组件使用说明文档

版本记录

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 时间 | 修改人 | 修改内容 |
| 1.0 | 20180321 | 冯傲 | 创建文档，添加了侧边栏说明 |
| 1.0.1 | 20180321 | 张冲 | 新增<柱状图>,<表格>,<折线图> |
| 1.0.2 | 20180321 | 张冲 | <表格>组件增加方法**refresh-data** |
| 1.0.3 | 20180322 | 张冲 | <表格>组件增加方法**click-tablecell** |
| 1.0.4 | 20180323 | 冯傲 | 增加饼状图说明 |

目录

侧边栏 3

柱状图 5

折线图 6

表格(tables) 7

侧边栏

### 1 概述

提供了侧边栏，并且提供了选择筛选目标的功能，调用者需要提供一个回调函数，本组件会通过回调函数方式通知使用者。

### 2 使用方式

1. 引入组件

import sidebar from '@/components/common/sidebar/sidebar'

1. 在<template></template>中使用 <sidebar />标签即可引用该组件
2. 组件的定位应该采用绝对定位 fixed
3. 通过传入 props 方式传入回调函数 :propClickListener="..."
4. 或者在生命周期函数mounted: 中获取侧边栏vue组件实例，然后直接给组件的dataClickListener 字段赋值一个回调函数即可
5. 4和5只能同时使用一个，假如同时使用4方式生效 5方式失效
6. 现在回调函数只会返回一个点击的字符串，具体返回值后期再商榷

### 3 使用示例

1. 通过props 方式传入回调函数

<template>

<sidebar style="position: fixed;left: 0px;width: 173px;top: 91px;bottom: 35px" :propClickListener="clickListener"

/>

</template>

<script>

import sidebar from '@/components/common/sidebar/sidebar'

export default {

name: "demo",

methods: {

clickListener: (text) => {

console.log(text);

}

}

}

</script>

1. 使用手动赋值方式

<template>

<sidebar style="position: fixed;left: 0px;width: 173px;top: 91px;bottom: 35px"

ref="sidebar"/>

</template>

<script>

import sidebar from '@/components/common/sidebar/sidebar'

export default {

name: "demo",

mounted: function () {

var vueComp = this.$refs.sidebar;

vueComp.dataClickListener = function (text) {

alert(text);

};

}

}

</script>

柱状图

### 1 参数(Attributes)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | | 类型 | 举例 | 说明 |
| options  (必传) | id  (必传) | String | **id = ‘first’** | 表示当前图片的唯一标识 |
| categoryArr  (必传) | Array | [**'网易'**,**'腾讯网'**,**'百度'**,**……]** | 标识当前图标的类目列表 |
| chartHeight  (必传) | String/Number | **chartHeight = 300** | 表示当前图片的高度 |
| direction  (选传) | String | **direction = ‘horizontal’** | 参数表示当前柱状图的展示方向  可选值: horizontal/vertical |
| title  (选传) | String | **title =** **‘我的柱状图表’** | 表示柱状图名称 |
| showLegend  (选传) | Boolean | **showLegend = ‘true’** | 表示是否显示图例控件 |
| valueArr | Array | [  {  **name**: **'首页'**,  **data**: [1,3,5,7,9]  },  ………… ] | 表示当前图标的数据  数组的length表示用于展示的数据的组数 |
| (待开发) | colorArr  (选填) | Array | [**‘#fff’**,**’#000’**,……] | 表示使用的颜色集合 |

### 2 方法(Methods)

### 3 其他

折线图

### 1 参数(Attributes)

### 2 方法(Methods)

### 3 其他

表格(tables)

### 1 参数(Attributes)

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 举例 | 说明 |
| data  (必传) | Array | [  {  **platform**: **'看苏州App'**,  **title**:**'携手2018年世界杯蒙牛拥抱新时代'**,  **reprints**: **'97'**,  **comments**: **'97'**,  **readings**: **'103'**,  **praise**: **'97'**,  **time**: **'2017-12-22 15:51'**,  **author**: **'GG'**  **……** }  **……**  [ | 参数搭配columnData参数使用 |
| columnData  (必传) | Array | [  {**prop**: **'platform'**, **label**: **'平台'**},  {**prop**: **'title'**, **label**: **'文章标题'**},  {**prop**: **'reprints'**, **label**: **'转载数'**},  {**prop**: **'comments'**, **label**: **'评论数'**},  {**prop**: **'readings'**, **label**: **'阅读数'**},  {**prop**: **'praise'**, **label**: **'点赞数'**},  {**prop**: **'time'**, **label**: **'发布时间'**},  {**prop**: **'author'**, **label**: **'发布者'**}, ] | 1.参数代表呈现多少列  2.通过此参数从data参数中摘取需要呈现的数据  3.prop的值表示获取数据的标识,需要和data中的key对应  4.label的值表示当前列表头显示的文本 |
| columnStyle  (选传) | Object | {  **indexs**: [1,……],  **styles**: [  {  **color**: **'#ff0000'**,  **textAlign**:**'center'**  },  ……  ],  **commons:{}** } | 1.indexs表示要修改的列的index  2.styles表示要将该列增加的样式  3.commons表示该列的公共样式 |
| rowStyle  (选传) | Object | {  **indexs**: [0,…],  **styles**: [  {  **color**: **'#ff0000'**,  **textAlign**:**'center'**  },  ……  ],  **commons:{}** } | 1.indexs表示要修改的行的index  2.styles表示要将该行增加的样式  3.commons表示该行的公共样式 |
| headerStyle  (选传) | Object | {  **indexs**: [1,…],  **styles**: [  {  **color**:**'#ff0000'**,  **textAlign**:**'center'**  **backgroundColor:’#333’**  },  ……  ],  **commons:{}** } | 1.indexs表示要修改的表头单元格中单元格的index  2.styles表示要将该单元格增加的样式  3.commons表示头部的公共样式 |
| showBorder  (选传) | Boolean | **showBorder** = **‘true’**  s**howBorder** = **‘false’** | 表示是否带有纵向边框  默认:false |
| showHeader  (选传) | Boolean | **showHeader** = **‘true’**  **showHeader** = **‘false’** | 表示是否显示表头  默认:true |
| highlightCurrentRow  (选传) | Boolean | **highlightCurrentRow** = **‘true’**  **highlightCurrentRow** = **‘false’** | 表示是否要高亮当前行  默认false |
| tableType  (选填) | Object | {  **indexs**:[],  **types**:[],  **commons**:**'selection'**,  **position**:**’leading’** } | 1.表示表格类型  2. types可选值为selection/index/expand  3.commons设置统一type  4.position表示出现的位置,可选值为leading(头)/trailing(尾)  默认:无 |

### 2 方法(Methods)

|  |  |  |
| --- | --- | --- |
| 方法名 | 说明 | 参数 |
| **refresh-data** | 当表格进行了筛选操作,更改容量操作,更改当前页操作,排序操作时候都会调用这个方法用于表格数据更新 | selected(筛选项),  currentPage(当前页),  pageSize(容量),  sortingIndex(排序) |
| **click-tablecell** | 当点击了当前表格内容时候会调用此方法,用于进行跳转操作或者其他自定义操作 | **rowIndex(行号)**  **columnIndex(列号)** |

### 3 其他

# 饼状图

### 1 使用方式

1. 引入组件

import pieCharts from '@/components/common/pieCharts.vue'

1. 传入参数

通过传入 props 方式传入回调函数 :pieData={}

1. 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 含义 | 是否必须 | 类型 |
| pieIdName | 控件id 具有唯一性 不可以存在相同的id | Y | string |
| pieName | 饼图名称 | Y | string |
| pieFormatterFun | 控制文字格式，可以配合pieRich使用富文本 | N | fun |
| pieRich | 富文本格式 | N | object |
| radius | 控制饼图大小，有两个参数一个代表内圆大小一个代表外圆, | N | string[] |
| center | 控制饼图位置 | N | string[] |
| data | 饼图数据源 | Y | object |
| pieColor | 色盘 | N | string[] |

### 2 示例

<template>

<piecharts :pieData="getOriginalPercentA()"/>

</template>

import pieCharts from '@/components/common/pieCharts.vue'

export default {

name: 'hello',

components: {

'piecharts': pieCharts,

},

data() {

},

methods: {

getOriginalPercentA: function () {

return {

pieIdName: 'originalPercentA',

pieName: '原创与非原创',

pieFormatterFun: (params, ticket, callback) => {

return '{black|' + params.name + '：}{black|' + params.value + '篇}\n{blue|' + params.percent + '%}';

},

pieRich: {

black: {

align: 'center',

fontSize: 14,

padding: [10, 0],

color: '#444444'

},

blue: {

align: 'center',

fontSize: 16,

padding: [8, 0, 0, 0],

fontWeight: 'bold',

}

},

data: [{

value: 103,

name: '原创'

}, {

value: 96,

name: '非原创'

}],

pieColor: ['#4642ff', '#00c6ff']

}

}

}

}

</script>

效果

