

通过v-on 监听 和\$emit触发来实现:

- 1、在父组件中 通过v-on 监听 当前实例上的 自定义事件。
- 2、在子组件 中 通过'\$emit' 触发 当前实例上的 自定义事件。

### 示例:

父组件:

```
<template>
  <div class="fatherPageWrap">
    <h1>这是父组件</h1>
    <!-- 引入子组件, v-on监听自定义事件 -->
    <emitChild v-on:emitMethods="fatherMethod"></emitChild>
  </div>
</template>
<script type="text/javascript">
  import emitChild from '@/page/children/emitChild.vue';
  export default {
    data () {
      return {}
    },
    components : {
      emitChild
    },
    methods : {
      fatherMethod(params) {
        alert(JSON.stringify(params));
      }
    }
  }
</script>
```

子组件:

```
<template>
  <div class="childPageWrap">
    <h1>这是子组件</h1>
  </div>
</template>
<script type="text/javascript">
  export default {
    data () {
```

```
        return {}
    },
    mounted () {
        //通过 emit 触发
        this.$emit('emitMethods', {"name" : 123});
    }
}
</script>
```

结果:

子组件 会调用 父组件的`fatherMethod` 方法, 该并且会`alert` 传递过去的参数:

```
{"name":123}
```