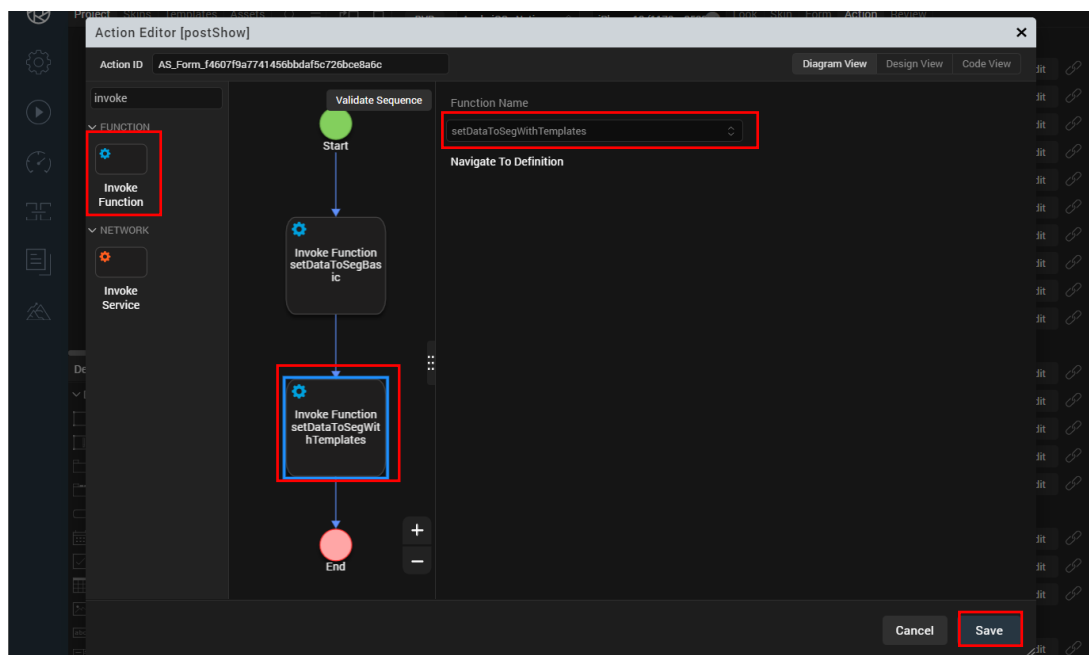
- **frmSegmentNew** のプロパティの **Action** タブで、**postShow > Edit** でこのフォームに postShow イベントを設定します。



- Action Editor で、**Invoke Function** アクションを検索し、このアクションを選択します。
- アクションシーケンスに追加されたアクションを確認したら、右側のドロップダウンから、関数 **setDataToSegBasic** を選択します。



- 再度、Action Editor で **Invoke Function** アクションを検索し、このアクションを選択します。
- アクションシーケンスに追加されたアクションを確認したら、右側のドロップダウンから **setDataToSegWithTemplates** という関数を選択します。



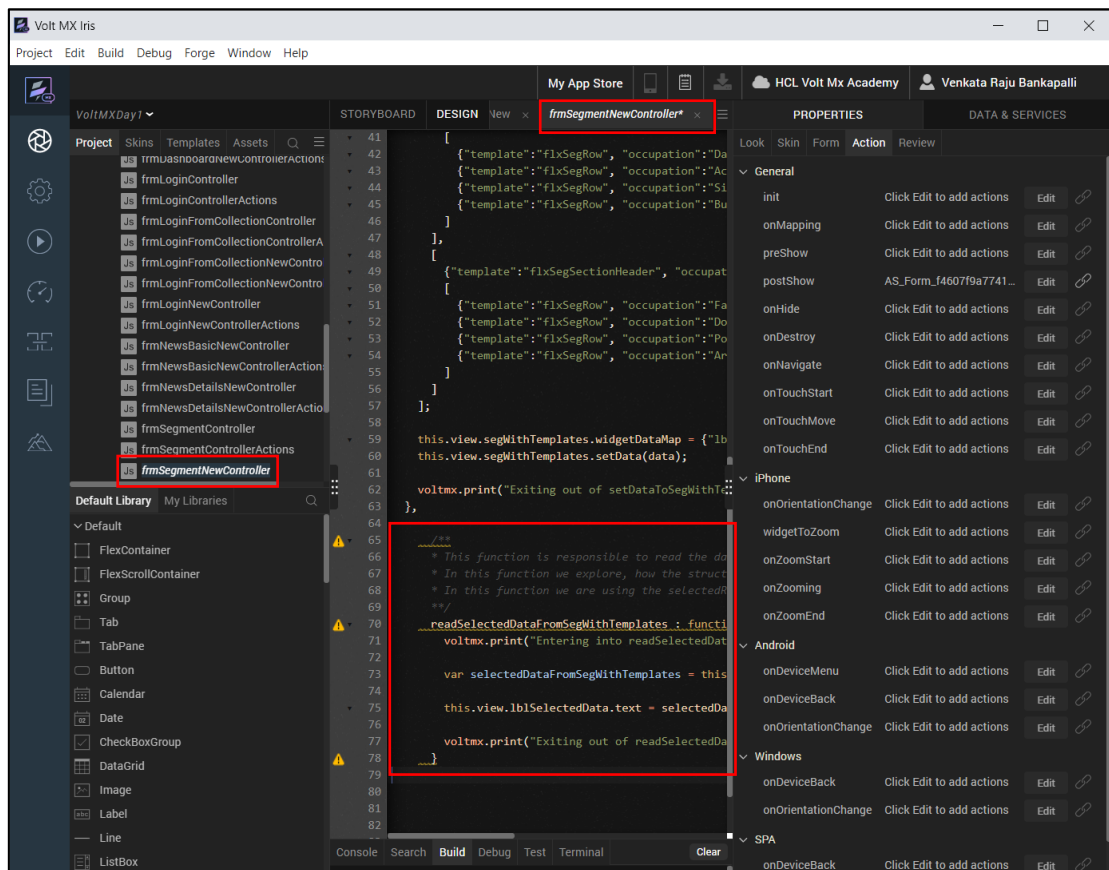
- Save** をクリックして、アクションシーケンスを保存します。
- フォーム **frmSegmentNew** を開きます。
- アウトラインのフォーム **frmSegmentNew** を右クリックし、**Navigate to Controller** オプションをクリックします。このフォームに関連するフォームコントローラが開かれます。
- フォーム上の Segment ウィジェット **segWithTemplates** から選択されたデータを読み込むために、以下の関数を追加します。ペースト位置に注意してください (下図参照)。

```
/**
 * この関数は、Segment ウィジェットの readSelectedDataFromSegWithTemplates から選択されたデータ
 * を読み込む役割を担います。
 * この関数では、セグメントから選択されたデータがどのような構造になっているかを調べます。
 * この関数では、Segment ウィジェットの selectedRowItems API を使用しています。
 */
readSelectedDataFromSegWithTemplates : function () {
    voltmx.print("Entering into readSelectedDataFromSegWithTemplates");

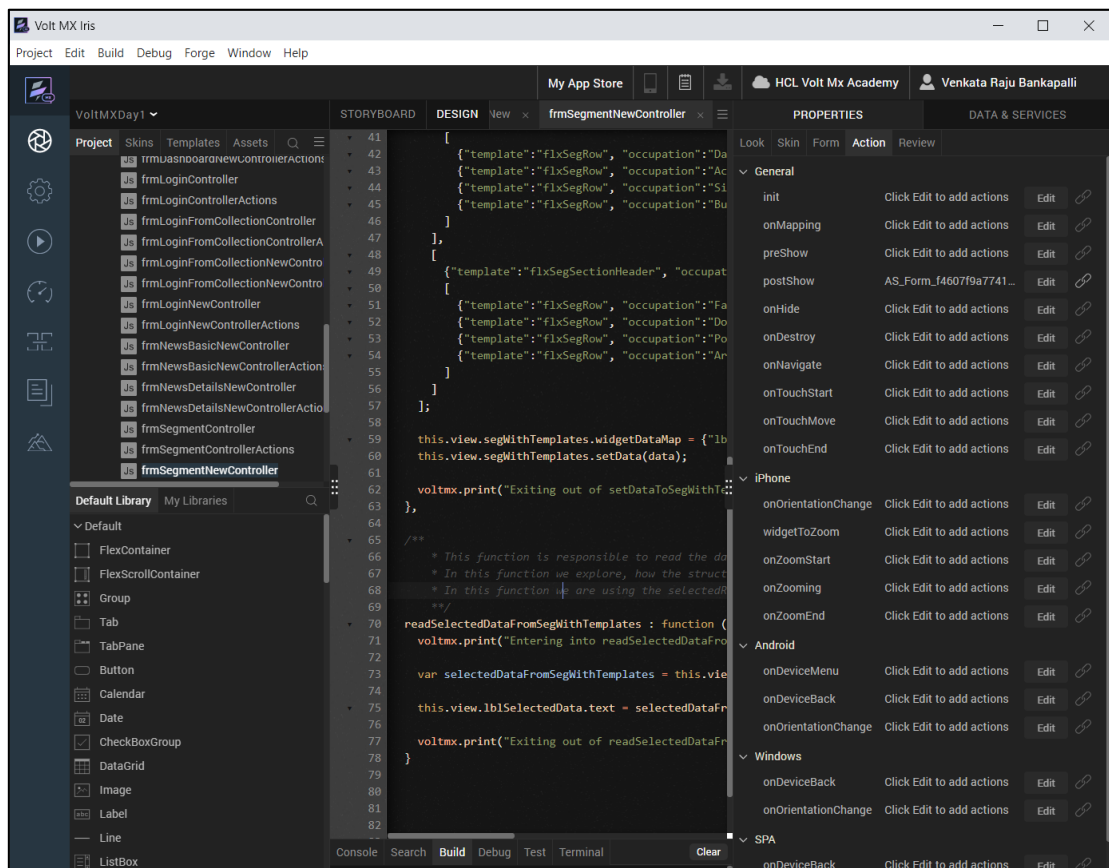
    var selectedDataFromSegWithTemplates = this.view.segWithTemplates.selectedRowItems;

    this.view.lblSelectedData.text = selectedDataFromSegWithTemplates[0]['occupation'];

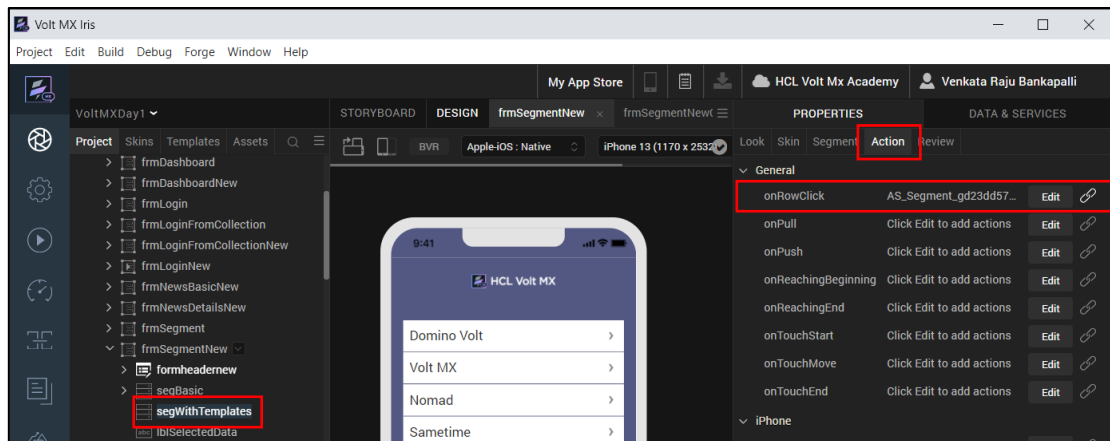
    voltmx.print("Exiting out of readSelectedDataFromSegWithTemplates");
}
```

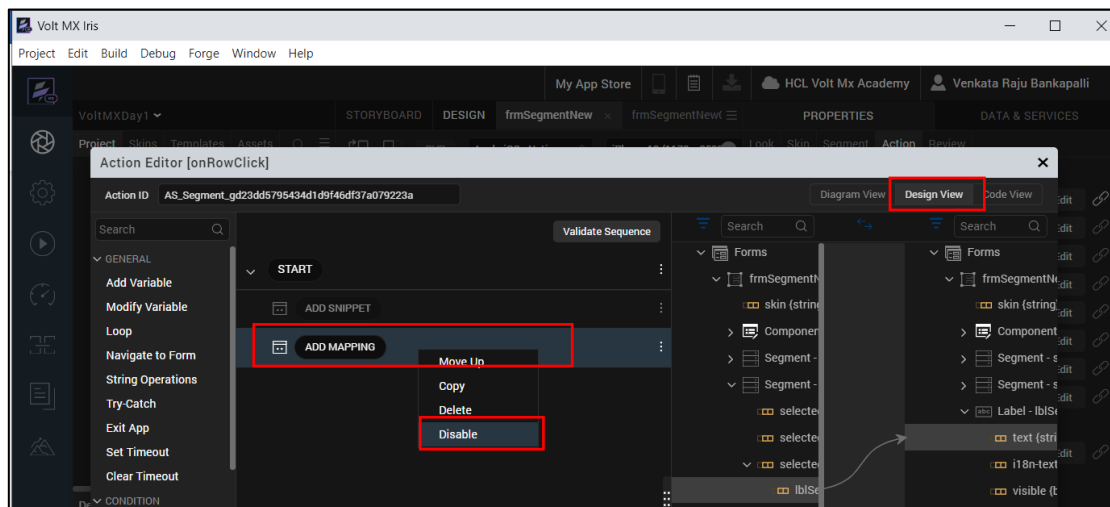
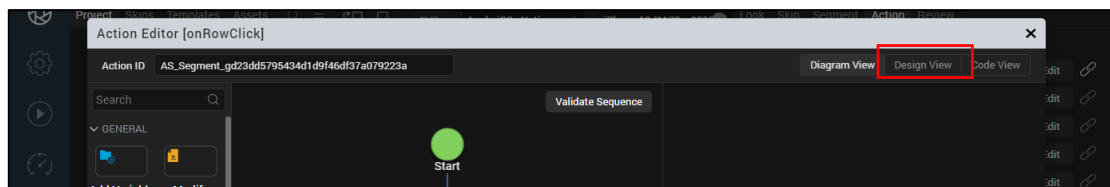
- コードを整形するために、コードエディタ上で右クリックし、**Select All** を選択し、再度右クリックで **Format** をクリックします。フォーマット後の状態は以下のとおりです。



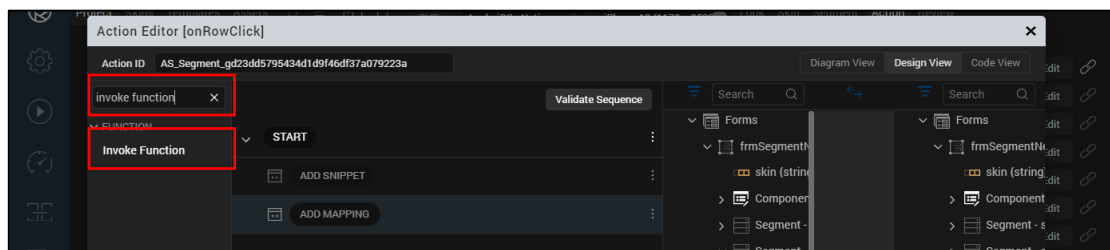
- Properties > **Action** > **onRowClick** > **Edit** で、onRowClick イベントを **segWithTemplates Segment** ウィジェットに設定します。



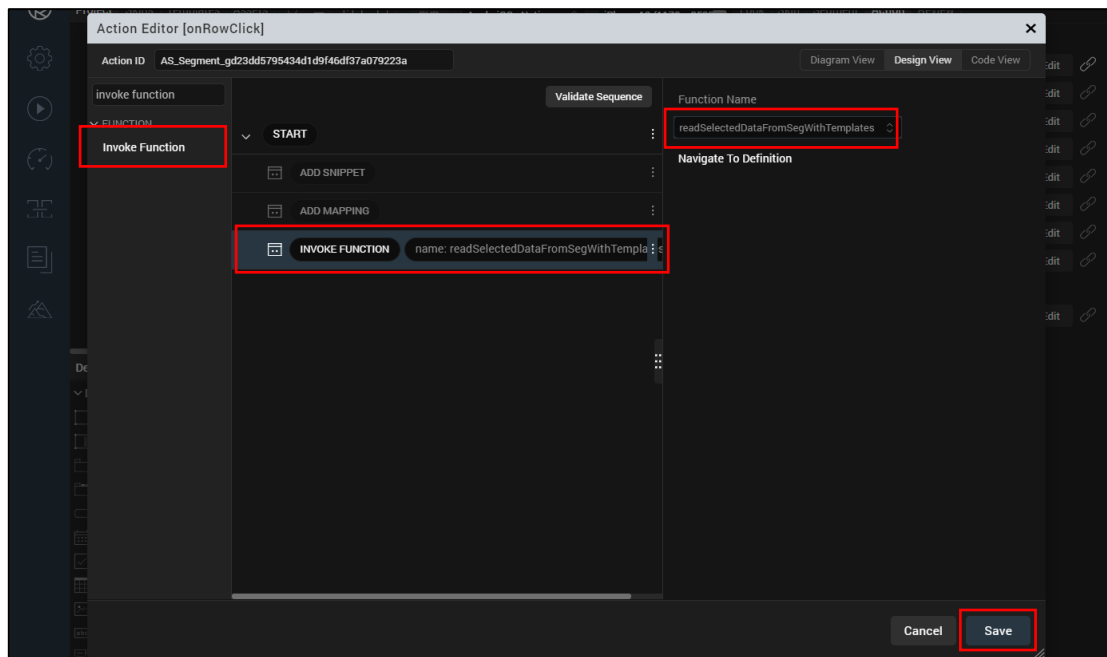
- Action Editor で、**Design View** に切り替え、設定されているアクションを無効化します。



- Invoke Function** アクションを検索し、このアクションを選択します。



- アクションシーケンスに追加されたアクションを確認したら、右側のドロップダウンから **readSelectedDataFromSegWithTemplates** という関数を選択します。**Save** をクリックし、アクションシーケンスを保存します。



- Build > Run Live Preview** メニューでライブプレビュービルドを生成し、コードによるセグメントへのデータ設定と、コードによるセグメントからの選択データの読み込みをテストします。

注意

- ライブプレビューの設定で以前選択したチャンネル/プラットフォーム/アプリケーションの種類はそのまま残ります。また、ライブプレビュービルドは、これらの選択されたチャンネル/プラットフォーム/アプリケーションの種類で生成されます。
- アダプティブウェブのライブプレビュービルドは、お使いのマシンで開く **Volt MX** アプリのプレビューを使用してテストします。
- Wi-fi モードの **Volt MX** アプリプレビューアプリを使用して、IP アドレスを入力するか、バーコードをスキャンして、ネイティブライブプレビュービルドをテストします。
- スマホが接続されているネットワークが、**Volt MX Iris** が動作しているマシンのネットワークと通信できる必要があることに注意してください。

おめでとうございます。このレッスンのハンズオンは終了です。