

## Vue中组件间通信有哪几种方式

- 父子组件

- props/emit

子组件有时需要与父组件进行沟通，沟通的方式就是子组件 `$emit` 传递一个自定义事件，父组件通过监听这个事件来做进一步动作。而父组件与子组件通信则使用 `props`来传递一个绑定在子组件上的属性，这个属性值来源于父组件

- parent/children

- 父组件中

使用 `$children` 操作子组件。并在父组件通过 `$children` 访问到已经在父组件当中引入了的子组件，`$children` 返回的是子组件列表，是一个数组。

- 子组件中

子组件可通过 `$parent` 来访问父组件里的数据和定义的方法，如修改父组件中的 `$data`

- ref

父组件中可为子组件定义一个`ref`属性，然后父组件里通过`$ref[对象子组件上定义的属性名]`来访问子组件里的数据

- provide/inject

适用于祖先和后代关系的组件间的通信，祖先元素通过 `provide` 提供一个值，后代元素则通过 `inject` 获取到这个值。这个值默认是非响应的，如果是对象那么则是响应式的：

- 兄弟组件

- bus

在Vue . 2.x 中， 推荐使用一个空的Vue 实例作为中央事件总线（ bus ），也就是一个中介。通信的组件必须都引入这个实例。

通过 `bus.$emit()` 传递出一个自定义事件

在需要进行操作的组件里，通过 `bus.$on` 监听这个自定义事件, 进行操作

- Vuex

`vuex` 是Vue推出的状态管理工具

`vuex` 就是把需要共享的变量全部存储在一个对象里面，然后将这个对象放在顶层组件中供其他组件使用。这么说吧，将vue想作是一个js文件、组件是函数，那么vuex就是一个全局变量，只是这个“全局变量”包含了一些特定的规则而已。

通过 `$store` 这个对象访问数据和调用在 `vuex` 里定义的公共方法

- 跨级组件

- bus vuex provide inject