## 2017-1-20

1、排序等组件内点击事件使用回调函数 研发自定义function传入

## 2017-1-22

1、封装时候注意对外暴露的方法和属性,有时候需要隐藏一些

2、通讯传递参数时候 使用一个大对象 把东西都放到里面传

这样处理可以增加参数时候不用改变 如下面的:参数=”值” 不用总添加

'<grid :tt="tt" :collist="collist" :datalist="datalist" :sortfun="sortfun"></grid>'

<grid :prop=”prop” ></grid>

3、外部js文件定义的方法传入到Vue组件内部的methods

**methods**: Object.assign(aaa,{  
 test: function (value) {  
 **console**.log(value)  
 alert('test')  
 }  
})

Index.html 内

var aaa = {}  
aaa.a = function ok (value) {  
 **console**.log(value)  
 alert(1)  
}  
aaa.b = function om () {  
 alert(2)  
**}**

## 2017-1-23

1. new vue的El属性的参数可以接收为dom对象

**el**: **div**[0],

## 2017-1-24

1. 自定义Component 封装

var acomponent = {  
 **template**: ['<a href="javascript:void(0)" @click="tdcbfun(click,colname)"> {{rowdata[colname]}}

</a>'].join(''),  
 **props**: {  
 **rowdata**: {},  
 **colname**: {},  
 **tdcbfun**: {  
 **type**: **Function**,  
 default: function (value) {  
 **window**.alert('这个方法是默认的td回调函数,请设置options.collist对象的cbfun!')  
 }  
 }  
 }  
}

<a href="javascript:void(0)" onclick="cb1(str1,str2)"> {{ str1 [str2]}}

</a>

封装要求:不想让研发人员使用@click=tdcbfun 都自己写 后面的变量随便输 位置相符就能取到值， {rowdata[colname] }也是

解决办法 首先 原来的tbcbfun 为 <component :tbcbfun=col.cbfun>

改为col.cbfun 由 stirng 改为 数组[] , 然后研发携程 onclick=clickcbfun(参数1,2)

程序改成 onclick替换成@click & clickcbfun 改成 col.cbfun[cb1]

2、点击某一行 前面单选框选择 数据来驱动

首先collist里增加一列叫checkbox 然后他是特殊的列 他有一个checkboxarray 改变这个就选中了

3、是否可以使用 addevent 和 fireevent 在mounted的时候fireevent

在pagea.js里addevent?

4、把el暴露出来