Date : 15/02/2022

Lottie Animator

Version : 1.0.3

**1. Overview**

Lottie Animator supports Lottie which is a JSON-based animation file format that enables designers to ship animations on any platform as easily as shipping static assets. They are small files that work on any device and can scale up or down without pixelation.

.

## **Use case**

#### Animations can highly glorify the UI of an app

## **Features**

Lottie Animator supports following features:

* Play
* Pause
* Resume
* Stop
* Aspect Ratio
* Looping
* Source in String/URL
* Speed
* Autoplay

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

80-90% re-use.

# **Getting Started**

## **Prerequisites**

Before you start using the Lottie Animator component, ensure the following:

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

1. Volt MX Iris

## **Platforms Supported**

### 

### i. Mobile

#### iOS

#### Android

### Tablets

Lottie Files is a collection of animations for Lottie, which is a JSON-based animation file format. The Lottie Animator component adds Lottie animations to Volt MX Iris. You can use the component to add animations from [Lottie Files](https://lottiefiles.com/) to your iOS and Android apps.

The animation source can be provided as a JSON file or a URL. For information about the compatibility of the different types of sources, refer to the following table.

|  |  |  |
| --- | --- | --- |
| Type of Source | Volt MX App | Emulators or Simulators |
| JSON | No | Yes |
| URL | Yes | Yes |

### 

#### **For Android Device**

* Add the Lottie library to the build.gradle entries. For more information, refer to [Configuring Native Settings (Android)](https://docs.kony.com/marketplace/LottieAnimator/Content/Getting_Started.htm#Native_Android)
* For the iOS platform, the component does not require any additional integration steps (such as the gradle build dependency for Android).

1. Note: Minimum SDK version for Android Mobile/ Tablet should be 9.0(28)

[**Configuring Native Settings (Android)**](javascript:void(0);)

To use the Lottie Animator component for Android, you need to add the Lottie library to the build.gradle entries to suffix. To add the library to the build.gradle entries, follow these steps.

1. From the left navigation bar, select **Project Settings**.
2. In the Project Settings window, navigate to **Native** → **Android Mobile/Tablet**.
3. Under **Manifest Permissions, Tags and Gradle Build Entries**, go to **Gradle Entries**.
4. Under **build.gradle entries to Suffix**, type the following code:

dependencies

{

implementation 'com.airbnb.android:lottie:6.4.1'

}

A screenshot of a computer

Description automatically generated

**Note:** Need to enable "Use Google Pay Location Services" for android in the project settings. From **Project Settings** window, go to **Native → Android Mobile/Tablet** and enable **Use** **Google Pay Location Services**.

A screenshot of a computer

Description automatically generated

[Open](javascript:void(0);)

## **Importing the Component**

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

## **To import the Lottie Animator component, do the following:**

## Open your app project in Volt MX Iris.

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

Graphical user interface, text, application

Description automatically generated

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

Graphical user interface, text, application, Teams

Description automatically generated

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.

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 Iris User Guide.

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

# **References**

## **Dynamic Usage of Lottie Animator**

You can also add a **LottieAnimator**  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 LottieAnimator = new com.voltmx.LottieAnimator(

{

"clipBounds": true,

"height": "100%",

"id": "LottieAnimator",

"isVisible": true,

"left": "0dp",

"top": "10%",

"width": "100%",

"zIndex": 1

}, {}, {});

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

LottieAnimator.source = "workfrom.json";

LottieAnimator.loop = 2;

LottieAnimator.speed = 1;

LottieAnimator.autoPlay = true;

LottieAnimator.scaleMode = "Maintain Aspect Ratio";

LottieAnimator.repeatMode = "Restart";

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

this.view.flxLottieContainer.add(LottieAnimator);

}

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**

#### **General**

[[Open](javascript:void(0);)source](javascript:void(0);)

|  |  |  |  |
| --- | --- | --- | --- |
| **Description:** | | Specifies the source of the Lottie File that contains the animation. | |
| **Syntax:** | | Source | |
| **Type:** | | String | |
| **Read/Write:** | | Write | |
| **Remarks:** | | * You can provide a file location or a URL for this property.  1. If you want to provide a local file, make sure that the file exists in the **raw** directory of your project. For more information, refer to [Add and Manage Images and Other Media](https://docs.kony.com/konylibrary/visualizer/visualizer_user_guide/content/Adding_and_Managing_Images.htm#AddImagetoAssetsTab). 2. JSON sources are not supported in the Volt Mx App. | |
| **Example:** | | this.view.componentID.source = "workfrom.json"; | |

[[Open](javascript:void(0);)loop](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies the number of times the animation must repeat. |
| **Syntax:** | Loop |
| **Type:** | Integer |
| **Read/Write:** | Write |
| **Default Value:** | 1 |
| **Example:** | this.view.componentID.loop = 2; |

[[Open](javascript:void(0);)speed](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies the speed at which the animation runs. |
| **Syntax:** | Speed |
| **Type:** | Float |
| **Read/Write:** | Write |
| **Default Value:** | 1 |
| **Example:** | this.view.componentID.speed = 1; |

[[Open](javascript:void(0);)autoPlay](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies whether the animation plays automatically after the image loads. |
| **Syntax:** | autoPlay |
| **Type:** | Boolean |
| **Read/Write:** | Write |
| **Default Value:** | True |
| **Example:** | this.view.componentID.autoPlay = true; |

[[Open](javascript:void(0);)scaleMode](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies the scaling mode for the Lottie image. |
| **Syntax:** | scaleMode |
| **Type:** | * List Selector  1. String |
| **Read/Write:** | Write |
| **Values:** | * Maintain Aspect Ratio  1. Fit To Dimensions |
| **Default Value:** | Maintain Aspect Ratio |
| **Example:** | this.view.componentID.scaleMode = "Maintain Aspect Ratio"; |

#### **Android**

[[Open](javascript:void(0);)repeatMode](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies the movement of the animation when it repeats. |
| **Syntax:** | repeatMode |
| **Type:** | * List Selector  1. String |
| **Read/Write:** | Write |
| **Values:** | * Restart  1. Reverse |
| **Default Value:** | Restart |
| **Remarks:** | This property is only supported on the Android platform. |
| **Example:** | this.view.componentID.repeatMode = "Restart"; |

## **Events**

The component invokes events when its corresponding action is performed. You can configure logic that you want the component to perform when an event occurs.You can configure the events on the **Actions** tab in the **Properties** panel. You can also configure the events by using a JavaScript code. For more information, refer to [Add Actions](https://docs.kony.com/konylibrary/visualizer/visualizer_user_guide/Content/working_with_Action_Editor.htm) in the Volt MX User Guide.

[**[Open](javascript:void(0);)onAnimationEnd**](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Invoked when the animation is completed after the specified number of loops. |
| **Syntax:** | onAnimationEnd |
|  |  |
| **Parameters:** | *eventobject [Object]*: Contains information about the Handle to the widget reference to which the animation is assigned. |
| **Example:** | this.view.componentID.onAnimationEnd = function(eventobject)  {  alert("Animation Completed");  }.bind(this); |

[**[Open](javascript:void(0);)onClick**](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Invoked when the user clicks or taps the animation. |
| **Syntax:** | onClick |
| **Parameters:** | *eventobject [Object]*: Contains information about the Handle to the widget reference to which onClick is assigned. |
| **Example:** | this.view.componentID.onClick = function(eventobject)  {  alert("Animation Clicked");  }.bind(this); |

## **APIs**

The following APIs pertain to the Lottie Animator component:

[Open](javascript:void(0);)[play](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Plays the animation of the Lottie image. |
| **Syntax:** | play() |
| **Parameters:** | None |
| **Return Value:** | None |
| **Example:** | this.view.componentID.play(); |

[[Open](javascript:void(0);)stop](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Stops the animation of the Lottie image. |
| **Syntax:** | stop() |
| **Parameters:** | None |
| **Return Value:** | None |
| **Remarks:** | If the animation is stopped, it restarts when the [play](https://docs.kony.com/marketplace/LottieAnimator/Content/Reference.htm#play) API is used. |
| **Example:** | this.view.componentID.stop(); |

[[Open](javascript:void(0);)pause](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Pauses the animation of the Lottie image. |
| **Syntax:** | pause() |
| **Parameters:** | None |
| **Return Value:** | None |
| **Remarks:** | If the animation is paused, it continues when the [resume](https://docs.kony.com/marketplace/LottieAnimator/Content/Reference.htm#resume) API or [play](https://docs.kony.com/marketplace/LottieAnimator/Content/Reference.htm#play) API is used. |
| **Example:** | this.view.componentID.pause(); |

[Open](javascript:void(0);)[resume](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Resumes the animation of the Lottie image. |
| **Syntax:** | resume() |
| **Parameters:** | None |
| **Return Value:** | None |
| **Example:** | this.view.componentID.resume(); |

[[Open](javascript:void(0);)isAnimating](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies whether the animation of the Lottie image is playing. |
| **Syntax:** | isAnimating() |
| **Parameters:** | None |
| **Return Value:** | *animationStatus [Boolean]*: Returns **true** if the animation is playing; returns **false** if the animation is stopped or paused. |
| **Remarks:** | This API is only supported on the **Android** platform. |
| **Example:** | var animationStatus = this.view.componentID.isAnimating();  if(animationStatus == true)  {  alert("Animation is playing");  }  else  {  alert("Animation is stopped");  } |

## **Limitations:**

* Android version should be greater than 10.0 version

## Lottie Animator does not support landscape mode

## In Android: Animation plays from the beginning whenever user click on play button.

* In IOS: Animation plays from where we paused it.

# **Revision History**

App version 1.0.3