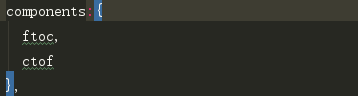
# VUE组件通信

## 父子组件通信

## （1）怎么定义是父子组件？

第一种：将其他组件以import引入用自定义标签接收，在当前组件中component里注册该标签，页面上可以直接用<自定义标签></自定义标签>样子使用。当前组件为父组件，被引入的组件为子组件。







第二种：在路由中定义好组件，组件中含有children，页面上通过<router-view></router-view>形式使用，也算是父子组件，当前组件为父组件，router-view路由的为子组件。





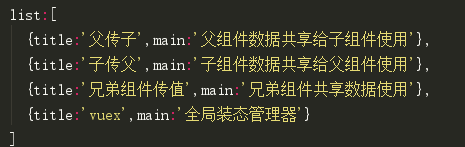
## 父组件的数据、方法传递给子组件

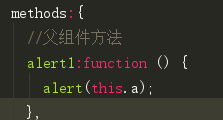
方式1：

在父组件里面，子组件以自定义标签，或者router-view的形式调用，在这些标签里通过:lists=’list’，:alert1=’alert1’，形式传递给子组件。:后面跟的是自己定义的参数名，后面子组件接收时用到。’’里面跟的是父组件里的数据，或者方法。

子组件中，用props:[‘lists’,’alert1’],形式接收父组件的数据和方法，注意接收使用的名字跟父组件自己定义的名字要一致，此时如果接收的方法用到了父组件的变量，在子组件中调用时，还是指向父组件的变量。

父组件中的数据和方法





在子组件自定义标签中声明传递的数据和方法





子组件props中接收传入的数据和方法。



调用父组件的数据和方法





方式2：

父组件基本同上，区别在于绑定的时候用v-on:或@而不是用:亲测只有:不起作用。

子组件中不使用props来接收父组件的方法，在methods中通过定义一个新方法，在方法体里面写this.$emit(‘alert1’,’canshu1’)，以这种形式获取父组件的方法，在赋值给子组件新定义的方法，注意alert1是父组件中绑定时候自己定义的名字，canshu1若该方法带参数则加上，不带则加空字符串即可。

父组件定义：



子组件在methods里定义新方法

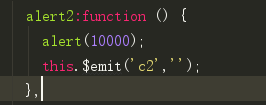


## 子组件的数据和方法传递给父组件

传递方法：

子组件定义好方法，父组件在引用子组件的时候在自定义标签中添加标识符，ref=”child1”，用来表示这个子组件。父组件在methods中定义自己的方法名，方法主体：this.$refs.child1.fun。意思是通过调用$refs中名字叫child1的子组件里的fun方法，赋值给父组件。

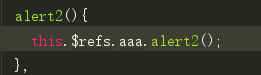
子组件中定义方法



父组件中引用子组件时，添加一个标识符rel=’aaa’,注意rel是语法，aaa是标识



父组件中定义一个方法，主体内容如图，表示方法就是标识为aaa的子组件里的alert2方法。



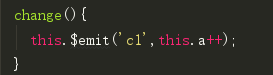
调用



传递数据：

子组件定义一个方法，方法中通过this.$emit(‘change’,’changeMsg’)方式发送数据changgeMsg，父组件中监听change事件，当事件触发时，直行函数，进行赋值。

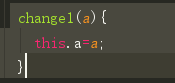
子组件定义方法



父组件监听方法事件触发时直行函数



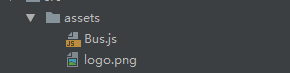
定义一个带参数的方法，调用时，让本组件的数据等于传递过来的参数，事件数据传递

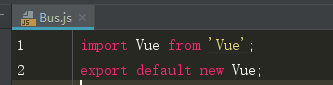


## 兄弟组件通信

方式1：

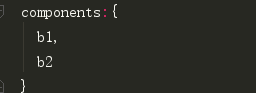
第一步：借助中央事件总线：在外部新建一个js文件，取名Bus.js，可放在assets文件夹目录下。

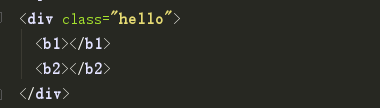




第二步：两个组件作为子组件被同一个父组件所引用，在同一个页面显示。



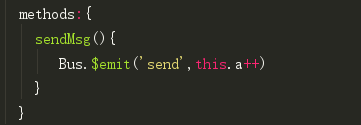




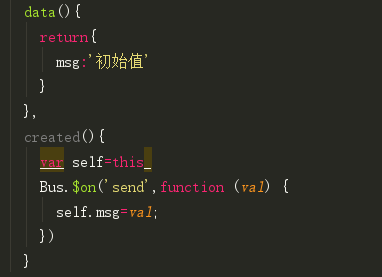
第三步：两个兄弟组件之间的语法

组件1需要定义一个发送数据的方法，此处sendMsg是定义在此的方法，页面点击时候触发此函数，函数主体，通过Bus中央事件总线用$emit发送一个send事件，事件的功能就是传送一个this.a++的数据，区分sendMsg和send，前者是当前页面的点击事件，后者是发送出去，需要其他组件监听的事件。



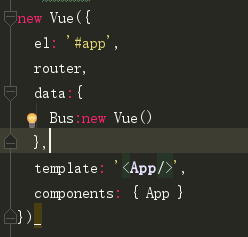


组件2在组件2事先定义一个变量，在生命周期为created的时候，通过Bus中央事件总线用$on监听组件1中发送的send事件，用一个带参数的回调函数，接收穿过来的值，参数即为传的值 ，把值赋值给当前组件的变量。

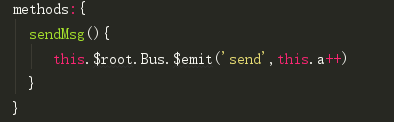


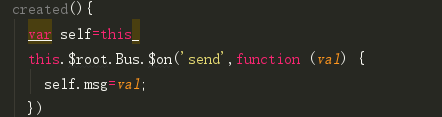
方式2：可以看做方式1的简化版

第一步：省略了外部的js文件，直接把总线放在main.js里的vue实例中，写法如图。



第二步：在使用的时候，不需要引入外部文件，只需要在Bus前加this.$root，即可调用。





## 页面跳转传值

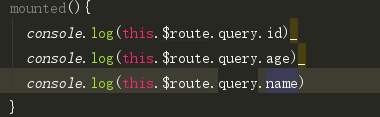
方式1：query传值在router-link标签内to的后面直接加



显示的效果如图：



怎么获取id值，



在全局已经存在这个变量，只需要按照图中所示，找到这个变量重新赋值即可。

注意：传值的时候会在url地址后面明码拼接所传的值，另外，在组件1中发送数据，到了组件2的url上才会拼接出来，所以，数据也只能在组件2中获取。

刷新不会清空

方式2：params传值：在router-link中加

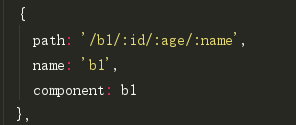


显示的效果

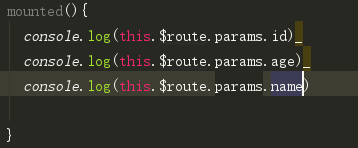


跟query的传值在写法上很相似，url的显示则完全不同，注意params有一个name属性，对应的是你需要传过去的那个组件的name名在路由中有定义，必须一致。

路由书写

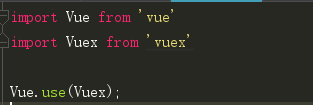


路由要定义好后面跟随的参数名



调用数据跟query类似，修改query为params。

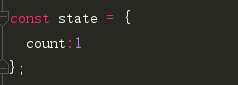
## VUEX

1. 安装
   1. Cnpm install vuex
   2. 在src文件夹下新建vuex文件夹，新建store.js文件中引入vue和vuex
   3. 注意：[引入vuex只需要直接引入，路径中不需要加其他./或者@或者../之类的](mailto:引入vuex只需要直接引入，路径中不需要加其他./或者@或者../之类的)，vuex引入后要主动使用Vue.use(Vuex)调用
   4. 在main中引入store.js文件在Vue实例中添加store对象
2. 什么是vuex
   1. 多个不同组件之间，需要用到同一个状态，当某一个组件中用户的输入导致状态改变，其他组件也要相应的改变。

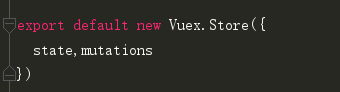
自我理解：类似全局作用域，所有的组件都能公用全局变量，当全局变量改变时，所以引用此变量的组件都相应改变。

1. 核心概念
2. state状态仓库

1.存放所有共享状态的仓库，语法规定，不能随意摆放。



在store.js文件中写。定义好之后需要export default导出

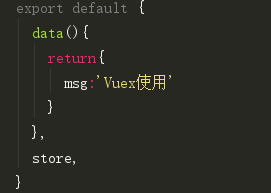


组件中调用

第一步要先import引入



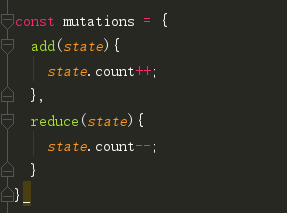
第二步要在vue中注册store



第三步html中使用时要加$store.state才能访问到这个变量



b) mutations 更改状态的唯一方法



同样是在store文件中定义对象，对象内存放的是修改状态值的方法，定义的时候要带state参数，固定写法，变量前要带state，同理写完后要export default导出

调用



Import引入注册后，页面即可调用，固定写法如图。

更多vuex相关资料

http://blog.csdn.net/h5\_queenstyle12/article/details/75386359