# Echarts组件使用手册

## 如何安装

Package.json:

**“dependencies”："echarts"**: **"^4.0.0"**,

**"zrender"**: **"^4.0.0",**

**"echarts-liquidfill": "^2.0.0"**

**"json-loader": "^0.5.7"**

**添加loader**

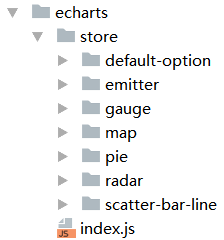
**{  
 test: /\.(json|geojson)$/,  
 use: {  
 loader: 'json-loader'  
 }  
}**

入口文件：

**import** Vechart **from '../pages/manage/echarts/index.js'**;

Vue.use(Vechart);

**../pages/manage/echarts/index.js ：是echarts组件的入口文件，以下是组件目录**

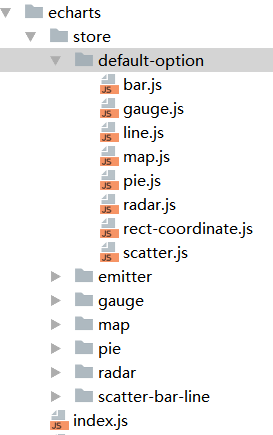


## 组件目录介绍

Index.js:组件入口文件，使用时需要import

Store：存放逻辑代码和默认配置的文件

Default-option: 存放echarts图表的默认配置文件



在同一项目下，一般echarts图表的风格保持一致，比如字体，颜色，背景色等。

这些配置如果每画一个图表就写一次，未免有些重复，因此开发者在项目开始前可以在默认配置里设置好基本风格样式，画具体图表时再通过组件属性细微调整。

Gauge，map，pie，radar：仪表图、地图、饼图、雷达图的组件代码

Scatter-bar-line: 散点图，柱状图，折线图的组件代码

Emitter：混合代码文件，放置组件的公共方法和属性。

## 组件使用介绍

### 饼图

**组件名: <ve-pie></ve-pie>**

|  |  |  |  |
| --- | --- | --- | --- |
| **属性** | **说明** | **类型** | **默认值** |
| **id** | **图表的id，必填** | **String** | **无** |
| **rendererType** | **渲染器类型**  **可选:svg/canvas** | **String** | **canvas** |
| **title** | **图表标题** | **String** | **无** |
| **backgroundColor** | **图表背景颜色** | **String** | **白色:#fff** |
| **Color** | **调色盘颜色** | **Array** | **Echarts图表默认调色盘** |
| **showLegend** | **是否展示legend** | **布尔值** | **true** |
| **center** | **饼图的中心（圆心）坐标，数组的第一项是横坐标，第二项是纵坐标。支持设置成百分比，设置成百分比时第一项是相对于容器宽度，第二项是相对于容器高度。** | **Array** | **['50%','50%']** |
| **radius** | **饼图的半径，数组的第一项是内半径，第二项是外半径,支持设置成百分比** | **Array** | **[0,'75%']** |
| **rippleAnimation** | **是否开启涟漪动画,布尔值(true/false)，默认false** | **Boolean** | **false** |
| **rippleOption** | **涟漪动画的配置项** | **Object** | **？待完善** |
| **coverOption** | **自定义覆盖配置** | **Object** | **null** |
| **data** | **饼图数据,默认格式：**  **[**  **{name:name1,value:val1}，{name:name2,value:val2}...**  **]** | **Array** | **空: []** |

**数据格式举例：**

**pieData:**  
[ {**value**:335, **name**:**'aaaa'**},  
 {**value**:310, **name**:**'bbbb'**},  
 {**value**:274, **name**:**'cccc'**},  
 {**value**:235, **name**:**'dddd'**},  
 {**value**:400, **name**:**'eeee'**}  
]

**coverOption: 如果对于组件暴露的属性不满意，觉得不够详细，无法满足需求，则可以使用**

**coverOption使用自定义的optinon对象，该设置优先级最高，如果设置了coverOption，则里面的属性会覆盖静态文件和通过接口设置的相同属性。**

**例子： 在coverOption里设置了title标题为’标题1’,通过暴露出来的属性设置title为’标题2’,则最终结果以coverOption里面的为准，为标题1**

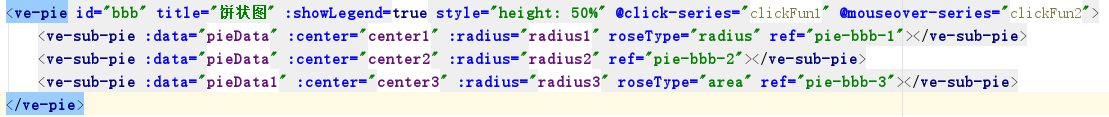
**let** lastOption = *$*.extend(**true**,{},option , coverOption);*//true开启深拷贝*

Option：为静态配置，dataOption为设置属性后生成的配置, coverOption为自定义的option。

**组件名: <ve-sub-pie></ve-sub-pie>，当一个图表内想画多个饼图时可用**

|  |  |  |  |
| --- | --- | --- | --- |
| **属性** | **说明** | **类型** | **默认值** |
| **roseType** | **是否展示成南丁格尔图，通过半径区分数据大小。可选择两种模式:**  **radius' 扇区圆心角展现数据的百分比，半径展现数据的大小。 \* 'area' 所有扇区圆心角相同，仅通过半径展现数据大小。** | **String** | **空字符串** |
| **center** | **饼图的中心（圆心）坐标，数组的第一项是横坐标，第二项是纵坐标。** | **Array** | **['50%','50%']** |
| **radius** | **饼图的半径，数组的第一项是内半径，第二项是外半径,支持设置成百分比** | **Array** | **[0,'75%']** |
| **data** | **饼图数据,默认格式：**  **[**  **{name:name1,value:val1}，{name:name2,value:val2}...**  **]** | **Array** | **空: []** |

*使用举例:*

**

**注意：**

如果在<ve-pie>和<ve-sub-pie>中同时设置了data，则只有<ve-sub-pie>中的data会生效。

若在父子组件中同时出现相同属性配置，以子组件中为主。

### 直角坐标系图表

由于折线图、柱状图、散点图都是基于直角坐标系，因此将这种图表都放在scatter-bar-line文件夹内。

使用时，必须先写<ve-rect-coordinate></ve-rect-coordinate>标签，再在里面添加

柱状图：<ve-bar> 折线图：<ve-line> 散点图：<ve-scatter>

之所以这么设计，是为了能将一些坐标轴的公用属性分配到同一个标签上，以免每个图表标签的属性太多。

使用例子：

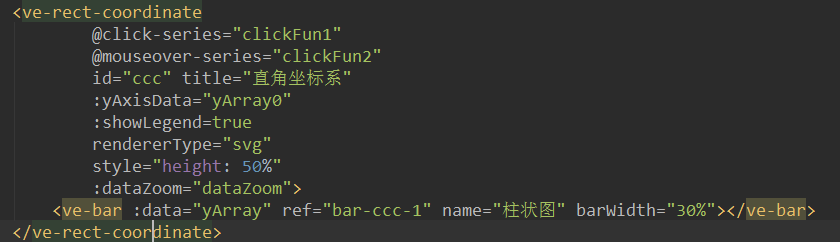


**<ve-rect-coordinate></ve-rect-coordinate>**

**Props**

|  |  |  |  |
| --- | --- | --- | --- |
| **属性** | **说明** | **类型** | **默认值** |
| **id** | **图表的id，必填** | **String** | **无** |
| **rendererType** | **渲染器类型**  **可选:svg/canvas** | **String** | **canvas** |
| **title** | **图表标题** | **String** | **无** |
| **backgroundColor** | **图表背景颜色** | **String** | **白色:#fff** |
| **color** | **调色盘颜色** | **Array** | **Echarts图表默认调色盘** |
| **showLegend** | **是否展示legend** | **布尔值** | **true** |
| **dataZoom** | **区域缩放组件** | **Object**  **{x: true,y: false},显示x轴上的区域缩放，y轴上的区域缩放不显示** | {  **x**: **false**,  **y**: **false** } |
| **coverOption** | **自定义覆盖配置** | **Object** | **null** |
| **xAxisData** | **x轴刻度数据**  **可以是离散的类目数据，也可以是连续的数值，组件会根据数组第一项的数据类型判断是‘**category**’还是‘**value**’** | **Array** | **空：[]** |
| **yAxisData** | **y轴刻度数据**  **可以是离散的类目数据，也可以是连续的数值，组件会根据数组第一项的数据类型判断是‘**category**’还是‘**value**’** | **Array** | **空：[]** |

**注意：当你想画横向柱状图时，可以设置yAxisData为类目类的数组；想画纵向柱状图时，设置xAxisData为类目类数组。**

****

**yArray0: ['周一', '周二', '周三', '周四', '周五', '周六', '周日']**

**yArray: [10, 52, 200, 334, 390, 330, 220]**

**显示结果如下：**

****

Events

|  |  |  |
| --- | --- | --- |
| 事件名 | 说明 | 返回值 |
| **click-series** | 点击到echarts图的series上 | 包含点击图形数据信息的对象params |
| **mouseover-series** | 鼠标移动到echarts图的series上 | 包含点击图形数据信息的对象params |

所有的鼠标事件包含参数 params，这是一个包含点击图形的数据信息的对象，格式如下：

{

// 当前点击的图形元素所属的组件名称，

// 其值如 'series'、'markLine'、'markPoint'、'timeLine' 等。

componentType: **string**,

// 系列类型。值可能为：'line'、'bar'、'pie' 等。当 componentType 为 'series' 时有意义。

seriesType: **string**,

// 系列在传入的 option.series 中的 index。当 componentType 为 'series' 时有意义。

seriesIndex: number,

// 系列名称。当 componentType 为 'series' 时有意义。

seriesName: **string**,

// 数据名，类目名

name: **string**,

// 数据在传入的 data 数组中的 index

dataIndex: number,

// 传入的原始数据项

data: Object,

// sankey、graph 等图表同时含有 nodeData 和 edgeData 两种 data，

// dataType 的值会是 'node' 或者 'edge'，表示当前点击在 node 还是 edge 上。

// 其他大部分图表中只有一种 data，dataType 无意义。

dataType: **string**,

// 传入的数据值

**value**: number|Array

// 数据图形的颜色。当 componentType 为 'series' 时有意义。

color: **string**

}

}

}

}

});

<ve-bar>

|  |  |  |  |
| --- | --- | --- | --- |
| **属性** | **说明** | **类型** | **默认值** |
| **Name** | **系列名称, 用于tooltip的显示，legend的图例** | **Sting** | **无** |
| **barWidth** | **柱条的宽度** | **String** | **‘自适应’** |
| **barMaxWidth** | **柱条的最大宽度** | **String** | **‘自适应’** |
| **barMinHeight** | **柱条的最小高度** | **Number** | **0** |
| **barGap** | **柱间距离，可设固定值（如 20）或者百分比（如 '30%'，表示柱子宽度的 30%）。** | **String** | **‘30%’** |
| **gradientDirection** | **渐变方向, 有四种选择: top-bottom bottom-top left-right right-left**  **如果不是这四种之一，就不展示渐变** | **String** | **‘null’** |
| **linearGradient** | **柱状图渐变配置** | **Array** | **[{  offset: 0,  color: '#00feff'  },  {  offset: 0.5,  color: '#027eff'  },  {  offset: 1,  color: '#0286ff'  } ]** |
| **data** | **数据数组** | **Array,每一项为Number类型** | **空：[]** |

<ve-line>

|  |  |  |  |
| --- | --- | --- | --- |
| **属性** | **说明** | **类型** | **默认值** |
| **Name** | **系列名称, 用于tooltip的显示，legend的图例** | **Sting** | **无** |
| **Symbol** | **标记的图形** | **String,**  **可选项：**  **’circle’，‘rect’，**  **’roundRect’，’triangle’，'diamond', 'pin', 'arrow'** | **‘emptyCircle’** |
| **symbolSize** | **标记的大小，可以设置成诸如 10 这样单一的数字，也可以用数组分开表示宽和高，例如 [20, 10] 表示标记宽为20，高为10。****如果需要每个数据的图形大小不一样，可以设置为如下格式的回调函数：**  **(value:Array|number,params:Object)=>number|Array**  **其中第一个参数 value 为**[**data**](http://echarts.baidu.com/option.html#series-.data)**中的数据值。第二个参数params 是其它的数据项参数。** | [***String***,Array,**Function**] | **4** |
| **step** | **是否是阶梯线图** | **Boolean** | **false** |
| **smooth** | **是否平滑曲线显示** | **Boolean** | **false** |
| **data** | **数据数组** | **Array,每一项为Number类型** | **空：[]** |

<ve-scatter>

|  |  |  |  |
| --- | --- | --- | --- |
| **属性** | **说明** | **类型** | **默认值** |
| **Name** | **系列名称, 用于tooltip的显示，legend的图例** | **Sting** | **无** |
| Symbol | **标记的图形** | **String,**  **可选项：**  **’circle’，‘rect’，**  **’roundRect’，’triangle’，'diamond', 'pin', 'arrow'** | **‘circle’** |
| **symbolSize** | **标记的大小，可以设置成诸如 10 这样单一的数字，也可以用数组分开表示宽和高，例如 [20, 10] 表示标记宽为20，高为10。如果需要每个数据的图形大小不一样，可以设置为如下格式的回调函数：**  **(value:Array|number,params:Object)=>number|Array**  **其中第一个参数 value 为**[**data**](http://echarts.baidu.com/option.html#series-.data)**中的数据值。第二个参数params 是其它的数据项参数。** | [***String***,Array,**Function**] | **4** |

### 地图

<ve-map>

|  |  |  |  |
| --- | --- | --- | --- |
| **属性** | **说明** | **类型** | **默认值** |
| **id** | **图表的id，必填** | **String** | **无** |
| **rendererType** | **渲染器类型**  **可选:svg/canvas** | **String** | **canvas** |
| **title** | **图表标题** | **String** | **无** |
| **backgroundColor** | **图表背景颜色** | **String** | **白色:#fff** |
| **placeName** | **地图名字：省份对应拼音，中国对应china，世界对应world** | **String** | **china** |
| **backgroundColor** | **地图背景颜色** | **String** | **‘#000’** |
| **normalAreaColor** | **静态地图块颜色** | **String** | **‘#323c48’** |
| **emphasisAreaColor** | **动态地图块颜色** | **String** | **‘darkorange’** |
| **normalBorderColor** | **静态地图块边界颜色** | **String** | **‘dodgerblue’** |
| **normalLabel** | **静态地图图形上的文本标签，可设置字体样式** | **Object** | **{ textStyle: { color: '#999', fontWeight:'normal', fontFamily:'Microsoft YaHei', fontSize: 13}, show: true }** |
| **emphasisLabel** | **动态地图图形上的文本标签，可设置字体样式** | **Object** | **{ textStyle: { color: '#fff', fontWeight:'normal', font Family:'Microsoft YaHei', fontSize: 13}, show: true }** |
| **scatterData** | **要在地图上画带有涟漪特效动画的散点（气泡）图传入的数据** | **Array,**  **例子：**  **[  {name:'福州',value:95},  {name:'太原',value:90},  {name:'长春',value:80}]** | **[]** |
| **scatterColor** | **涟漪特效动画的颜色** | **String** | **‘#F4E925’** |
| **scatterSymbolSize** | **涟漪特效动画的点的尺寸, 可以设置成诸如 10 这样单一的数字，也可以用数组分开表示宽和高，例如 [20, 10] 表示标记宽为20，高为10. 如果需要每个数据的图形大小不一样，可以设置为如下格式的回调函数：**  **(value:Array|number,params:Object)=>number|Array**  **其中第一个参数 value 为**[**data**](http://echarts.baidu.com/option.html#series-.data)**中的数据值。第二个参数params 是其它的数据项参数。** | **[String,Array,Function]** | **(val) => { return val[2] / 8;}** |
| **brushType** | **涟漪特效波纹绘制方式, 可选'stroke' 和 'fill'** | **String** | **‘fill’** |
| **lineData** | **要在地图上画类带有起点和终点信息的线数据时传入的数据** | **Array,例子：**  **[  [{name:'广州'},{name:'福州'}],  [{name:'广州'},{name:'太原'}],  [{name:'广州'},{name:'长春'}]]** | **[]** |
| **lineColor** | **线数据动画的颜色** | **String** | **‘#fff’** |
| **lineSymbol** | **线数据标记类型包括** | **String** | **'circle'** |
| **lineSymbolSize** | **线数据特效标记的大小** | **Number** | **3** |
| **linePeriod** | **线数据特效动画的时间，单位为 s** | **Number** | **6** |
| **lineTrailLength** | **线数据特效尾迹的长度。取从 0 到 1 的值，数值越大尾迹越长** | **Number** | **0.7** |
| **coverOption** | **自定义覆盖配置** | **Object** | **null** |

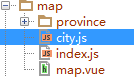
地图组件的目录结构：

Province：所有省份地图、中国地图、世界地图的json格式数据

city.js: 全国主要城市坐标文件

map.vue: 组件代码

index.js: 组件入口文件



使用例子：

<**ve-map id="eee"  
 style="height**: 100%**"  
 title="地图"  
 placeName="china"  
 :scatterData="mapData2"  
 :lineData="mapData1"  
 scatterColor="green"  
 brushType="stroke"** >  
</**ve-map**>

显示中国地图，改变placeName为zhejiang，即显示浙江省地图。

### 仪表盘

<ve-gauge>

|  |  |  |  |
| --- | --- | --- | --- |
| **属性** | **说明** | **类型** | **默认值** |
| **id** | **图表的id，必填** | **String** | **无** |
| **rendererType** | **渲染器类型**  **可选:svg/canvas** | **String** | **canvas** |
| **title** | **图表标题** | **String** | **无** |
| **backgroundColor** | **图表背景颜色** | **String** | **白色:#fff** |
| **center** | **仪表盘的中心（圆心）坐标，数组的第一项是横坐标，第二项是纵坐标。支持设置成百分比，设置成百分比时第一项是相对于容器宽度，第二项是相对于容器高度** | **Array** | **['50%', '50%']** |
| **radius** | **仪表盘半径，可以是相对于容器高宽中较小的一项的一半的百分比，也可以是绝对的数值。** | **[String,Number]** | **'70%'** |
| **dataRange** | **仪表盘数据范围数组[min,max],分别映射到startAngle,endAngle** | **Array** | **[10,100]** |
| **startAnger** | **仪表盘起始角度。圆心 正右手侧为0度，正上方为90度，正左手侧为180度** | **Number** | **225** |
| **endAngle** | **仪表盘结束角度** | **Number** | **-45** |
| **clockwise** | **仪表盘刻度是否是顺时针增长，顺时针是从min到max还是max到min。** | **Boolean** | **True** |
| **splitNumber** | **仪表盘刻度的分割段数,即数据从min到max被分割为几段** | **Number** | **10** |
| **coverOption** | **自定义覆盖配置** | **Object** | **Null** |
| **data** | **仪表盘数据** | **Array** | **[]** |

Events

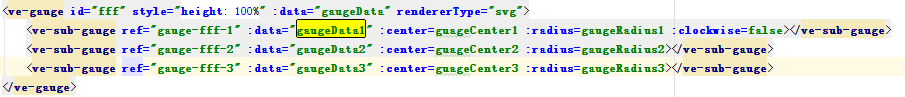
|  |  |  |
| --- | --- | --- |
| 事件名 | 说明 | 返回值 |
| **click-series** | 点击到echarts图的series上 | 包含点击图形数据信息的对象params |
| **mouseover-series** | 鼠标移动到echarts图的series上 | 包含点击图形数据信息的对象params |

*添加多个仪表图时可用*

<**ve-sub-gauge**>

|  |  |  |  |
| --- | --- | --- | --- |
| **center** | **仪表盘的中心（圆心）坐标，数组的第一项是横坐标，第二项是纵坐标。支持设置成百分比，设置成百分比时第一项是相对于容器宽度，第二项是相对于容器高度** | **Array** | **['50%', '50%']** |
| **radius** | **仪表盘半径，可以是相对于容器高宽中较小的一项的一半的百分比，也可以是绝对的数值。** | **[String,Number]** | **'70%'** |
| **dataRange** | **仪表盘数据范围数组[min,max],分别映射到startAngle,endAngle** | **Array** | **[10,100]** |
| **startAnger** | **仪表盘起始角度。圆心 正右手侧为0度，正上方为90度，正左手侧为180度** | **Number** | **225** |
| **endAngle** | **仪表盘结束角度** | **Number** | **-45** |
| **clockwise** | **仪表盘刻度是否是顺时针增长，顺时针是从min到max还是max到min。** | **Boolean** | **True** |
| **splitNumber** | **仪表盘刻度****的分割段数,即数据从min到max被分割为几段** | **Number** | **10** |
| **data** | **仪表盘数据, 默认格式：[{value: 0.5, name:** **'gas'},{value: 0.6, name: 'gas'}]**  **例子：**  **[{value: 30, name: 'km/h'}]**  **数组里多个项时name保持一致。** | **Array** | **[]** |

若在父子组件中同时出现相同属性配置，以子组件中为主。



### 雷达图

<ve-radar>

|  |  |  |  |
| --- | --- | --- | --- |
| **属性** | **说明** | **类型** | **默认值** |
| **id** | **图表的id，必填** | **String** | **无** |
| **rendererType** | **渲染器类型**  **可选:svg/canvas** | **String** | **canvas** |
| **title** | **图表标题** | **String** | **无** |
| **backgroundColor** | **图表背景颜色** | **String** | **白色:#fff** |
| **showLegend** | **是否展示legend** | **Boolean** | **true** |
| **color** | **调色盘颜色** | **Array** | **Echarts图表默认调色盘** |
| **indicator** | **雷达图的指示器，用来指定雷达图中的多个维度**  **,举例：**  **[  { name: '销售（sales）', max: 6500},  { name: '管理（Administration）', max: 16000, color: 'red'}, // 标签设置为红色  { name: '信息技术（Information Techology）', max: 30000},  { name: '客服（Customer Support）', max: 38000},  { name: '研发（Development）', max: 52000},  { name: '市场（Marketing）', max: 25000} ]** | Array | **[]** |
| **data** | **雷达数据,默认格式：[{name:'name',value:[va11,val2,val3,val4,val5,val6,]}，{name:'name',value:[va11,val2,val3,val4,val5,val6,]}...]**  **举例：  [{ value : [4300, 10000, 28000, 35000, 50000, 19000],  name : '预算分配（Allocated Budget）'  },{  value : [5000, 14000, 28000, 31000, 42000, 21000],  name : '实际开销（Actual Spending）'  }]** | **Array** | **[]** |
| **radius** | **雷达图的半径** | **[Number,String]** | **'50%'** |
| **center** | **雷达图的中心（圆心）坐标，数组的第一项是横坐标，第二项是纵坐标。支持设置成百分比，设置成百分比时第一项是相对于容器宽度，第二项是相对于容器高度。** | **Array** | **['50%','50%']** |
| **startAngle** | **坐标系起始角度，也就是第一个指示器轴的角度。** | **Number** | **90** |
| **splitNumber** | **指示器轴的分割段数** | **Number** | **4** |
| **shape** | **雷达图绘制类型, 支持 'polygon' 和 'circle'** | **String** | **'polygon'** |
| **coverOption** | **自定义覆盖配置** | **Object** | **[]** |

Events

|  |  |  |
| --- | --- | --- |
| 事件名 | 说明 | 返回值 |
| **click-series** | 点击到echarts图的series上 | 包含点击图形数据信息的对象params |
| **mouseover-series** | 鼠标移动到echarts图的series上 | 包含点击图形数据信息的对象params |

*添加多个雷达时可用*

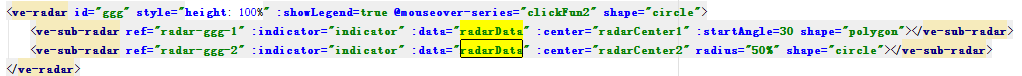
<**ve-sub-radar>**

|  |  |  |  |
| --- | --- | --- | --- |
| **radius** | **雷达图的半径** | **[Number,String]** | **'50%'** |
| **center** | **雷达图的中心（圆心）坐标，数组的第一项是横坐标，第二项是纵坐标。支持设置成百分比，设置成百分比时第一项是相对于容器宽度，第二项是相对于容器高度。** | **Array** | **['50%','50%']** |
| **startAngle** | **坐标系起始角度，也就是第一个指示器轴的角度。** | **Number** | **90** |
| **splitNumber** | **指示器轴的分割段数** | **Number** | **4** |
| **shape** | **雷达图绘制类型, 支持 'polygon' 和 'circle'** | **String** | **'polygon'** |
| **indicator** | **雷达图的指示器，用来指定雷达图中的多个维度**  **,举例：**  **[  { name: '销售（sales）', max: 6500},  { name: '管理（Administration）', max: 16000, color: 'red'}, // 标签设置为红色  { name: '信息技术（Information Techology）', max: 30000},  { name: '客服（Customer Support）', max: 38000},  { name: '研发（Development）', max: 52000},  { name: '市场（Marketing）', max: 25000} ]** | Array | **[]** |
| **data** | **雷达数据,默认格式：[{name:'name',value:[va11,val2,val3,val4,val5,val6,]}，{name:'name',value:[va11,val2,val3,val4,val5,val6,]}...]**  **举例：  [{ value : [4300, 10000, 28000, 35000, 50000, 19000],  name : '预算分配（Allocated Budget）'  },{  value : [5000, 14000, 28000, 31000, 42000, 21000],  name : '实际开销（Actual Spending）'  }]** | **Array** | **[]** |

注意：

若在父子组件中同时出现相同属性配置，以子组件中为主。

例子：



## 组件使用优势

1. 使用组件，无需用户再去初始化通过document.getElementById获得的dom节点，即屏蔽了dom操作，符合vue一贯隔绝dom操作的风格。
2. 将初始化chart对象，setOption, 图表自适应resize()，监听数据变化改变图表这些写每个图表都需要的重复操作封装，减少重复劳动。
3. 写业务的vue组件时，使用标签化的echarts组件可以简化页面代码，利于维护。