5 Nov 2021

Checklist

Version: 1.0.1

# **OVERVIEW**

The Checklist, also known as the Questionnaire Sample, is a very common component where a user needs a checklist to verify pre-service/ post-service process compliance. A simple list of checkboxes can go a long way in determining the streamlined execution of any request.

## **Use case:**

### Consider a case that you are developing a Journey Management application that allows you to have user details such as name, mobile no, destination. Using the Checklist (Questionnaire Sample) component, you can display different types of question such as Checkbox, Radio button and text type questions. User can choose accordingly.

### You can import the Questionnaire Sample component into your app and make use of its features. The Questionnaire Sample component also facilitates a set of properties and events that helps you customize the fields and the functionality.

## **Features:**

## Prompts user with check box, radio button and text box questions.

## Can add any number of questions.

## Customize the component with your own questions.

## **Percentage of re-use:**

Approximately 80% of reuse.

# **Getting Started**

## **Prerequisites**

Before you start using the Questionnaire Sample component, ensure the following:

• [HCL Foundry](https://manage.hclvoltmx.com/)

• Volt MX Iris

## **Platforms Supported**

### Mobile

#### iOS

#### Android

### Tablet & iPad

### PWA & Responsive Web

## **Importing the App**

## You can import the Forge components only into the apps that are of the Reference Architecture type.

## **To import the Questionnaire Sample component, do the following:**

## Open your app project in Volt MX Iris.

* 1. In the Project Explorer, click the **Templates** tab.

Graphical user interface, text, application

Description automatically generated

## Right-click **Components**, and then select **Import Component**. The **Import Component** dialog box appears.

## Graphical user interface, text, application, Teams Description automatically generated

## Click **Browse** to navigate to the location of the component, select the component, and then click **Import**. The component and its associated widgets and modules are added to your project.

## A screenshot of a computer Description automatically generated

Once you have imported a component to your project, you can easily add the component to a form. For more information, refer [Add a Component to a Form](https://opensource.hcltechsw.com/volt-mx-docs/docs/documentation/Iris/iris_user_guide/Content/C_UsingComponents.html" \l "add-a-component-to-a-form)

## **Building and previewing the app**

## After performing all the above steps, you can build your app and run it on your device. For more information, you can refer to the [Building and Viewing an Application](https://opensource.hcltechsw.com/volt-mx-docs/docs/documentation/Iris/iris_user_guide/Content/Cloud_Build_in_VoltMX_Iris.html#cloud) section of the Volt MX User Guide.

You can then run your app to see the Questionnaire Sample-FS work in real time.

# **REFERENCES**

## **Dynamic Usage**

You can also add **Questionnaire Sample** component dynamically. To do so,

* + - 1. In Project **Explorer**, on the **Projects** tab, click **Controllers** section to access the respective **Form Controller**. Create a method and implement the code snippet similar to the sample code mentioned below.



/\* Creating Questionnaire Sample component instance \*/

createComponent:function(){

var questionaireInstance = new com.voltmx.questionaire ({

"clipBounds": true,

"height": "100%",

"id": "questionaire",

"isVisible": true,

"left": "0dp",

"top": "0dp",

"width": "98%",

"centerX":"50%",

"zIndex":"1"

}, {}, {

});

questionaireInstance.skinHeaderLabel="sknLblPleaseConfirm";

questionaireInstance.rowSkin="sknSegmentQuestions";

questionaireInstance.textHeaderLabel="Please Confirm";

this.view.add(questionaireInstance);

}

In the code snippet, you can edit the properties of the component as per your requirement. For more information, see Setting Properties.

* + - 1. Save the file.

## **Properties**

## The properties provided on the **Component** tab allows you to customize the elements in the **Questionnaire Sample** component. These elements can be UI elements, service parameters, and so on. You can set the properties from the Volt MX Iris Properties panel on the right-hand side. You can also configure these properties using a JavaScript code.

**General Properties**

**1. Header Label Text**

|  |  |
| --- | --- |
| **Description:** | Specifies the text that you want to display on the header label. |
| **Syntax**: | text |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.questionaireInstance.textHeaderLabel="Please Confirm"; |
| **Remarks:** | The default value for the property is "Please Confirm";. |

**Skins Section**

**2. Header Label Skin**

|  |  |
| --- | --- |
| **Description:** | Specifies the skin of the Header label. |
| **Syntax:** | SkinHeaderLabel |
| **Example:** | this.view.questionaireInstance.skinHeaderLabel="sknLblPleaseConfirm"; |
| **Remarks:** | The default skin for the property is "sknLblPleaseConfirm". |

**3. Segment Skin**

|  |  |
| --- | --- |
| **Description:** | Specifies the skin of a row in segment. |
| **Syntax**: | rowSkin |
| **Example:** | this.view.questionaireInstance.rowSkin="sknSegmentQuestions”; |
| **Remarks:** | The default skin for the property is "sknSegmentQuestions ". |

## **Events**

## There is no event in this component.

## **APIs**

### **setQuestions**

|  |  |
| --- | --- |
| **Description:** | It can be used to set custom data from controller/ Services. |
| **Syntax**: | setQuestions |
| **Parameters:** | JSON array. |
| **Return Value:** | No return value. |
| **Example:** | this.view.questionaire.setQuestions(); |

# **Revision History**

App version 1.0.1:

## **Limitations:**

## No Limitations.

## **Known Issues:**

## Lines are highlighted when we select the row segment in Safari which is the default behavior.