11th April 2023

Wallet(1.0.0)

# Overview

A Wallet component to preview and add the pass into the iOS wallet app.

## Use case

## Passes generated from the backend will be adding to the iOS wallet app.

## Percentage of re-use:

Approximate 90 % of reuse.

## Features

1. The component Wallet utilizes native frameworks to add the passes into the iOS wallet app from the base64 string or the . pkpass file.
2. Uses NFI 3.0

# Getting Started

## Prerequisites

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

* Volt MX Iris

## Platforms Supported

### Mobile - iOS

### Tablets - iOS

## Importing the component

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

## **To import the** Wallet **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, application

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 Scanner work in real time.

# References

## Dynamic Usage

You can also add Wallet 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 Wallet component instance \*/

var Wallet = new com.technohub.AppWallet({

"height": "100%",

"id": "AppWallet",

"isVisible": true,

"left": "0dp",

"masterType": constants.MASTER\_TYPE\_USERWIDGET,

"isModalContainer": false,

"skin": "slFbox",

"top": "0dp",

"width": "100%",

"zIndex": 1,

"overrides": {

"Wallet": {

"right": "viz.val\_cleared",

"bottom": "viz.val\_cleared",

"minWidth": "viz.val\_cleared",

"minHeight": "viz.val\_cleared",

"maxWidth": "viz.val\_cleared",

"maxHeight": "viz.val\_cleared",

"centerX": "viz.val\_cleared",

"centerY": "viz.val\_cleared"

}

}

}, {

"overrides": {}

}, {

"overrides": {}

});

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

this.view.add(Wallet);

},

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

## Configuring Deployment Target

1. From the left navigation menu, click **Project Settings**.
2. In the **Project Settings** window, go to **Native** → **iPhone/iPad**.
3. Under **Target Versions**, from the **iOS Version** list, select **11.0** or higher.  
   Graphical user interface, application

   Description automatically generated

## API

### isPassLibraryAvailable

|  |  |
| --- | --- |
| **Description:** | Checks passkit library is available in the iPhone device. |
| Syntax: | isPassLibraryAvailable() |
| **Parameters:** | None |
| **Return Value:** | None |
| **Remarks:** | NA. |
| **Example:** | this.view.AppWallet.isPassLibraryAvailable() |

### addPassFromBase64

|  |  |
| --- | --- |
| **Description:** | Adds the pass from the base64 string. |
| Syntax: | addPassFromBase64(base64, completionHandler) |
| **Parameters:** | Base64 - String, completion Handler - Callback |
| **Return Value:** | None |
| **Remarks:** | NA. |
| **Example:** | this.view.AppWallet.addPassFromBase64 (base64, this.competionHandler)  competionHandler: function(data) {  } |

### addPassFromFile

|  |  |
| --- | --- |
| **Description:** | Adds the pass from the .pkpass file, which is palced in the iphone/mainbundle folder. |
| Syntax: | addPassFromFile(fileName, completionHandler) |
| **Parameters:** | Filename - String, completion Handler - Callback |
| **Return Value:** | None |
| **Remarks:** | NA. |
| **Example:** | this.view.AppWallet.addPassFromFile(fileName, this.competionHandler)  competionHandler: function(data) {  } |

### getPasses

|  |  |
| --- | --- |
| **Description:** | Gives all the passes objects added in the Wallet app. To get this function work, it passes capability must be added to the bundle identifier. |
| Syntax: | getPasses() |
| **Parameters:** | None |
| **Return Value:** | None |
| **Remarks:** | NA. |
| **Example:** | this.view.AppWallet.getPasses() |

### removePass

|  |  |
| --- | --- |
| **Description:** | Removes the pass which is being passed as a parameter. |
| Syntax: | removePass(pass) |
| **Parameters:** | Pass – native PKPass object |
| **Return Value:** | None |
| **Remarks:** | NA. |
| **Example:** | this.view.AppWallet.removePass(pass); |

### getPassWithPassTypeIdentifierAndSerialNumber

|  |  |
| --- | --- |
| **Description:** | Gets the native pass object with the identifier and the serial number |
| Syntax: | getPassWithPassTypeIdentifierAndSerialNumber(identifier, serialNumber) |
| **Parameters:** | Identifier – String – Pass identifier, Serial Number – String – Serial Number. |
| **Return Value:** | None |
| **Remarks:** | NA. |
| **Example:** | this.view.AppWallet. getPassWithPassTypeIdentifierAndSerialNumber (identifier, serialNumber); |

### addPasses

|  |  |
| --- | --- |
| **Description:** | Adds multiple passes to the Wallet |
| Syntax: | addPasses([passes]) |
| **Parameters:** | Passes – [PKPass] - multiple native passes objects |
| **Return Value:** | None |
| **Remarks:** | NA. |
| **Example:** | this.view.AppWallet.addPasses ([passes]); |

## App version 1.0.0:

## Limitations

NA

## Known Issue

NA