15 Nov 2021

SIGNATURE PAD (1.0.1)

1. Overview

The Signature Pad Component is a custom widget that can be used for Digital Signatures and other written content.

## Use case

### A delivery app where the user needs to sign into the app to authorize and accept the delivery.

## Features:

## This component doesn’t use NFI and works on Iris Preview.

## It is very lightweight to be integrated in any use case.

## Percentage of re-use:

Approximate 90% of reuse.

# Getting Started

## Prerequisites

Before you start using the Signature Pad 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 Signature Pad 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.

## Graphical user interface, text 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 Signature Pad work in real time.

## Modifying the UI of the scenes

You can modify the UI of the carousel by editing a CSS file. To access the CSS file, follow these steps:

1. From the **Project** explorer, go to **Web → Localfiles → signature\_pad → css.**
2. From the **css** folder, select **signature-pad.css**.
3. Use the text editor to edit the CSS based on your preference.

Graphical user interface, text

Description automatically generated

## After you modify the CSS, you can Building and previewing the app to view the changes on the carousel.

# References

## Dynamic Usage

You can also add **Signature Pad** 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.



|  |
| --- |
| /\*Creating the component's object \*/  var signaturePadObj = new com.voltmxqfs.signaturepad({  id: "signaturepad",  isVisible: true,  top:"0dp",  left:"0dp",  width:"100%",  height:"100%",  clipBounds: true,  layoutType: voltmx.flex.FREE\_FORM,  skin: "slFbox",  zIndex:1  },{},{});  /\*Adding the component to a Form\*/  this.view.add(signaturePadObj); |

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

2. Save the file

## Properties

The properties provided on the **Component** tab allows you to customize the elements in the **Signature Pad** 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.

**Note:** If you copy code from any example, make sure that you replace componentID with the name of your component.

General Properties

No Properties

## Events

## No Events

## API’s

### getSignature

|  |  |
| --- | --- |
| **Description:** | Invoked to get signature image base64 string. |
| **Syntax:** | getSignature() |
| **Parameters:** | None |
| **Return Value:** | sgnBase64 |
| **Example:** | this.view.componentID.getSignature(); |

# Revision History

App version 1.0.1:

## Known Issues

## No Known Issues

## Limitations

When we modify the browser size the content of the canvas is cleared.