Date : 07-Feb-2-2022

RECAPTCHA

version: 1.1.1

**1. OVERVIEW**

The reCAPTCHA component supports Google reCAPTCHA V3 and Google reCAPTCHA V2 (“I’m not a robot” Checkbox). When using with Google reCAPTCHA V3, it verifies whether an interaction is legitimate without any user interaction. It returns a score (based on interactions with your site) that you can use to take appropriate actions. When using with the "I'm not a robot" Checkbox, it requires the user to click a checkbox indicating the user is not a robot. This will either pass the user immediately (with No CAPTCHA) or challenge them to validate whether they are human.

You can use the component in your web applications with a site key and a secret key. For more information, you can refer to the [reCAPTCHA documentation](https://developers.google.com/recaptcha/docs/v3).

## **Use case:**

### Protects your site from spam and abuse.

### **B. Features:**

### Ready to use UI.

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

100% (Data can be customizable)

# **Getting Started**

## **Prerequisites**

Before you start using the reCAPTCHA component, ensure you have the following:

* [HCL Foundry](https://manage.hclvoltmx.com/)
* Volt MX Iris
* A **Site Key**and **Secret Key** from reCAPTCHA.

You need to register at [reCAPTCHA v3  |  Google Developers](https://developers.google.com/recaptcha/docs/v3) to get the keys.

## **Platforms Supported**

### PWA

## **Importing the Component**

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

## **To import the reCAPTCHA component, do the following:**

## Open your app project in Volt MX Iris.

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

## Graphical user interface, text Description automatically generated

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

## Graphical user interface, text, application Description automatically generated

## 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#add-a-component-to-a-form).

# **3.References**

## **A. Dynamic Usage**

You can also add a **reCaptcha** 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.

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

/\* Creating a component's Object \*/

createRecaptchaComponent: function(){

var recaptcha = new com.voltmxmp.recaptcha(

{

"clipBounds": true,

"id": "recaptcha",

"isVisible": true,

"left": "6dp",

"skin": "slFbox",

"top": "20%",

"width": "100%",

"height": "70%",

}, {}, {});

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

recaptcha.siteKey = "<reCAPTCHA-SITE-KEY>";

recaptcha.secretKey = "<reCAPTCHA-SECRET-KEY>";

recaptcha.pageName = "Formv2";

recaptcha.autoLoad = true;

recaptcha.siteKeyType = "V2"; /\* or "V2\_CHECKBOX" \*/

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

this.view.add(recaptcha);

},

1. **Save** the file.

## **B. Properties**

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

**[Open](javascript:void(0);)1. Site Key**

|  |  |
| --- | --- |
| **Category:** | recaptcha |
| **Description:** | The site key that you generated from reCAPTCHA. |
| **Syntax**: | siteKey |
| **Type:** | String |
| **Read/Write:** | Write |
| **Example:** | this.view.componetID.siteKey = "<reCAPTCHA-SITE-KEY>"; |

**2. Secret Key**

|  |  |
| --- | --- |
| **Category:** | recaptcha |
| **Description:** | The secret key that you generated from reCAPTCHA. |
| **Syntax**: | secretKey |
| **Type:** | String |
| **Read/Write:** | Write |
| **Example:** | this.view.componetID.secretKey = "<reCAPTCHA-SECRET-KEY>"; |

**3. Page Name**

|  |  |
| --- | --- |
| **Category:** | recaptcha |
| **Description:** | Specifies the name that you want to use for the page that contains the component. |
| **Syntax**: | pageName |
| **Type:** | String |
| **Read/Write:** | Write |
| **Example:** | this.view.componentID.pageName = "Formv2"; |

**4. Auto Load**

|  |  |
| --- | --- |
| **Category:** | recaptcha |
| **Description:** | Enable or disable the Protection by reCAPTCHA banner on the page. |
| **Syntax**: | autoLoad |
| **Type:** | Boolean |
| **Read/Write:** | Write |
| **Remarks:** | * If you set this property to **true**, the component displays the banner when it loads the page. * If you set this property to **false**, you can use the renderCaptcha API to render the reCAPTCHA logo. |
| **Example:** | this.view.componentID.autoLoad = false; |

**5. Site Key Type**

|  |  |
| --- | --- |
| **Category:** | siteKeyType |
| **Description:** | The type of site key that you generated from reCAPTCHA. |
| **Syntax**: | siteKeyType |
| **Type:** | String |
| **Read/Write:** | Write |
| **Remarks:** | Either V3 or V2\_CHECKBOX |
| **Example:** | this.view.componentID.siteKeyType = "V2"; |

## **C. Events**

**1. verificationCallback**

|  |  |
| --- | --- |
| **Syntax:** | verificationCallback |
| **Parameters:** | result [JSON] The response from the reCAPTCHA console. |
| **Return Value:** | None |
| **Sample Input Format:** | this.view.componentID.verificationCallback = function(result){  alert("Response: "+result);  }.bind(this); |

**2. errorCallback**

|  |  |
| --- | --- |
| **Syntax:** | errorCallback |
| **Parameters:** | error [JSON] Error information such as error code and error message*.* |
| **Return Value:** | None |
| **Sample Input Format:** | this.view.componentID.errorCallback = function(error){ |

## **D. APIs**

**1. renderCaptcha**

|  |  |
| --- | --- |
| **Syntax:** | renderCaptcha |
| **Parameters:** | *None* |
| **Return Value:** | None |
| **Sample Input Format:** | this.view.componentID.renderCaptcha(); |

# **4. Revision History**

App version 1.1.1

**A. Known issues:** NA

**B. Limitations:** NA