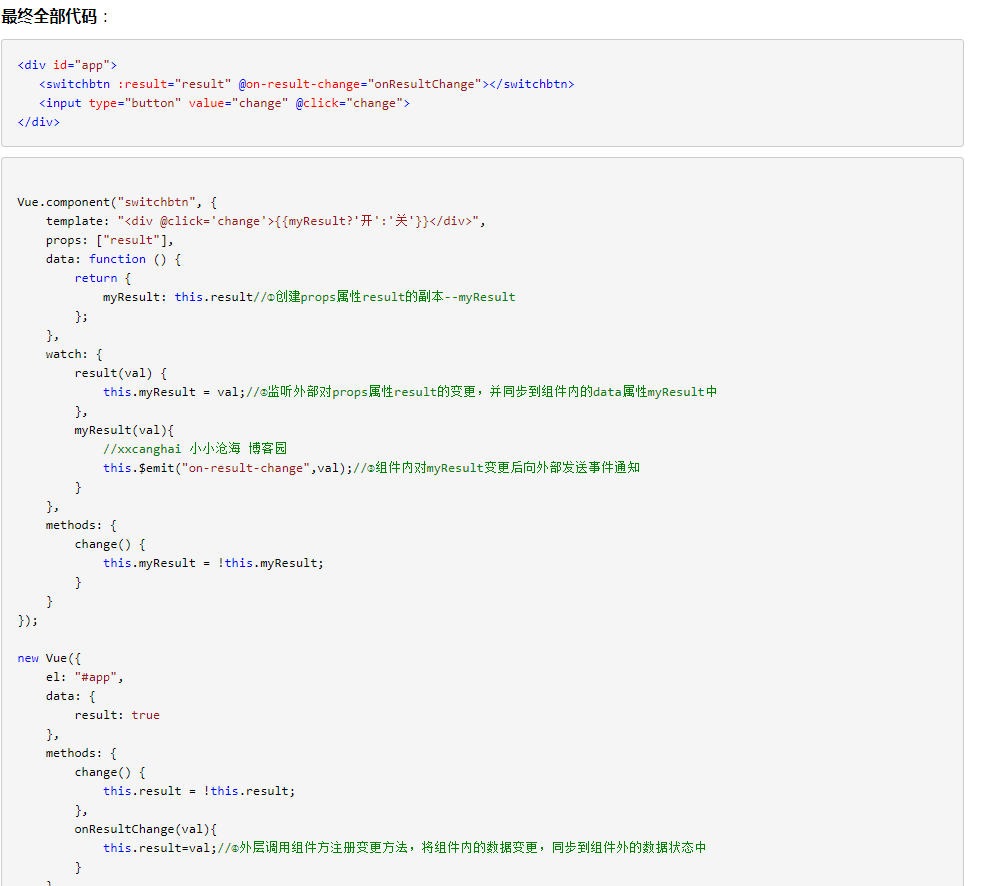
1. mock数据 没获取到 但dev-server.js 已经配置好了
2. 组件命名规范 载入组件标签的时候 不要用之前已有的标签名 可以用驼峰形势 或者用v-xxxx



1. Props双向传递数据

个人心得 （重要的很）

1. 引入外部组件时 要注意 第一步 在父组件里 import 引入 第二步 要在父组件中的components里引入调用 第三步 在父组件中的html中相应位置添加组件标签
2. Ref 相当于给个id 看官网 介绍的比较好 在父组件中 给组件一个ref=“food”标识

然后在触发的事件中 用

this.$nextTick(()=>{

this.$refs.food.show() // 调用指定子组件food的show方法

})

1. 组件间数据的传递 $emit $on 如果在一个页面的话 直接引用就可以

形式为 this.$emit(“给个名字当个标识”，data（传要传的值）)

接收在该组件上

<zujian v-on:”标识名”=“datfn（定义一个函数接收数据）”><zujian>

然后在methods中完善这个函数

datafn(data){

Console.log(data)

This.selectType = data;

}

但如果不是在一个页面的话 需要另外一种形式才能生效 这个也是最常用的方法

就是创建一个空vue对象 把它当作中间量来实现数据的传递

比如

新建了一个sell.js

Import Vue from “vue” //创建vue实例就要引入

Export var sell = new Vue()

然后把这个js 分别引入到要传递数据的组件文件下 形式是

Import { sell } from “xxxxxx/xxx/sell.js” // 因为要引用的是一个函数或方法 需要用{xxx}形式来引用

在子组件中

Sell.$emit(‘content.toggle’,this.onlyContent);

在父组件中

可以用钩子函数还赋值

Created(){

Sell.$on(“content.toggle”,(data)=>{

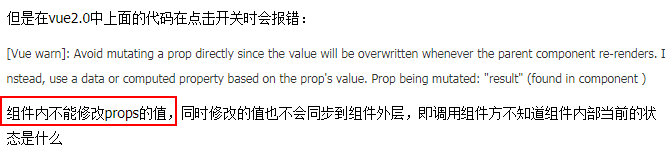
This.onlyContent = data;

Console.log(data);

})

}

4.这个很重要！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！



Props双向传递数据时 在子组件中 不要对需要传的参数进行修改 否则就会报上面的错

所以可以换另外一种思路 创建一个参数副本

data(){

return{

emitSelectType:this.selectType,//创建props属性需要双向传递的副本

emitonlyContent:this.onlyContent//创建props属性需要双向传递的副本

}

},

然后在方法中 或者需要用到需要的值得时候都用副本代替 然后把副本的值传给父组件但props是没有修改的就不会报错了 然后在父组件中就能获取改变的值了 相应的 赋值上就可以了

子组件

methods: {

select(type, event) {

if(!event.\_constructed) {

return;

}

this.emitSelectType = type;

sellEmit.$emit('ratingtype.select', this.emitSelectType);

}

父组件

created(){

sellEmit.$on("ratingtype.select",(data)=>{

this.selectType = data;

})

sellEmit.$on("content.toggle",(data)=>{

this.onlyContent = data;

})

}