12 June 2023

**BOTTOM APP BAR [1.0.0] (FOOT MENU)**

**Details**

Overview.

Bottom App Bar allows you add a footer menu or a navigation bar to application, to navigate between form. This includes a pre-defined function to navigate between forms which can be overridden. The app bar is divided into five segments which consists of an Font-Icon, Label, segment bottom bar, App top bar. Which can be edited or turned depending on one’s preference.

FEATURES.

E

D

C

B

A



**A**. Notification Icon (Label) – to display notification. Visibility if false by default.

**B**. Top border (Flex Container) – For contrast purpose, if the app bar and form have the same color.

**C**. Tab bottom border (Flex Container) – By default only the selected tab or segments bottom border is visible. Note that if you decide to make use of tab bottom border, Tab labels will be removed (Recommended for a better User Interface).

**D**. Name (Label) – Names for Segments. Note that if you decide to make use of tab bottom border, Tab labels will be removed (Recommended for a better User Interface).

**E**. Icons (Label) – Icons for segments or tabs. By default, it uses Font Awesome Icons. This can be changed creating a custom skin set with the preferred Font-Family. And the name of skin passed into the Tab Icon Normal.



Getting Started.

A. Prerequisites

 Before you start using the Hamburger Menu component, ensure the following:

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

 B. Platforms Supported

1. Mobile
2. iOS
3. Android
4. Tablets
5. PWA

C. Importing the Component

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

**To import the Hamburger Menu component, do the following:**

 1. Open your app project in Volt MX Iris.

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



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.
2. Set component to ones preferrence.

PROPERTIES.





* General
* Active Tab – Determines which tab is to be active in a form. And the active skin will be applied to that tab. It contains a drop-down list of all the tabs.

Active tab



* Use Default Nav function (Boolean) – Uses the pre-defined function to navigate between form if set to true.
* Setup
* Include Tab Title (Boolean) – If set to true by default Tab names will be visible but Tab bottom border will be unavailable.

 

 Left: Preview of Tab title included and setup Right: Setup

* Include Tab Bottom Border (Boolean) – If set to true tab bottom border/bar will be visible for the active tab but Tab names will not applicable.

 

Left: Preview of Tab Bottom Border Included Right: Setup

* Include Notify Icons (Boolean) – When set to false notifications icons are removed.
* Include AppBar Top Border (Boolean) – When set to false, top appbar border visibility is set to false.
* Icons
* The icon sub group is us to setup icons for the navigation menu, depending on the Font Family used. By default, its FontAwesome Thin. Simply type in the name of the Icon.

 

Preview of icon names and their respective icons (Default icons)

 

Preview of icon names and their respective icons.

* Title
* The title sub group is for setting tab names.

 

* Navigate
* This sub group is only available if ‘Use Default Nav function (Boolean)’ is set to true. This enables to simply pass in the names of the forms to their respective tabs.

 



In the above illustration formA is mapped TaB as shown by the red arrows, formB to Tab C. etc.

Meaning if TabB (Search Icon), the user will be navigated to formA.

* Flex
* Use this to set the dimensions of the widgets.



* Skin
* Used to set skins to the widgets by creating custom skins and passing the name to the required widget.

 

Normal - skin applied to a widget in normal state i.e not clicked.

Active – skin applied to a widget after widget is clicked.

EVENTS.

1. **onTabClick**

|  |  |
| --- | --- |
| Description: | Invoked when the user clicks a tab. |
| Syntax: | onTabClick |
| Parameters: | tabName:Id/name of Tab clicked. |

Usage:



1. **navigateTo**

|  |  |
| --- | --- |
| Description: | Invoked when the user clicks a tab. |
| Syntax: | navigateTo |
| Parameters: | tabName:Id/name of Tab clicked. |

Usage:

****

Note: Both onTabClick and navigateTo have the same parameters and came be used a similar way only different is that, navigateTo is only called when Use Default Nav function (Boolean)  is set to false while onTabClick is called regardless of Default Nav function (Boolean) state.

MarketPlace URl : https://marketplace.kony.com/items/

*By ZaNa*