Date : 20-Sep-21

TIMELINE

 Version: 1.0.2

1. **OVERVIEW**

Timeline is a Volt MX Iris component for showing the events of a single day/multiple days in a single view of the application.

## **A. Use case:**

### Event Schedule

### Travel Itinerary Plan

### Workout or Daily Planner App

### **B. Features**

### Ready to use UI.

### JSON input format

### Auto resizing height as per content size

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

80-90% (Data can be customizable and skins are not customized but can be changed manually)

# **2. Getting Started**

## **Prerequisites**

Before you start using the Timeline component, ensure you have the following:

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

## **Platforms Supported**

### Mobile

* 1. iOS
	2. 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 TimeLine component, do the following:**

## Open your app project in Volt MX Iris.

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

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

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



1. 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 Volt Mx Iris User Guide.

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

# **3. References**

## **A. Dynamic Usage**

You can also add a **Timeline** 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 like the sample code mentioned below.

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

/\* Creating a component's Object \*/

var TimeLine = new com.voltmx.timeline(

 {

 "clipBounds": true,

 "height": "100%",

 "id": "TimeLine ",

 "isVisible": true,

 "left": "0dp",

 "top": "0dp",

 "width": "100%",

 "zIndex": 1

 }, {}, {});

 TimeLine.eventTitle = "<Title of the Event>";

 TimeLine.masterdata =

 {

 "data":

 [

 {

 "date": "2017-02-01T11:30AM ",

 "desc": "<desc>",

 "name": "<name>",

 "sub1": "<sub1>",

 "sub1icon": "<sublicon>",

 "sub2": "<motive>",

 "sub2icon": "<sub2icon>"

 }

 ]

 };

 TimeLine.verticalLineText = "|\n|\n|\n|\n|\n|\n|\n|\n|\n|\n|\n|\n|\n|\n|\n|\n|\n|\n|\n|\n|\n|\n|\n|\n|\n|\n|\n|\n|\n|\n ";

 TimeLine.verticalLineheight = "100%";

 TimeLine.verticalLineWidth = "7px";

 /\*Adding the TimeLine component to a Form\*/

 this.view.add(TimeLine);

1. **Save** the file.

## **B. Properties**

The properties provided on the **Component** tab allow you to customize the UI elements in the **Timeline** component. You can set the properties directly on the **Component** tab or by writing a JavaScript.

**1. Master Data(masterdata)**

|  |  |
| --- | --- |
| **Category:** | Custom |
| **Description:** | Specifies the list of data sources to the TimeLine. |
| Syntax: | masterdata |
| **Type:** | Data Grid |
| **Read/Write:** | Write |
| **Example:** | this.view.componentID.masterdata ={ data: [ { "date": "2017-02-01T11:30AM ", "desc": "", "name": "session1", "sub1": "", "sub1icon": "", "sub2": "", "sub2icon": "" }, {} ]} |

 **2. Event Title(eventTitle)**

|  |  |
| --- | --- |
| **Category:** | Pass Through |
| **Description:** | Specifies the title of the TimeLine. |
| Syntax: | eventTitle |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.eventTitle = "Title of the Event"; |

1. **Vertical Line Text (****verticalLineText)**

Specifies the text of the Line

1. **Vertical Line** **Height(****verticalLineheight)**

 Specifies the Height of the Line

1. **Vertical Line** **Width(****verticalLineWidth)**

 Specifies the Width of the Line

1. **Segment Master Data(segData)**

 Specifies the Master Data of the segment (so that we can add data statically)

**Example**: this.view.componentID.segData =[

{

 lblDate: "07 JUN",

 lblNum: "03:00 PM",

 lblName: "Welcome Note ",

 lblSub1: "Alex Sion, Ben Cortez",

 imgthumb1: "thumb1.png",

 lblSub2: "Brief discussion on Quarterly product performance",

 imgthumb2: "thumb1.png"

 },{ }

];

## **C. Events**

-- None of the events are exposed.

## **D. APIs**

This API sets the data to the TimeLine. You need to give the input data as a parameter.

**Syntax**

setData(data)

**Parameters**

*data [JSON]*
*The data that you want to insert into the TimeLine.*

**Return Value**

None

**Sample Input Format**

{

 title : <TITLE>,

timeLineList:

[

{

 "name" : <NAME> , (Mandatory)

 "sub1" : <SUB HEADING 1>,

 "sub1 icon" : <IMAGE NAME>

 "date" : <DATE>, (Mandatory)

 "sub2" : <SUB HEADING 2>,

 "sub2 icon" : < IMAGE NAME >

 "desc" : <DESCRIPTION>,

 "thums": <ARRAY OF THUMBNAIL IMAGES>(optional)

 "largeImage" : <ARRAY OF LARGE IMAGES>(optional)

}

]

}

In the given JSON format:

* **title** is a String that specifies the Title of the Event.
* **timeLineList** is a JSON Array that represents the number of sessions in an event.

The **thumbnail image** and **large image** can be of the **src**, **rawbytes**, or **base64** formats.
For example

thums:

[

{"src":<image source>},

{"base64":<base64 String>},

{"rawbytes":<image rawbytes>}

]

The **date** should be in the **yyyy-mm-ddThh:mm<AM/PM>** format. Here, the **T** represents the separator between the Date and the Time.
For example:

24th Feb 2018, 11:30 AM should be written as 2018-02-24T11:30AM.

**Note:** The Timeline list should be sorted in the ascending order by date.

# **4. Revision History**

App version 1.0.2

## **A. Limitations**

* The initial version of the component does not support landscape mode.
* If the browser Height is minimized, then UI gets distorted.