Date :29 Oct 2021

GET DIRECTIONS USING GOOGLE MAPS - APPLE MAPS

VERSION: 1.1.1

**1. OVERVIEW**

The Get Directions using Google Maps-Apple Maps component launches Google Maps or Apple Maps to provide navigation directions to the user. Under the hood, it utilizes URL schemes provided by [Google Maps](https://developers.google.com/maps/documentation/urls/guide) and [Apple Maps](https://developer.apple.com/library/content/featuredarticles/iPhoneURLScheme_Reference/MapLinks/MapLinks.html). To get Place IDs we can configure from the documentation [Places API](https://developers.google.com/maps/documentation/places/web-service/place-id)

## **Use case:**

## Provide navigation directions to your app users using Google Maps/ Apple Maps

## **Features**

## Configure origin, destination, travel modes etc. properties directly from Iris property panel.

## Customize look and feel of the component as per your company's branding guidelines.

## The component uses URL schemes provided by Google Maps/ Apple Maps.

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

90%

# **Getting Started**

## **Prerequisites**

Before you start using the Get Direction component, ensure the following:

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

## **Platforms Supported**

### Mobile

#### iOS

#### Android

### Tablets

### 

## **Importing the Component**

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

## **To import the Get Direction component, do the following:**

## 1. 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

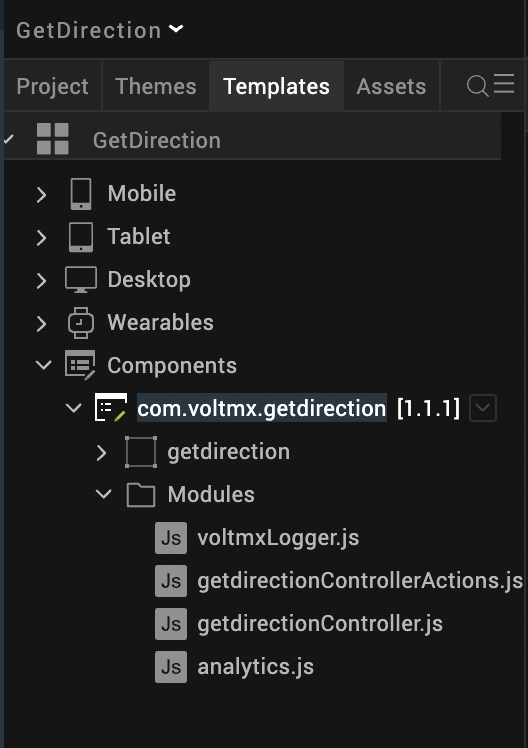
4. Click **Browse** to navigate to the location of the component, select the component, and then click **Import**. The component and its associated widgets and modules are added to your project.

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 VoltMx Iris User Guide.

You can then run your app to see the Get Direction Component work in real time.



# **References**

## **Dynamic Usage**

You can also add a **Get Direction** 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.

|  |
| --- |
| define({  onNavigate: function(){  /\* Creating component s object \*/  var getdirection = new com.voltmx.getdirection(  {  "autogrowMode": voltmx.flex.AUTOGROW\_NONE,  "clipBounds": true,  "height": "80dp",  "id": "getdirection",  "isVisible": true,  "layoutType": voltmx.flex.FREE\_FORM,  "left": "30%",  "masterType": constants.MASTER\_TYPE\_USERWIDGET,  "skin": "slFbox",  "top": "20%",  "width": "80dp"  }, {}, {});  /\* Setting the component's Properties \*/  getdirection.icon = getdirection.png ;  getdirection.iconScaleMode = constants.IMAGE\_SCALE\_MODE\_MAINTAIN\_ASPECT\_RATIO;  getdirection.headerMainText = "Get Directions";  getdirection.iOSMapOptions = "Both Maps";  getdirection.travelModesForAppleMaps = "driving";  getdirection.travelModesForGoogleMaps = "driving";  /\* For iOS Devices\*/  getdirection.originPlace = "paris";  getdirection.destinationPlace = "frankfurt";  /\* For Android Devices\*/  getdirection.destinationPlaceId = "ChIJxZZwR28JvUcRAMawKVBDIgQ";  getdirection.originPlaceId = "ChIJD7fiBh9u5kcRYJSMaMOCCwQ";  getdirection.wayPoints = "berlin|amsterdam";  getdirection.waypointPlaceIds="ChIJAVkDPzdOqEcRcDteW0YgIQQ|ChIJVXealLU\_xkcRja\_At0z9AGY";  getdirection.turnByTurnNavigation = false;  /\*adding component to a form\*/  this.view.add(getdirection);  }  }); |

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**

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

General Properties

**1. icon Source(getDirectionIconSrc)**

|  |  |
| --- | --- |
| **Description:** | Specifies the source of the component's icon. |
| **Syntax:** | getDirectionIconSrc |
| **Type:** | String |
| **Remarks:** | Make sure that the source file is present in the project resources. |
| **Example:** | this.view.componentID.getDirectionIconSrc = directionImg.png; |

**2. icon Scale Mode(iconScaleMode)**

|  |  |
| --- | --- |
| **Description:** | Specifies the scale mode of the component's icon. |

**Syntax:** iconScaleMode

**Type :**  Constant

**Read/Write**: Read+write

**Values:** constants.IMAGE\_SCALE\_MODE\_CROP

constants.IMAGE\_SCALE\_MODE\_FIT\_TO\_DIMENSIONS

constants.IMAGE\_SCALE\_MODE\_MAINTAIN\_ASPECT\_RATIO

**Example**: this.view.componentID.iconScaleMode = constants.IMAGE\_SCALE\_MODE\_MAINTAIN\_ASPECT\_RATIO;

**3. Origin(originPlace)**

|  |  |
| --- | --- |
| **Description:** | Specifies the starting point for the directions. |
| Syntax: | originPlace |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.originPlace = paris ; |
| **Remarks:** | The **saddr** parameter in Apple Maps URL scheme.  The Origin Parameter in Google Maps Directions URL scheme. |
|  | For more information, you can refer to the [Google Maps](https://developers.google.com/maps/documentation/urls/guide#directions-action) documentation and [Apple Maps](https://developer.apple.com/library/content/featuredarticles/iPhoneURLScheme_Reference/MapLinks/MapLinks.html) reference. |

**4. Destination(destinationPlace)**

|  |  |
| --- | --- |
| **Description:** | Specifies the endpoint of the directions. |
| Syntax: | destinationPlace |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.destinationPlace = Frankfurt; |
| **Remarks:** | 1. For more information, you can refer to the [Google Maps](https://developers.google.com/maps/documentation/urls/guide#directions-action)  Documentation and [Apple Maps](https://developer.apple.com/library/content/featuredarticles/iPhoneURLScheme_Reference/MapLinks/MapLinks.html) reference.  2.An exception stating “Please provide a valid destination” is thrown if there is no destination provided. |
|  |  |

**iOS**

**1. Map options(iOSMapoptions)**

|  |  |  |
| --- | --- | --- |
| **Description:** |  | This property is used to ask the users which map they want to use. This property invokes an **Action Sheet** pop-up that presents the choice of map to the user - Google Maps, or Apple Maps. This property has the following options:   * Apple Maps * Google Maps * Both Maps |

**Syntax:** iOSMapOptions

**Type:**  String

**Read/Write:** Read + Write

**Default Value:** ”Both Maps”

**Remarks:** The component throws an exception if the given value is not a valid string.

The **Action Sheet** will not pop up if you enable only one map.

**2.Action Sheet Main Text**

|  |  |
| --- | --- |
| **Description:** | Specifies the text to be displayed as the main header of the Action Sheet. You can refer to the [Map Options](https://docs.kony.com/marketplace/V8Marketplace/Content/Marketplace/getdirection.htm#Map) property for more information regarding Action Sheet. |
| **Syntax:** | headerMainText |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Remarks:** | The Action Sheet will not display the Header if you do not provide any text. |
| **Example:** | this.view.componentID.headerMainText = "Launch Map"; |

**3.Action Sheet Sub Text**

|  |  |
| --- | --- |
| **Description:** | Specifies the text to be displayed as the sub header of the Action Sheet. You can refer to the [Map Options](https://docs.kony.com/marketplace/V8Marketplace/Content/Marketplace/getdirection.htm#Map) property for more information regarding Action Sheet. |
| **Syntax:** | headerSubText |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Remarks:** | The Action Sheet will not display the Sub Header if you do not provide any text. |
| **Example:** | this.view.componentID.headerSubText = "Which map would you like to use?"; |

**Action Sheet Skin Properties**

An **Action Sheet** is a pop-up in iOS that presents the choice of map to the user - Google Maps, or Apple Maps.  
You can change the exposed skins of the Action Sheet component in the **Skins** tab under the **Properties** panel on the right side of VoltMx Iris

**1. Sheet Background Skin**

|  |  |
| --- | --- |
| **Description:** | Specifies the skin that is applied to the background container of **Header** and **Options** of the Action Sheet. |
| **Syntax:** | sheetBackgroundSkin |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.sheetBackgroundSkin = "sheetBGSkin"; |

**2. Header Main Text Skin**

|  |  |
| --- | --- |
| **Description:** | Specifies the skin that is applied to the header's main text in the Action Sheet. |
| **Syntax:** | headerMainTextSkin |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.headerMainTextSkin = "headerTextSkin"; |

**3. Header Sub Text Skin**

|  |  |
| --- | --- |
| **Description:** | Specifies the skin that is applied to the header's sub text in the Action Sheet. |
| **Syntax:** | headerSubTextSkin |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.headerSubTextSkin = "headerTextSkin2"; |

**4. Separator Skin**

|  |  |
| --- | --- |
| **Description:** | Specifies the skin that is applied to the separator lines in the Action Sheet. |
| **Syntax:** | separatorSkin |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.separatorSkin = "separatorSkin"; |

**5. Options Skin**

|  |  |
| --- | --- |
| **Description:** | Specifies the skin to be applied to the options in the Action Sheet. |
| **Syntax:** | optionsSkin |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.optionsSkin = "optSkin"; |

**6. Options Focus Skin**

|  |  |
| --- | --- |
| **Description:** | Specifies the skin to be applied to the options in the Action Sheet when it is focused. |
| **Syntax:** | optionsFocusSkin |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.optionsFocusSkin = "optFocusSkin"; |

**7. Footer Background Skin**

|  |  |
| --- | --- |
| **Description:** | Specifies the skin to be applied to the background container of the Footer in the Action Sheet. |
| **Syntax:** | footerBackgroundSkin |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.footerBackgroundSkin = "footBGSkin"; |

**8. Footer Button Skin**

|  |  |
| --- | --- |
| **Description:** | Specifies the skin to be applied to the Footer button in the Action Sheet. |
| **Syntax:** | footerButtonSkin |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.footerButtonSkin = "footBtnSkin"; |

**9. Footer Button Focus Skin**

|  |  |
| --- | --- |
| **Description:** | Specifies the skin to be applied to the Footer button in the Action Sheet when it is focused. |
| **Syntax:** | footerButtonFocusSkin |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Example:** | this.view.componentID.footerButtonFocusSkin = "footBtnFocusSkin"; |

**Google Maps Properties**

**1. Travel Mode**

|  |  |
| --- | --- |
| **Description:** | Defines the method of travel. The Options for the Travel Mode are:   * **Driving** - It routes through major roads and streets. * **Walking** - It prefers available pedestrian paths and sidewalks. * **Bicycling** - It routes through available bike paths and preferred streets. * **Transit** - It routes through available Bus or Train transits. |
| **Syntax:** | travelModesForGoogleMaps |
| **Type:** | List Selector |
| **Read/Write:** | Read + Write |
| **Default Value:** | **None** |
| **Remarks:** | * This property can be set to **none** if the user does not want to give any travel mode. If the mode is set to **none**, Google Maps will show the most relevant modes for the specified route or the user preferred mode. * It refers to the **travelmode** parameter of the Google Maps Directions URL scheme. For more details you can refer to [Google Maps Documentation](https://developers.google.com/maps/documentation/urls/guide#directions-action). * If you give an empty string or an invalid value, the component throws an exception stating "Please provide a valid travel mode for Google Maps". |
| **Example:** | this.view.componentID.travelModesForGoogleMaps = "Driving"; |

**2. Origin Place ID**

|  |  |
| --- | --- |
| **Description:** | Specifies Google Maps **Place ID** of the **Origin**. A **Place ID** is a unique identifier for a place in the Google Places database and Google Maps. |
| **Syntax:** | originPlaceId |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Remarks:** | * This property refers to the **origin\_place\_id** parameter of Google Maps Directions URL scheme. For more details you can refer to [Google Maps Documentation](https://developers.google.com/maps/documentation/urls/guide#directions-action). * Make sure this property is used only if the **Origin** is provided. |
| **Example:** | this.view.componentID.originPlaceId = "frankfurt"; |

**3 . Destination Place ID**

|  |  |
| --- | --- |
| **Description:** | Specifies Google Maps **Place ID** of the **Destination**. A **Place ID** is a unique identifier for a place in the Google Places database and Google Maps. |
| **Syntax:** | destinationPlaceId |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Remarks:** | * This property refers to the **destination\_place\_id** parameter of Google Maps Directions URL scheme. For more details you can refer to [Google Maps Documentation](https://developers.google.com/maps/documentation/urls/guide#directions-action). * Make sure this property is used only if the **Destination** is provided. |
| **Example:** | this.view.componentID.destinationPlaceId = "paris"; |

**4 . Waypoints**

|  |  |
| --- | --- |
| **Description:** | Specifies one or more intermediary places between the origin and destination to route the directions through.   * Multiple waypoints can be specified by using the pipe character ( | ) to separate places (for example, Berlin,Germany|Paris,France). * Each waypoint can be either a place name, address or comma-separated latitude/longitude coordinates. |
| **Syntax:** | wayPoints |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Remarks:** | * The number of waypoints allowed varies by the platform where the link opens, with up to **three** waypoints supported on mobile browsers, and a maximum of **nine** waypoints supported otherwise. * Waypoints are displayed on the map in the same order that they are listed in the URL. |
| **Example:** | this.view.componentID.wayPoints = "berlin|amsterdam"; |

**5. waypoints Place IDs**

|  |  |
| --- | --- |
| **Description:** | Specifies a list of Google Maps Place IDs to match the list of waypoints. A **Place ID** is a unique identifier for a place in the Google Places database and Google Maps.   * Each waypoint Place ID should be separated using the pipe character ( | ) |
| **Syntax:** | waypointPlaceIds |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Remarks:** | * This property refers to the **waypoints\_place\_ids** parameter of Google Maps Directions URL scheme. For more details you can refer to [Google Maps Documentation](https://developers.google.com/maps/documentation/urls/guide#directions-action). * Make sure it is used only if waypoints are provided. |
| **Example:** | this.view.componentID.waypointPlaceIds = "ChIJAVkDPzdOqEcRcDteW0YgIQQ|ChIJVXealLU\_xkcRja\_At0z9AGY"; |

**6. Turn By Turn Navigation**

|  |  |
| --- | --- |
| **Description:** | Launches either turn-by-turn navigation or route preview to the specified destination, based on whether the origin is available.  The map launches a route preview if:   * The **origin** is not close to the user's current location. * The user's current location is not available.   The map launches a turn-by-turn navigation if:   * The user does not specify an **origin** - The origin defaults to the user's current location. * The **Origin** is close to the user's current location. |
| **Syntax:** | turnByTurnNavigation |
| **Type:** | Boolean |
| **Read/Write:** | Read + Write |
| **Remarks:** | This property refers to the **dir\_action = navigate** parameter of Google Maps Directions URL scheme. For more details you can refer to [Google Maps Documentation](https://developers.google.com/maps/documentation/urls/guide#directions-action). |
| **Example:** | this.view.componentID.turnByTurnNavigation = true; |

**Apple Maps properties**

**1. Travel Modes**

|  |  |
| --- | --- |
| **Description:** | Specifies the transport type for Apple Maps. The Options for the Travel Mode are:   * **driving** - It refers to **d (by car)**. * **walking** - It refers to **w (by foot)**. * **transit** - It refers to **r (by public transit)**. |
| **Syntax:** | travelModesForAppleMaps |
| **Type:** | List Selector |
| **Default Value:** | **None** |
| **Read/Write:** | Read + Write |
| **Remarks:** | * It refers to the **dirflg** parameter of the Apple Maps URL scheme. For more details you can refer to [Apple Maps URL Scheme reference](https://developer.apple.com/library/content/featuredarticles/iPhoneURLScheme_Reference/MapLinks/MapLinks.html). * If you give an empty string or an invalid value, the component throws an exception stating "Please provide a valid travel mode for Apple Maps". |
| **Example:** | this.view.componentID.travelModesForGoogleMaps = "driving"; |

## **Events**

|  |  |
| --- | --- |
|  |  |

---No Events are Used

## **APIs**

## --No APIs are exposed.

# **Revision History**

App version 1.1.1

**Limitations:**

1 . For iOS we will get the Both “Apple Map” & “Google Map” options.

2 . For Android we will get only “Google Map” option.