Date : 29/9/2021

 Carousel (1.0.1)

1. Overview For Carousel

Carousel is a front-end component that displays a list of images in the form of a scrollable catalog. The component supports automatic scrolling and infinite scrolling.

You can customize the size, scrolling direction, animation speed, and the delay for automatic scrolling. You can use the component in scenarios where you want to add a list of images in a fixed space such as an item catalog or an image gallery.

## Use case

The component can be used in scenarios where you want to add a list of images in a fixed space such as an item catalog or an image gallery

## Features

1. The component supports automatic scrolling and infinite scrolling
2. The size, scrolling direction, animation speed, and the delay for automatic scrolling are customizable as required

## Percentage of re-use:

80-90% re-use.

# Getting Started

## Prerequisites

Before you start using the CarouselVue component, ensure the following:

* [HCL Foundry](https://manage.hclvoltmx.com/)
* Volt MX Iris

Note: In this component we can see the Carousel and Vue Carousel Components Carousel Component is for Mobiles & Tablets.
Vue carousel Component is for PWA (Overview Of Vue Carousel)

## Platforms Supported

### Mobile

#### iOS

#### Android

### Tablets

### PWA

## Importing the Component

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

##  **To import the CarouselVue component, do the following:**

## Open your app project in Volt MX Iris.

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



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



4.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.



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%22%20%5Cl%20%22add-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 VoltMx Iris User Guide.

You can then run your app to see the CarouselVue work in real time.

# References

## Dynamic Usage of Carousel

You can also add a **Carousel** component dynamically. To do so,

1. In the **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.

createComponent: function()

 {

 /\* Creating the component's object \*/

 var carousel = new com.voltmx.carousel(

 {

 "clipBounds": true,

 "height": "100%",

 "id": "carousel",

 "isVisible": true,

 "left": "0dp",

 "top": "0dp",

 "width": "100%",

 "zIndex": 1

 }, {}, {});

 /\* Setting the component's properties \*/

 carousel.data =

 {

 "data":

 [

 {"DefaultColumn": "Image1.png"},

 {"DefaultColumn": "Image2.png"},

 {"DefaultColumn": "Image3.png"},

 {"DefaultColumn": "Image4.png"}

 ]

 };

 carousel.flexTop = "10";

 carousel.flexLeft = "30";

 carousel.flexWidth = "40";

 carousel.flexHeight = "70";

 carousel.animationSpeed = "Medium";

 carousel.animationDirection = "Horizontal";

 carousel.autoScroll = true;

 carousel.autoScrollDelay = 1;

 carousel.animateInLoop = false;

 this.view.add(carousel);

 }

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

You can use a component's **Properties** to customize and configure the elements. These elements can be UI elements, service parameters, and so on. For more information about properties, you can refer to the [Components Overview](https://opensource.hcltechsw.com/volt-mx-docs/docs/documentation/Iris/iris_user_guide/Content/C_ComponentsOverview.html) section of the VoltMx Iris User Guide.

You can set the properties from the VoltMx Iris's Properties panel on the right hand side. You can also configure these properties using a JavaScript code.

 General Properties

**1. Data(data)**

|  |  |
| --- | --- |
| **Description:** |  Specifies the list of images that you want to display on the carousel. |
| **Syntax:** |  data |
| **Type:** | * Data Grid
* Array of JSON
 |
| **Read/Write:** |  Write |
| **Remarks:** | * You can provide the images as local files or URLs.
* If you want to use local files, make sure that the files exist in the *resources*\*common* folder.
 |
| **Example:** | var imageData =[ { "MyColumnOne": "Image1.png" }, { "MyColumnOne": "Image2.png" }, { "MyColumnOne": "Image3.png" }, { "MyColumnOne": "Image4.png" }]this.view.componentID.data = imageData; |

**2. Animation Speed(animationSpeed)**

|  |  |
| --- | --- |
| **Description:** | Specifies the speed that you want to set for the scrolling animation. |
| **Syntax:** |  animationSpeed |
| **Type:** |  1 . List Selector 2 . String |
| **Read/Write:** |  Write |
| **Values:** |   1 . Medium 2 . Slow 3 . Fast |
| **Default Value:** | Medium |

**3. Animation Direction(animationDirection)**

|  |  |
| --- | --- |
| **Description:** | Specifies the direction that you want to set for the scrolling animation. |
| **Syntax:** | animationDirection |
| **Type:** |  1. List Selector 2. String |
| **Read/Write:** | Write |
| **Values:** |  1 .Horizontal2 .Vertical |
| **Default Value:** | Horizontal |
|  **Example:** |   this.view.componentID.animationDirection = "Horizontal"; |
|  |  |

**4. Auto Scroll(autoSroll)**

|  |  |
| --- | --- |
| **Description:** | Enables or disables the automatic scrolling of the carousel. |
| **Syntax:** | autoScroll |
| **Type:** | Boolean |
| **Read/Write:** | Write |
| **Default Value:** | false |
|  |  |

**5. Auto Scroll Delay(autoSrollDelay)**

|  |  |
| --- | --- |
| **Description:** | Specifies the time that you want to set before the carousel automatically scrolls. |
| **Syntax:** | autoScrollDelay |
| **Type:** | Integer |
| **Read/Write:** | Write |
| **Default Value:** | 1 |
| **Remarks:** | The component uses the **Auto Scroll Delay** value only when [Auto Scroll](https://docs.kony.com/marketplace/carousel/Content/Reference.htm#Auto) is enabled. |

**6.Animate In Loop(animateInLoop)**

|  |  |
| --- | --- |
| **Description:** | Enables or disables the infinite scrolling feature of the carousel. |
| Syntax: | animateInLoop |
| **Read/Write:** |  Write  |

|  |  |
| --- | --- |
|  **Default Value:** | false |
|  **Example:** | this.view.componentID.animateInLoop = true; |

**Flex Properties In Percentage**

**1 . Top(flexTop)**

|  |  |
| --- | --- |
| **Description:** | Specifies the position of the top edge of the carousel (in percentage). |
| **Syntax:** | flexTop |
| **Type:** | String |
| **Read/Write:** | Write |
| **Example:** | this.view.componentID.flexTop = "10"; |

**2 . Left(flexLeft)**

|  |  |
| --- | --- |
| **Description:** | Specifies the position of the left edge of the carousel (in percentage). |
| **Syntax:** | flexLeft |
| **Type:** | String |
| **Read/Write:** | Write |
| **Example:** | this.view.componentID.flexLeft = "30"; |

**3 . Width(flexWidth)**

|  |  |
| --- | --- |
| **Description:** | Specifies the width of the carousel (in percentage). |
| **Syntax:** | flexWidth |
| **Type:** | String |
| **Read/Write:** | Write |
| **Example:** | this.view.componentID.flexWidth = "40"; |

**4 . Height(flexHeight)**

|  |  |
| --- | --- |
| **Description:** | Specifies the height of the carousel (in percentage). |
| **Syntax:** | flexHeight |
| **Type:** | String |
| **Read/Write:** | Write |
| **Example:** | this.view.componentID.flexHeight = "70"; |

## **Events**

## **----No events**

|  |  |
| --- | --- |
|   |  |

## APIs

##  --No APIs are exposed.

# Overview Of Vue Carousel

Carousel - vue.js is a catalog of slides that is created by using **vue.js**. Under the hood, the component uses resources from [Vue Carousel 3D](https://github.com/wlada/vue-carousel-3d).

The component contains set of images that can scroll horizontally. Clicking a slide scrolls the carousel in the specified direction and brings the selected slide to the front.
The carousel also supports touch inputs, therefore, on Adaptive Web (Mobile SPA), dragging the images in one direction scrolls the carousel.

You can add custom images on the carousel by using local files or by using URLs. You can also customize the UI of the carousel by using CSS.

You can use the component in scenarios such as: a retail app, where you want to include a catalog of product categories.

# 1.References

## **A. Dynamic Usage of Carousel**

createComponent: function()

{

/\* Creating the component's object \*/

var vueCarousel = new com.voltmx.vuecarousel(

{

 "clipBounds": true,

 "height": "100%",

 "id": "vueCarousel",

 "isVisible": true,

 "left": "0dp",

 "top": "0dp",

 "width": "100%",

 "zIndex": 1

}, {}, {});

/\* Setting the component's properties \*/

vueCarousel.masterData =

{

"data":

[

 {

 "image": "https://i1.wp.com/www.centralofsuccess.com/wp-content/uploads/2019/03/25-Great-Inspirational-Quotes-and-Motivational-Quotes-8.jpg"

 },

 {

 "image": "https://i.pinimg.com/originals/95/cb/ab/95cbab34670fac85105457fc949a70eb.png"

 },

]

};

/\* Adding the component to a form \*/

this.view.add(vueCarousel);

}

To execute the code, you need to call the createComponent function.

**2.Save** the file.

### .

## **B. Properties of Vue carousel**

## 1. Master Data

Description : Specifies the URLs of the images that you want to display on the carousel.

## Syntax: masterData

## Type: 1 . Data Grid

##  2 .JSON

|  |  |
| --- | --- |
| **Read/Write: Write****Example:** |   |
|

|  |  |
| --- | --- |
|  | this.view.componentID.masterData ={"data":[ { "image": "img1.png" }, { "image": "img2.png" },]}; |

 |  |
| **Images for Vue Carousel should either be remote, or imported into Web > LocalFiles > VueCarousel** |  |

## **c. Events Of Vue Carousel**

## **1. onAfterSlideChange**

|  |  |
| --- | --- |
| **Description:** |  Invoked after the slide-change transition ends. |
| **Syntax:** | onAfterSlideChange |
| **Parameters:** | *param [Number]* :The ID of the slide that is selected from the carousel. |
| **Example:** | this.view.componentID.onAfterSlideChange = function(param){ alert("Slide Changed. Slide ID: "+param);}.bind(this); |

## 2 . onBeforeSlideChange

|  |  |
| --- | --- |
| **Description:** | Invoked when the user selects a slide and scrolls the carousel. |
| **Syntax:** | onBeforeSlideChange |
| **Parameters:** | *param [Number]* :The ID of the slide that is selected from the carousel. |
| **Example:** | this.view.componentID.onBeforeSlideChange = function(param){ alert("Changing Slides. Slide ID: "+param);}.bind(this); |

## 3 . onlastSlide

|  |  |
| --- | --- |
| **Description:** | Invoked when the user selects the last slide in the carousel. |
| **Syntax:** | onLastSlide |
| **Parameters:** | *param [Number]* :The ID of the last slide in the carousel. |
| **Example:** | this.view.componentID.onLastSlide = function(param){ alert("Scrolled to last slide. Slide ID: "+param);}.bind(this); |

## 4 . errorCallback

|  |  |
| --- | --- |
| **Description:** | Invoked when an error occurs in the component. |
| **Syntax:** | errorCallback |
| **Parameters:** | *errObj [JSON]* :Contains information about the error, such as the error code and the error message. |
| **Example:** | this.view.componentID.errorCallback = function(errObj){ alert("Error: "+JSON.stringify(errObj));}.bind(this); |

## **Graphical user interface, table  Description automatically generatedD. APIs**

## **1.setData**

**Description:** Displays the specified images on the slides of the carousel.

 **Syntax:** setData(param)

## **Parameters: Param[Array of JSON]**

##  **Contains the URLs of images that you want to diplay on the caurousle**

 **Return Value: None**

**Example**:

|  |
| --- |
| var param =[ { "image": "https://i1.wp.com/www.centralofsuccess.com/wp-content/uploads/2019/03/25-Great-Inspirational-Quotes-and-Motivational-Quotes-8.jpg" }, { "image": "https://i.pinimg.com/originals/95/cb/ab/95cbab34670fac85105457fc949a70eb.png" }];this.view.vuecarousel.setData(param); |
|  |



# Revision History

App version 1.0.1