**参考地址 ：** [**http://mirari.cc/v-viewer/**](http://mirari.cc/v-viewer/)

**安装**

通过NPM从GitHub安装

npm install v-viewer

**用法**

要使用v-viewer，只需导入它和css文件，然后调用Vue.use()install。

<template>

<div id="app">

*<!-- directive -->*

<div class="images" v-viewer>

<img src="1.jpg">

<img src="2.jpg">

...

</div>

*<!-- component -->*

<viewer :images="images">

<img v-for="src in images" :src="src" :key="src">

</viewer>

</div>

</template>

<script>

**import** 'viewerjs/dist/viewer.css'

**import** Viewer **from** 'v-viewer'

**import** Vue **from** 'vue'

Vue.use(Viewer)

**export** **default** {

data() {

images: ['1.jpg', '2.jpg']

}

}

</script>

**支持UMD**

**浏览器**

<link href="//path/viewer.css" rel="stylesheet">

<script src="//path/vue.js"></script>

<script src="//path/viewer.js"></script>

<script src="//path/v-viewer.js"></script>

...

<script>

Vue.use(VueViewer.default)

</script>

**CommonJS的**

**var** VueViewer = require('VueViewer')

**AMD**

require(['VueViewer'], **function** (VueViewer) {});

**指令的用法**

只需将指令添加v-viewer到任何元素，然后img将处理其中的所有元素viewer。

您可以设置如下选项： v-viewer="{inline: true}"

按选择器获取元素，然后根据需要使用el.$viewer获取viewer实例。

<template>

<div id="app">

<div class="images" v-viewer="{movable: false}">

<img v-for="src in images" :src="src" :key="src">

</div>

<button type="button" @click="show">Show</button>

</div>

</template>

<script>

**import** 'viewerjs/dist/viewer.css'

**import** Viewer **from** 'v-viewer'

**import** Vue **from** 'vue'

Vue.use(Viewer)

**export** **default** {

data() {

images: ['1.jpg', '2.jpg']

},

methods: {

show () {

**const** viewer = **this**.$el.querySelector('.images').$viewer

viewer.show()

}

}

}

</script>

**Direcitve修饰符**

**静态的**

该viewer渲染器将只进行一次指令绑定后执行。

如果您确定此元素中的图像不会再次更改，请使用它以避免不必要的重新渲染。

<div class="images" v-viewer.static="{inline: true}">

<img v-for="src in images" :src="src" :key="src">

</div>

**组件的用法**

您只需导入组件并在本地注册即可。

使用[范围插槽](https://vuejs.org/v2/guide/components.html" \l "Scoped-Slots)自定义图像的显示。

听取inited事件以获取viewer实例或使用this.refs.xxx.$viewer。

<template>

<div id="app">

<viewer :options="options" :images="images"

@inited="inited"

class="viewer" ref="viewer"

>

<template slot-scope="scope">

<img v-for="src in scope.images" :src="src" :key="src">

{{scope.options}}

</template>

</viewer>

<button type="button" @click="show">Show</button>

</div>

</template>

<script>

**import** 'viewerjs/dist/viewer.css'

**import** Viewer **from** "v-viewer/src/component.vue"

**export** **default** {

components: {

Viewer

},

data() {

images: ['1.jpg', '2.jpg']

},

methods: {

inited (viewer) {

**this**.$viewer = viewer

},

show () {

**this**.$viewer.show()

}

}

}

</script>

您可以替换images使用trigger，接受任何类型的道具。当trigger更改时，组件将重新呈现查看器。

<viewer :trigger="externallyGeneratedHtmlWithImages">

<div v-html="externallyGeneratedHtmlWithImages"/>

</viewer>

**选项和方法**

请参阅[viewer.js](https://github.com/fengyuanchen/viewerjs)。 （https://github.com/fengyuanchen/viewerjs）

**插件选项**

**名称**

* 类型： String
* 默认： viewer

如果您需要避免名称冲突，可以像这样导入：

<template>

<div id="app">

<div class="images" v-vuer="{movable: false}">

<img v-for="src in images" :src="src" :key="src">

</div>

<button type="button" @click="show">Show</button>

</div>

</template>

<script>

**import** 'viewerjs/dist/viewer.css'

**import** Vuer **from** 'v-viewer'

**import** Vue **from** 'vue'

Vue.use(Vuer, {name: 'vuer'})

**export** **default** {

data() {

images: ['1.jpg', '2.jpg']

},

methods: {

show () {

**const** vuer = **this**.$el.querySelector('.images').$vuer

vuer.show()

}

}

}

</script>

**defaultOptions**

* 类型： Object
* 默认： undefined

如果需要设置查看器默认选项，可以像这样导入：

**import** Viewer **from** 'v-viewer'

**import** Vue **from** 'vue'

Vue.use(Viewer, {

defaultOptions: {

zIndex: 9999

}

})

您可以在任何其他时间重置默认选项：

**import** Viewer **from** 'v-viewer'

**import** Vue **from** 'vue'

Vue.use(Viewer)

Viewer.setDefaults({

zIndexInline: 2017

})