

# Handle Global Events in Vue.js

---

 [alligator.io/vuejs/vue-global-events](https://alligator.io/vuejs/vue-global-events)

Vue's method of event handling is great. Just throw a little

`@eventName="eventHandler"` attribute on an element and call it a day. However, such conveniences don't extend to global events by default. You'll have to go with the tried-and-true method with `addEventListener`. Or, you could use this itty-bitty handy little Vue component brought to you by [Damian Dulisz](#) and [Eduardo San Martin Morote](#).

## Installation

---

```
# Yarn
$ yarn vue-global-events
# NPM
$ npm install vue-global-events --save
```

## Usage

---

Now install the component in `main.js`.

`main.js`

```
import Vue from 'vue';
import App from './App.vue';
import GlobalEvents from 'vue-global-events';
```

```
Vue.component(GlobalEvents);
```

```
new Vue({
  el: '#app',
  render: h => h(App)
});
```

Now you can listen to all sorts of global events from any component, while still using Vue's clear and simple syntax. You don't even need to worry about deregistering events! You can also use `v-if` and related directives to enable and disable the handlers.

```
<template>
  <div>
    <GlobalEvents
      @blur="handleAppBlur"
      @focus="handleAppFocus"
      @keydown.ctrl.tab="handleTabChange"
    ></GlobalEvents>

  </div>
</template>

<script>
export default {
  methods: {
    handleAppFocus() {
      console.log(`I've been focused!`);
    },

    handleAppBlur() {
      console.log(`I'm feeling a bit blurry...`);
    },

    handleTabChange() {
      console.log(`**Doing Tab Changy Things**`);
    }
  }
}
</script>
```