@Uway/UVUE 1.0.0

# 基础知识

## 安装node.js

### 安装包及源码下载地址

<https://nodejs.org/en/download/>



### Windows 安装

步骤 1 :  双击安装包：



步骤 2 : 点击以上的Run(运行)，将出现如下界面：



步骤 3 : 勾选接受协议选项，点击 next（下一步） 按钮 :



步骤 4 : Node.js默认安装目录为 "C:\Program Files\nodejs\" , 你可以修改目录，并点击 next（下一步）：



步骤 5 : 点击树形图标来选择你需要的安装模式 , 然后点击下一步 next（下一步）



步骤 6 :点击 Install（安装） 开始安装Node.js。你也可以点击 Back（返回）来修改先前的配置。 然后并点击 next（下一步）：



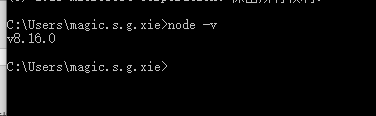
安装过程：



点击 Finish（完成）按钮退出安装向导。

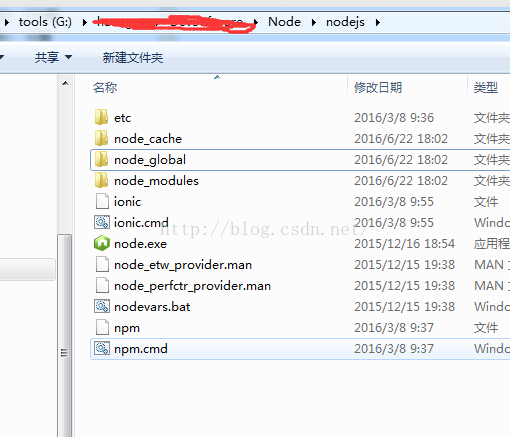


检查Node.js版本：



### 全局配置

在nodejs文件夹里面新建以下两个目录：node\_global，node\_cache



启动CMD依次执行以下两条命令

npm config set prefix "XXX\nodejs\node\_global"

npm config set cache "XXX\nodejs\node\_cache"

设置环境变量：

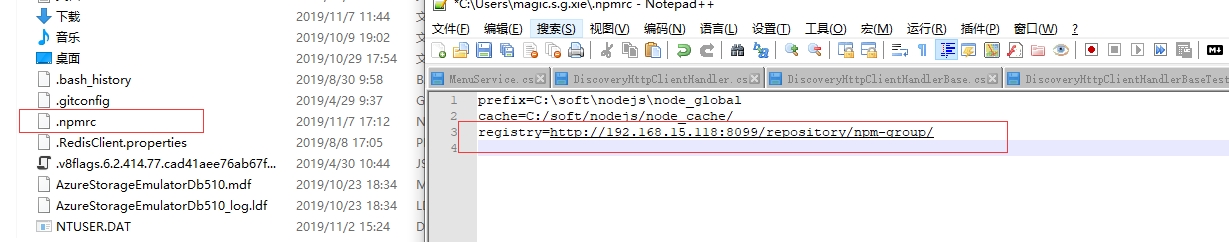
NODE\_PATH = XXX\Node\nodejs

PATH= %NODE\_PATH%\;%NODE\_PATH%\node\_modules;%NODE\_PATH%\node\_global;

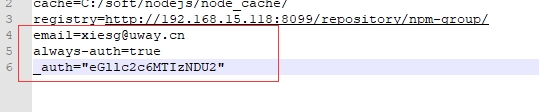
### 设置全局私服

私服地址：   <http://192.168.15.118:8099/repository/npm-group/>

设置私服：



配置发布信息：



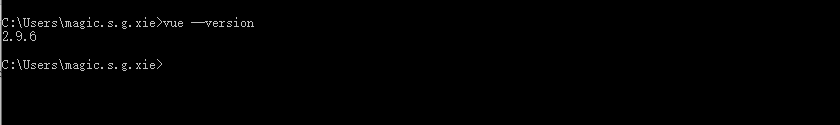
注意：\_auth是 username:password 的base64值

## 安装vue

打开Shell或者CMD,使用下列命令：

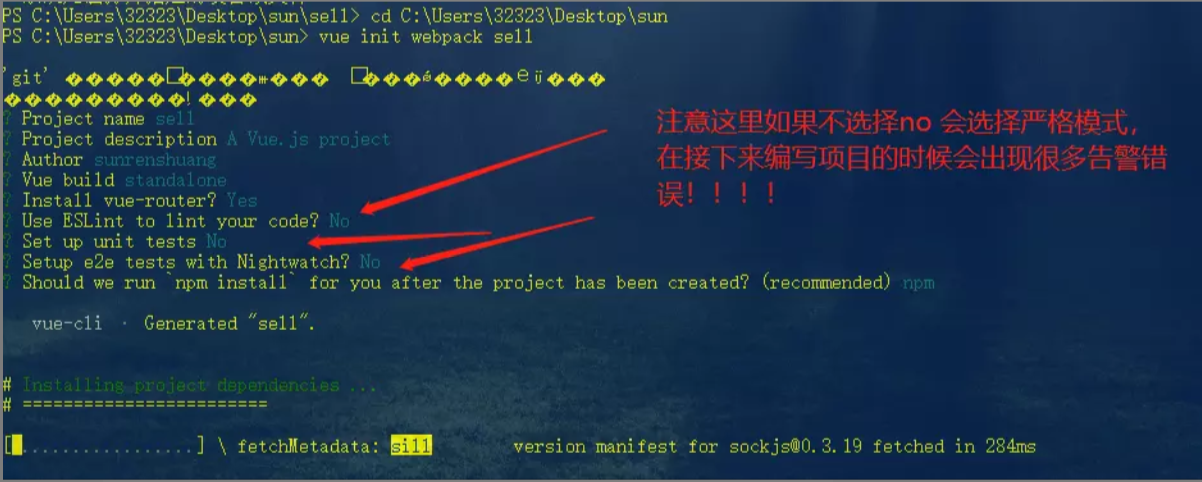
npm install -g @vue/cli

使用下列命令查看是否安装成功：vue --version

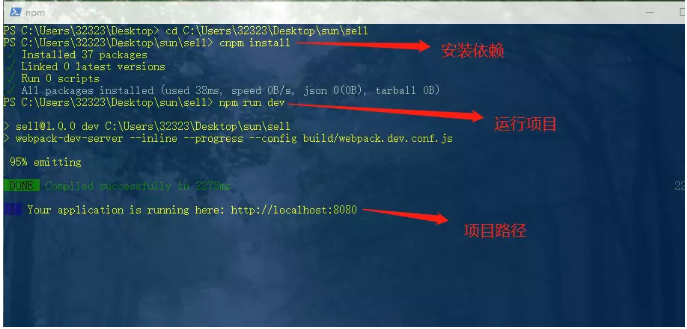


## Vue创建工程

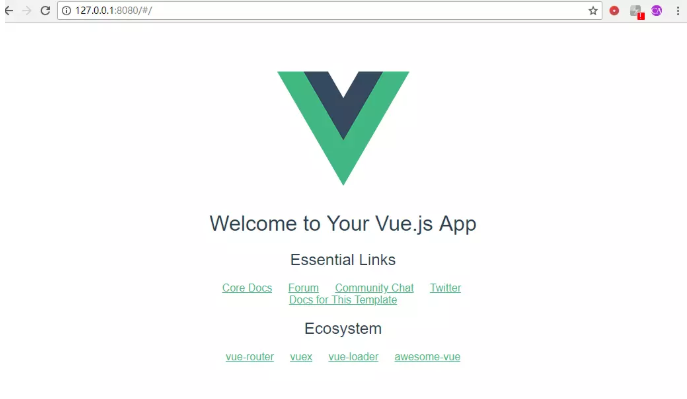
vue init webpack  ”项目名称“



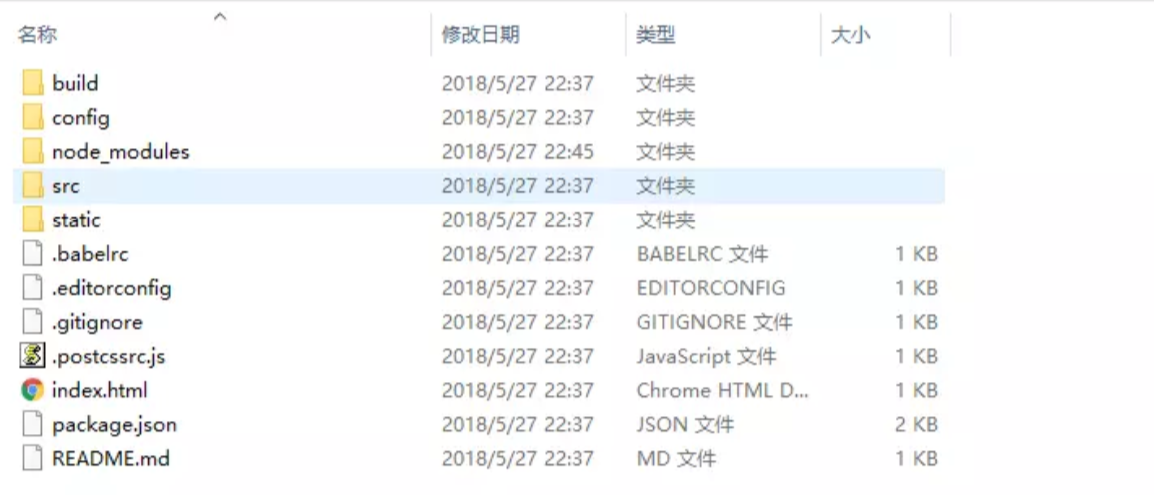
运行



成功



利用vue-cil初始化构建vue项目，我们会获得一个初始化的文件夹结构，如下：



# 快速上手

* UVUE SVN地址：<http://192.168.15.201/svn/cdma3/trunk/code/uvue>
* **导入**

min是生产包，另一个则是开发包

* uvue.js
* uvue.min.js
* **cdn方式**

将下载的包放入public目录下新建的lib下

//在index.html引入uvuex的包

uvue.min.js为压缩混淆包

uvue.js为没有压缩混淆的包

<link rel="stylesheet" href="/lib/index.css" />

<script src="/lib/uvue.min.js"></script>

//在main.js中使用

Vue.use(window.UVUE);

* **npm方式**

推荐使用 npm 的方式安装，它能更好地和 webpack 打包工具配合使用。

npm i @uway/uvue -S

import Uvue from '@uway/uvue';

import '@uway/uvue/lib/index.css';

Vue.use(Uvue);

**国际化**

Uvue 组件内部默认使用中文，若希望使用其他语言，则需要进行多语言设置。以英文为例，在 main.js 中：

// 完整引入英文国际化

**import** Vue **from** 'vue'

**import** Uvue **from** 'Uvue'

Vue.use(Uvue, { locale:'en' })

**兼容其他 i18n 插件**

如果不使用 vue-i18n@5.x，而是用其他的 i18n 插件，Element 将无法兼容，但是可以自定义 Element 的 i18n 的处理方法。

**import** Vue **from** 'vue'

**import** Uvue **from** 'Uvue'

Vue.use(Element, {

i18n: **function** (path, options) {

// ...

}

})

**全局配置**

在引入 Uvuex 时，可以传入一个全局配置对象。该对象目前支持 size和menuType字段。size用于改变组件的默认尺寸;menuType用于改变操作栏菜单按钮类型，按照引入 Uvue 的方式，具体操作如下：

* 拥有 size 属性的组件的默认尺寸均为 'small'。可选值 small / mini / medium;
* 拥有 menuType 属性的组件的默认尺寸均为 'menu'。可选值 button / icon / text / menu
* 拥有 qiniu 七牛云配置

{

AK: '',

SK: '',

scope: '',

url: '',

deadline: 1

}

* 拥有 ali 阿里云配置

{

region: '',

endpoint: '',

accessKeyId: '',

accessKeySecret: '',

bucket: '',

}

* 拥有 canvas全局水印配置

{

text: 'uvue',

fontFamily: 'microsoft yahei',

color: "#999",

fontSize: 16,

opacity: 100,

bottom: 10,

right: 10,

ratio: 1

}

Vue.use(window.UVUE,{

size:'',

menuType:'',

qiniu:{},

ali:{},

canvas:{}

})

**全局api**

在uvuex中内置封装了很多全局的api函数，在项目开发中常用的函数，在引入uvuex后你可以在任何组件里取调用这些方法

**$export excel导出**

传入相关的属性配置[在线例子](https://avuejs.com/doc/export)

let opt = {

title: '文档标题',

column: [{

label: '标题',

prop: 'title'

}],

data: [{

title: "测试数据1"

}, {

title: "测试数据2"

}]

}

this.$export.excel({

title: opt.title || new Date().getTime(),

columns: opt.column,

data: opt.data

});

**watermark全局水印**

传入水印的文案[在线例子](https://avuejs.com/doc/watermark)

this.watermark('我是水印的文案');

**deepClone对象深拷贝**

可对对象和数组深拷贝

var obj1 = {

name:'张三'

}

var obj2 = this.deepClone(obj1);

**setPx设置css像素方法**

如果传入是数字默认加px属性

var obj = 23

console.log(this.setPx(obj)) //'23px'

console.log(this.setPx('100%')) //'100%'

console.log(this.setPx('23px')) //'23px'

**validatenull判断空**

可以判断对象、数组、字符串是否为空

var obj1 = {}

var obj2 = []

var str1 = ''

var str2 = undefined

var str3 = null;

console.log(this.validatenull(obj1)) //true

console.log(this.validatenull(obj2)) //true

console.log(this.validatenull(str1)) //true

console.log(this.validatenull(str2)) //true

console.log(this.validatenull(str3)) //true

**findArray数组中寻找对象下标**

根据对象属性在数组中找到符合的下表返回，没有则为-1

var list = [{

prop:'name'

},{

prop:'sex'

}]

var index = this.findArray(list,'sex','prop');

console.log(index) //1

**vaildData验证对象**

验证是否为空的任何类型数据,为空择取默认的设置值

var obj = {}

console.log(this.vaildData(obj),'默认值') //默认值

console.log(this.vaildData(obj),{name:11}) //{name:11}

var obj2 = true

console.log(this.vaildData(obj2),false) //true

**$Log日志打印**

可以打印不同颜色和标注的日志

/\*\*

\* 内置5中常用颜色,默认为primary

\* default:#35495E

\* primary:#3488ff

\* success:#43B883

\* warning:#e6a23c

\* danger:#f56c6c

\* 也可以直接打印彩色文字

\*/

this.$Log.capsule('标题','内容','primary')

this.$Log.primary('内容')

**$Clipboard复制到剪切板**

可以赋值任意文本到剪切板[在线例子](https://avuejs.com/doc/clipboard)

this.$Clipboard({

text: '复制的文本内容'

}).then(() => {

this.$message.success('复制成功')

}).catch(() => {

this.$message.error('复制失败')

});

**$ImagePreview图片预览**

以赋值任赋值图片去放大预览(一张缩略图，一张放大图)[在线例子](https://avuejs.com/doc/image-preview)

data() {

var link = 'https://lokeshdhakar.com/projects/lightbox2/images/';

return {

datas: [

{ thumbUrl: `${link}thumb-4.jpg`, url: `${link}image-4.jpg` },

{ thumbUrl: `${link}thumb-5.jpg`, url: `${link}image-5.jpg` },

{ thumbUrl: `${link}thumb-6.jpg`, url: `${link}image-6.jpg` },

]

}

}

this.$ImagePreview(this.datas, index)

# 图表组件

## 字符云



<**uvue-echart-wordcloud** ref="echart" :option="config" :data="data" width="1000"></**uvue-echart-wordcloud**>

<**script**>

**export** **default** {

data() {

**return** {

data: [{

"name": "iPhone X",

"value": 19651

}, {

"name": "Mac Pro",

"value": 18341

}, {

"name": "华为P20",

"value": 17319

}, {

"name": "小米手机",

"value": 12550

}, {

"name": "欧莱雅",

"value": 11638

}, {

"name": "玛莎拉蒂",

"value": 10608

}, {

"name": "特斯拉Model S",

"value": 8298

}, {

"name": "jackjones",

"value": 7630

}, {

"name": "韩都衣舍",

"value": 7440

}, {

"name": "天生要强",

"value": 6065

}, {

"name": "梅西",

"value": 5065

}, {

"name": "三只松鼠",

"value": 4965

}],

config: {

"minFontSize": 30,

"maxFontSize": 80,

"split": 30,

"rotate": true

}

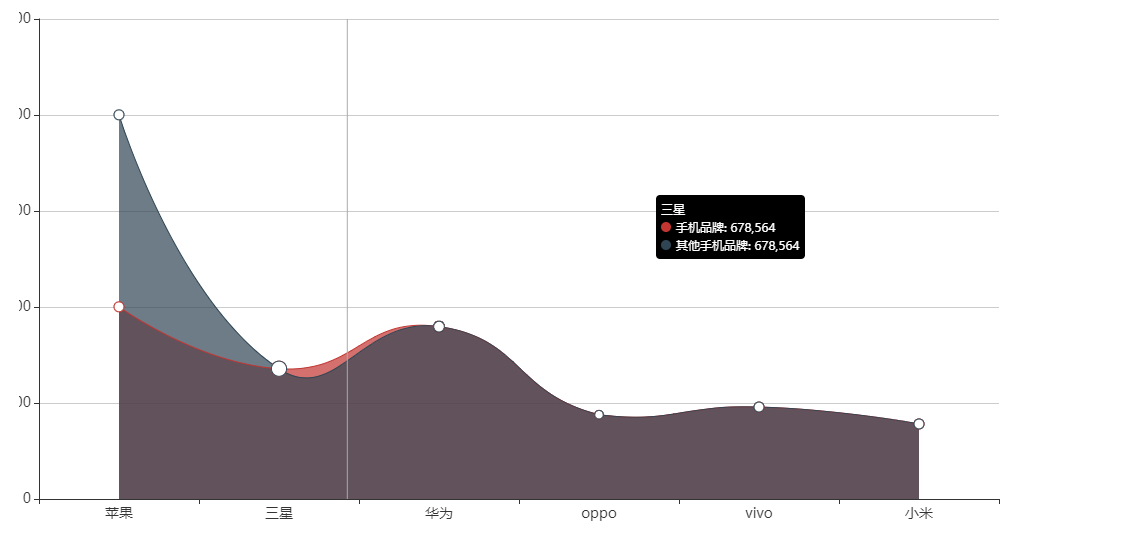
}

},

}

</**script**>

## 折线图



<**el-button** type="primary" @click="option.switchTheme=!option.switchTheme" size="small">打开主题</**el-button**>

<**el-button** @click="theme='macarons'" size="small">换紫色主题</**el-button**>

<**el-button** @click="theme='wonderland'" size="small">换绿色主题</**el-button**>

<**uvue-echart-line** :theme="theme" :option="option" :data="data" width="1000"></**uvue-echart-line**>

<**script**>

**export** **default** {

data() {

**return** {

theme:'',

data:{

categories: [

"苹果",

"三星",

"华为",

"oppo",

"vivo",

"小米"

],

series: [

{

name: "手机品牌",

data: [

1000879,

678564,

897744,

439087,

478900,

390877

]

}, {

name: "其他手机品牌",

data: [

2000879,

678564,

897744,

439087,

478900,

390877

]

}

]

},

option: {

switchTheme:true,

width: 1200,

height: 600,

title: '手机大比拼',

smooth: true,//是否顺滑

areaStyle: true,//是否面积

}

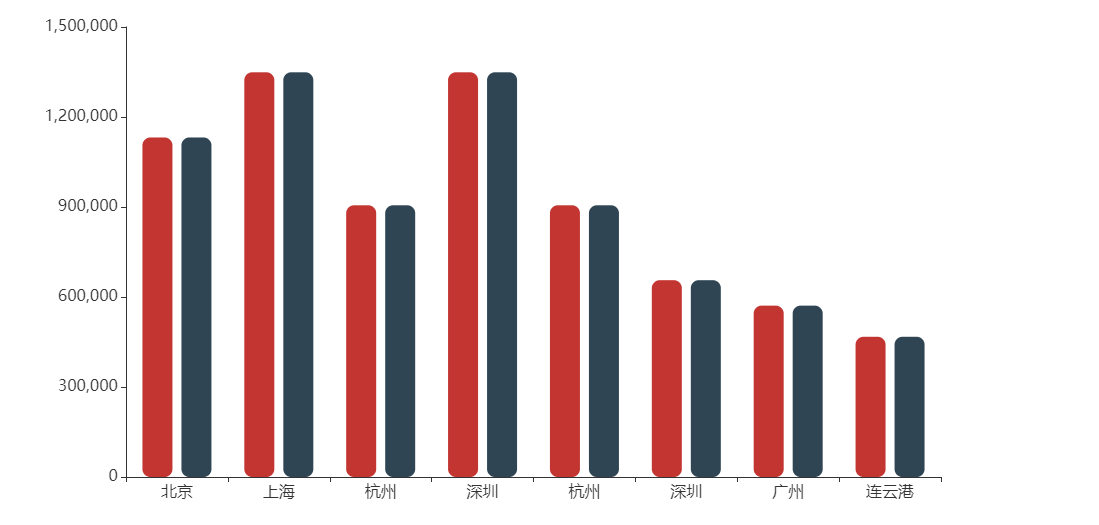
}

}

}

</**script**>

## 柱形图



<**el-button** type="primary" @click="config.switchTheme=!config.switchTheme" size="small">关闭主题</**el-button**>

<**el-button** @click="theme='macarons'" size="small">换紫色主题</**el-button**>

<**el-button** @click="theme='wonderland'" size="small">换绿色主题</**el-button**>

<**uvue-echart-bar** ref="echart" :theme="theme" :option="config" :data="data" width="1000"></**uvue-echart-bar**>

<**script**>

**export** **default** {

data() {

**return** {

theme:'',

data: {

"categories": [

"北京",

"上海",

"杭州",

"深圳",

"杭州",

"深圳",

"广州",

"连云港"

],

"series": [

{

"name": "手机品牌",

"data": [

1131776,

1349011,

905718,

1349011,

905718,

656020,

571234,

467397

]

}, {

"name": "其他手机品牌",

"data": [

1131776,

1349011,

905718,

1349011,

905718,

656020,

571234,

467397

]

}

]

},

config: {

switchTheme:true,

refreshTime: 5000,

title: '2019年度潍坊城市销售额排名',

titlePostion: '',

titleFontSize: 24,

titleColor: 'rgb(115, 170, 229)',

info: '这是图表的简介',

gridX: 105,

gridY: 50,

gridX2: 80,

gridY2: 100,

nameColor: '#333',

lineColor: '#333',

xNameFontSize: 16,

yNameFontSize: 16,

labelShow: false,

labelShowFontSize: 14,

labelShowFontWeight: 300,

yAxisInverse: false,

xAxisInverse: false,

xAxisShow: true,

yAxisShow: true,

xAxisSplitLineShow: false,

yAxisSplitLineShow: false,

refresh: 3000,

barWidth: 30,

barRadius: 8,

barColor: [{

color1: '#83bff6',

color2: '#188df0',

postion: 90

}, {

color1: '#23B7E5',

color2: '#564AA3',

postion: 50

}],

barMinHeight: 2,

}

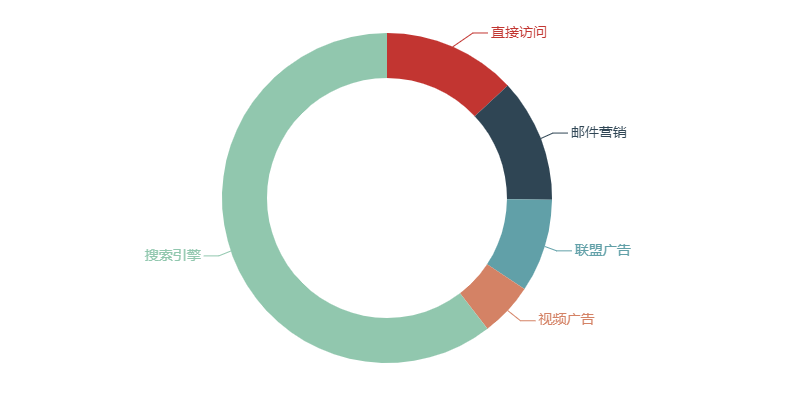
}

},

}

</**script**>

## 饼图



<**el-button** type="primary" @click="option.switchTheme=!option.switchTheme" size="small">关闭主题</**el-button**>

<**el-button** @click="theme='macarons'" size="small">换紫色主题</**el-button**>

<**el-button** @click="theme='wonderland'" size="small">换绿色主题</**el-button**>

<**uvue-echart-pie** :theme="theme" :option="option" :data="data" width="1000"></**uvue-echart-pie**>

<**script**>

**export** **default** {

data() {

**return** {

theme:'',

data: [

{ value: 335, name: '直接访问' },

{ value: 310, name: '邮件营销' },

{ value: 234, name: '联盟广告' },

{ value: 135, name: '视频广告' },

{ value: 1548, name: '搜索引擎' }

],

option: {

switchTheme:true,

width: '100%',

height: 600,

title: '手机大比拼',

subtitle: '纯属虚构',

labelShow:true,

radius: true,

"barColor": [

{

"color1": "#83bff6",

},

{

"color1": "#23B7E5",

},

{

"color1": "rgba(154, 168, 212, 1)",

},

{

"color1": "#188df0",

},

{

"color1": "#564AA3",

}

]

},

}

}

}

</**script**>

## 刻度盘



<**el-button** type="primary" @click="config.switchTheme=!config.switchTheme" size="small">关闭主题</**el-button**>

<**el-button** @click="theme='macarons'" size="small">换紫色主题</**el-button**>

<**el-button** @click="theme='wonderland'" size="small">换绿色主题</**el-button**>

<**uvue-echart-gauge** ref="echart" :theme="theme" :option="config" :data="data" width="1000"></**uvue-echart-gauge**>

<**script**>

**export** **default** {

data() {

**return** {

theme:'',

data: {

value: 50,

name: '完成率',

min: 0,

max: 100,

unit: '%'

},

config: {

switchTheme:true,

lineSize: 25,

barColor: [{

postion: 0.2,

color1: '#91c7ae'

},

{

postion: 0.8,

color1: '#63869e'

},

{

postion: 1,

color1: '#c23531'

}

]

}

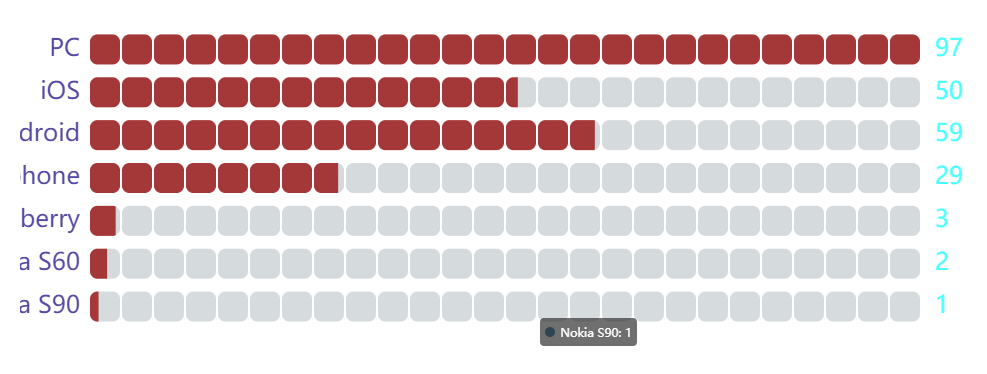
}

},

}

</**script**>

## 象形图



<**uvue-echart-pictorialbar** ref="echart" :option="config" :data="data" width="1000"></**uvue-echart-pictorialbar**>

<**script**>

**export** **default** {

data() {

**return** {

data: [{

"name": "PC",

"value": 97,

"url": "http://www.baidu.com"

},

{

"name": "iOS",

"value": 50,

"url": "http://www.baidu.com"

},

{

"name": "Android",

"value": 59,

"url": "http://www.baidu.com"

},

{

"name": "windows phone",

"value": 29

},

{

"name": "Black berry",

"value": 3

},

{

"name": "Nokia S60",

"value": 2

},

{

"name": "Nokia S90",

"value": 1

}

],

config: {

symbolSize: 30,

split: 30,

fontSize: '24',

xAxisShow: false,

yAxisShow: true,

color: '#4dffff',

nameColor: '#564AA3',

xNameFontSize: '24',

yNameFontSize: '24'

}

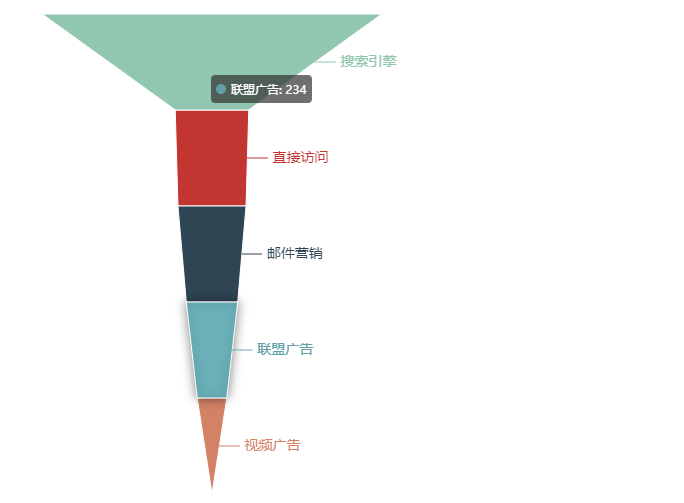
}

},

}

</**script**>

## 漏斗图



<**el-button** type="primary" @click="option.switchTheme=!option.switchTheme" size="small">关闭主题</**el-button**>

<**el-button** @click="theme='macarons'" size="small">换紫色主题</**el-button**>

<**el-button** @click="theme='wonderland'" size="small">换绿色主题</**el-button**>

<**uvue-echart-funnel** :theme="theme" :option="option" :data="data" :width="500"></**uvue-echart-funnel**>

<**script**>

**export** **default** {

data() {

**return** {

theme:'',

data: [{

value: 335,

name: '直接访问'

},

{

value: 310,

name: '邮件营销'

},

{

value: 234,

name: '联盟广告'

},

{

value: 135,

name: '视频广告'

},

{

value: 1548,

name: '搜索引擎'

}

],

option: {

switchTheme:true,

width: '100%',

height: 600,

title: '手机大比拼',

subtext:'最近一个月',

titleColor: '#666',

titleFontSize: 18,

// legendShow: true,

labelShow: true,

radius: true,

"barColor": [{

"color1": "#83bff6",

},

{

"color1": "#23B7E5",

},

{

"color1": "rgba(154, 168, 212, 1)",

},

{

"color1": "#188df0",

},

{

"color1": "#564AA3",

}

]

},

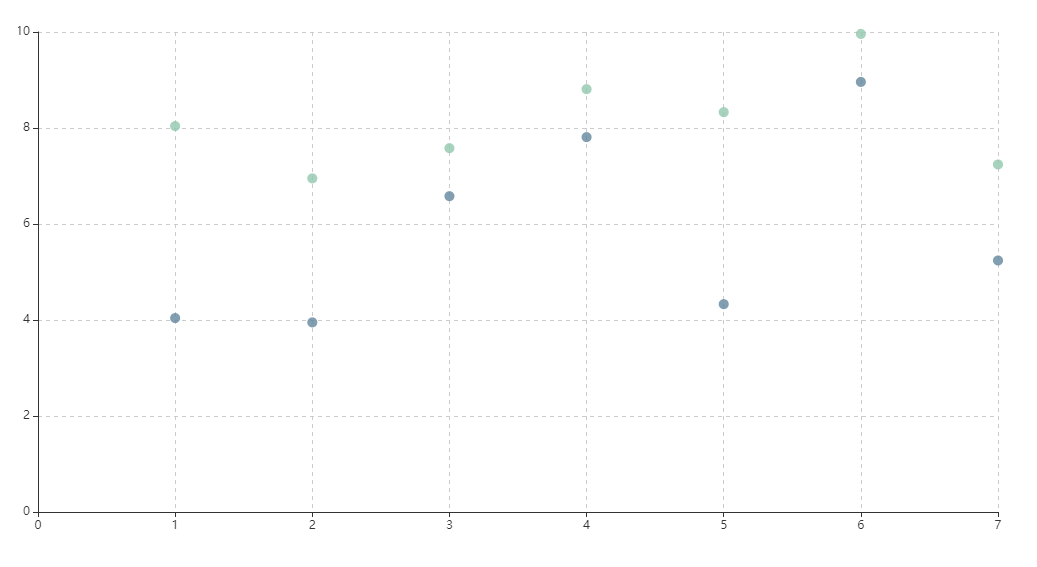
}

},

}

</**script**>

## 散点图



<**el-button** type="primary" @click="option.switchTheme=!option.switchTheme" size="small">关闭主题</**el-button**>

<**el-button** @click="theme='macarons'" size="small">换紫色主题</**el-button**>

<**el-button** @click="theme='wonderland'" size="small">换绿色主题</**el-button**>

<**uvue-echart-scatter** :option="option" :data="data" :width="1000"></**uvue-echart-scatter**>

<**script**>

**export** **default** {

data() {

**return** {

theme:'',

option: {

switchTheme:true,

title: '2019年度潍坊城市销售额排名',

lineSize: 25,

barColor: [{

postion: 0.2,

color1: '#91c7ae'

},

{

postion: 0.8,

color1: '#63869e'

},

{

postion: 1,

color1: '#c23531'

}

]

},

data: [{

data: [

[1, 8.04],

[2, 6.95],

[3, 7.58],

[4, 8.81],

[5, 8.33],

[6, 9.96],

[7, 7.24]

]

},

{

data: [

[1, 4.04],

[2, 3.95],

[3, 6.58],

[4, 7.81],

[5, 4.33],

[6, 8.96],

[7, 5.24]

]

}]

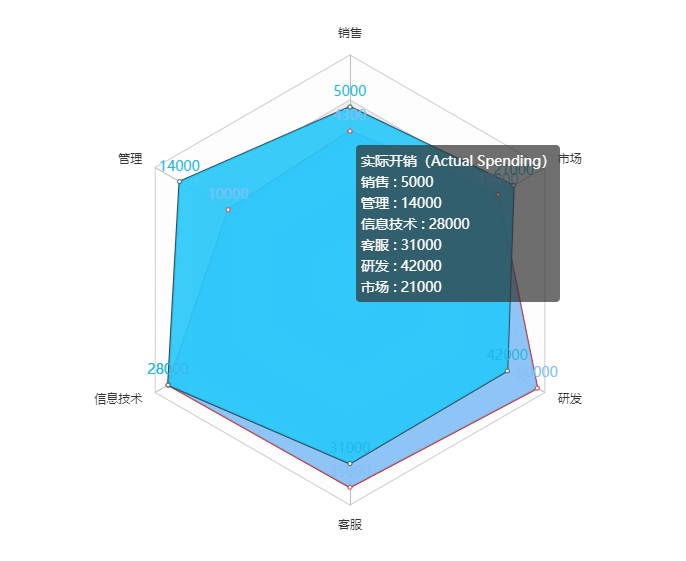
}

},

}

</**script**>

## 雷达图



<**el-button** type="primary" @click="option.switchTheme=!option.switchTheme" size="small">关闭主题</**el-button**>

<**el-button** @click="theme='macarons'" size="small">换紫色主题</**el-button**>

<**el-button** @click="theme='wonderland'" size="small">换绿色主题</**el-button**>

<**uvue-echart-radar** :theme="theme" :option="option" :data="data" :width="1000"></**uvue-echart-radar**>

<**script**>

**export** **default** {

data() {

**return** {

theme:'',

data: {

indicator: [{

name: '销售',

max: 6500

},

{

name: '管理',

max: 16000

},

{

name: '信息技术',

max: 30000

},

{

name: '客服',

max: 38000

},

{

name: '研发',

max: 52000

},

{

name: '市场',

max: 25000

}

],

series: [{

data: [{

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）'

}

]

}]

},

option: {

switchTheme:true,

width: '100%',

height: 600,

title: '各部门开销',

subtext: '最近一个月',

titleColor: '#666',

shape: 'polygon',

radius: "75%",

splitNumber: 4,

radarNameColor: '#333',

radarNameSize: 12,

titleFontSize: 18,

legendShow: true,

labelShow: true,

"barColor": [{

"color1": "#83bff6",

},

{

"color1": "#23B7E5",

},

{

"color1": "rgba(154, 168, 212, 1)",

},

{

"color1": "#188df0",

},

{

"color1": "#564AA3",

}

]

},

}

},

}

</**script**>

# 表单组件

## Form

### Variables

| **参数** | **说明** | **类型** | **可选值** | **默认值** |
| --- | --- | --- | --- | --- |
| option | 组件配置属性，详情见下面Option属性 | Object | — | — |
| upload-before | 图片上传前的回调,会暂停图片上传function(file,done)，done用于继续图片上传 | Function | — | — |
| upload-after | 图片上传后的回调,function(res,done)，done用于结束操作 | Function | — | — |

### Option Attribute

| **参数** | **说明** | **类型** | **可选值** | **默认值** |
| --- | --- | --- | --- | --- |
| card | 卡片效果 | Boolean | true/false | false |
| emptyBtn | 清空按钮 | Boolean | true/false | true |
| emptySize | 清空按钮的大小 | String | medium/mini/small | medium |
| emptyText | 清空按钮的文字 | String | - | 清空 |
| group | 分组 | Array | - | - |
| gutter | 项之间的间 | Number | - | 20 |
| icon | 表单的标题图标 | String | - | 80 |
| label | 表单的标题名称 | String | - | 80 |
| labelWidth | 表单的label宽度 | String | - | 80 |
| labelPosition | 表单的label位置 | String | left/top/right | left |
| menuBtn | 是否显示按钮 | Boolean | true/false | true |
| menuPostion | 按钮的位置 | String | left/center/right | center |
| prop | 表单的标题列字段 | String | - | 80 |
| props | 字典的全局key配置 | Object | - | - |
| size | 表单全局控件的大小 | String | medium/mini/small | medium |
| submitBtn | 提交按钮 | Boolean | true/false | true |
| submitSize | 提交按钮的大小 | String | medium/mini/small | medium |
| submitText | 提交按钮的文字 | String | - | 提交 |

### Props Attribute

| **参数** | **说明** | **类型** | **可选值** | **默认值** |
| --- | --- | --- | --- | --- |
| label | 字典的名称属性值 | String | — | — |
| value | 字典的值属性值 | String | — | — |
| children | 字典的子属性值 | String | — | — |
| res | 网络字典返回的数据格式 |  |  |  |

### Type Attribute

这是type属性可以配置的组件，当然你也可以自定义，参考第三方组件导入

| **参数** | **类型** |
| --- | --- |
| array | 数组框 |
| color | 颜色选择框 |
| cascader | 级联框 |
| checkbox | 多选框 |
| date | 日期框 |
| datetime | 日期时间框 |
| daterange | 日期范围 |
| datetimerange | 日期时间范围 |
| dates | 多个日期 |
| dynamic | 子表单 |
| icon | 图标选择框 |
| input | 输入框 |
| img | 图片框 |
| month | 月 |
| password | 密码框 |
| radio | 单选框 |
| select | 选择框 |
| switch | 开关框 |
| slider | 滑动框 |
| textarea | 文本框 |
| rate | 评价框 |
| time | 时间框 |
| timerange | 时间范围 |
| tree | 树框 |
| url | 超链 |
| week | 周 |
| year | 年 |

### Fomat Attribute

使用format指定输入框的格式；使用valueFormat指定绑定值的格式。

默认情况下，组件接受并返回Date对象。以下为可用的格式化字串，以UTC2017年1月2日03:04:05为例：

注意： 请注意大小写

| **格式** | **含义** | **备注** | **举例** |
| --- | --- | --- | --- |
| yyyy | 年 |  | 2017 |
| M | 月 | 不补0 | 1 |
| MM | 月 |  | 01 |
| W | 周 | 仅周选择器的format可用；不补0 | 1 |
| WW | 周 | 仅周选择器的format可用 | 01 |
| d | 日 | 不补0 | 2 |
| dd | 日 |  | 02 |
| H | 小时 | 24小时制；不补0 | 3 |
| HH | 小时 | 24小时制 | 03 |
| h | 小时 | 12小时制，须和A或a使用；不补0 | 3 |
| hh | 小时 | 12小时制，须和A或a使用 | 03 |
| m | 分钟 | 不补0 | 4 |
| mm | 分钟 |  | 04 |
| s | 秒 | 不补0 | 5 |
| ss | 秒 |  | 05 |
| A | AM/PM | 仅format可用，大写 | AM |
| a | am/pm | 仅format可用，小写 | am |
| timestamp | JS时间戳 | 仅value-format可用；组件绑定值为number类型 | 1483326245000 |

### Column Attribute

| **参数** | **说明** | **类型** | **可选值** | **默认值** |
| --- | --- | --- | --- | --- |
| accept | 为upload时文件类型 | String/Array | — | - |
| cascaderChange | 级联选择时下级是否自选 | Boolean | true/false | true |
| dataType | 数据的类型转换也就是  数组和用逗号隔开的字  符串（适用于checkbox  /select多选/tree多选/img/array/upload） | String | number/string | - |
| cascaderItem | 需要级联的子节点prop | Array | - | - |
| changeOnSelect | 是否允许选择任意一级  的选项 | Boolean | true/false | false |
| clearable | 表单清空 | Boolean | true/false | false |
| disabled | 全部是否禁止 | Boolean | true/false | false |
| defaultIndex | 级联选择下级默认  选择的index | Number | - | 1 |
| endPlaceholder | 日期范围结束占位符 | String | - | - |
| formHeight | 表单行高度 | Number | - | auto |
| formslot | 表单自定义 | Boolean | true/false | false |
| filesize | 为upload时文件的大小 | Number | - | - |
| filterable | 级联框是否可以搜索 | Boolean | true/false | false |
| Function(value,row,column) | - | - |  |  |
| format | 显示值时间格式（  当type为date/time  /datetime  /daterangetimerange  /datetimerange  /week/month  /year/dates | - | - | - |
| formatter | 用来格式化内容 | Function  (row,value,label,column) | - | - |
| label | 列名称 | String | — | - |
| typeslot | 表单组件自定义 | Boolean | true/false | false |
| multiple | 多选（当type为  select/tree时） | Boolean | true/false | false |
| minRows | 最小行（当type为  textarea） | Number | - | 2 |
| maxRows | 最大行（当type为  textarea） | Number | - | 4 |
| offset | 当前项的偏移量 | Number | - | 0 |
| labelSuffix | label后缀字符串 | String | - | : |
| precision | 数字框输入精度（  当type为number时） | Number | - | 2 |
| prop | 列字段 | String | — | - |
| placeholder | 辅助语 | String | — | 请选择/请输入+label |
| readonly | 只读 | Boolean | true/false | false |
| rules | 表单规则,参考ele表单  规则配置 | Object | - | - |
| row | 截断后面的排列 | Boolean | true/false | false |
| span | 表单栅列 | Number | - | 12 |
| startPlaceholder | 日期范围开始占位符 | String | - | - |
| size | 表单大小 | String | small/mini | small |
| showSllLevels | 级联输入框中是否  显示选中值的完整路径 | Boolean | true/false | true |
| separator | 级联输入框选项分隔符 | String | - | 斜杠'/' |
| tip | 辅助提示语 | String | - | - |
| tipPlacement | 辅助提示语的方向 | String | top-start/top  /top-end  /left-start  /leftleft-end  /right-start  /right  /right-endbottom-start  /bottom  /bottom-end | bottom |
| type | 类型 | String | Input  /select  /radio  /checkbox  /textarea  /cascade  /date  /time  /datetime  /daterange  /timerange  /datetimerange  /week  /month  /year  /dates  /ueditor  /password/switch/tree | input |
| display | 全部是否可见 | Boolean | true/false | true |
| valueDefault | 表单的默认值 | Number/String | - | - |
| checkStrictly | 不遵循父子规则 | Boolean | true/false | true |
| tags | select组件多选时是  否超出后合并标签 | Boolean | true/false | false |
| valueFormat | 真实值的时间格式  （当type为date/time/datetime  /daterangetimerange  /datetimerange/week/month/year/dates） | - | - | - |

### Events

| **事件名** | **说明** | **参数** |
| --- | --- | --- |
| reset-change | 重置表单回调 |  |
| submit | 表单提交事件 |  |

### Methods

| **方法名** | **说明** | **参数** |
| --- | --- | --- |
| init | 初始化参数（比如服务端加载option，需要调用该内置函数初始化） |  |
| clearValidate | 清空表格钟表单验证。 |  |
| validate | 对整个表单进行校验的方法，参数为一个回调函数。该回调函数会在校验结束后被调用，并传入两个参数：是否校验成功和未通过校验的字段。若不传入回调函数，则会返回一个promise |  |
| resetForm | 清空表单数据 |  |
| findColumnIndex | 根据prop去找column的配置属性的下标 |  |
| submit | 提交表单数据 |  |

### ScopedSlot

| **name** | **说明** |
| --- | --- |
| 列的名称 | 列自定义列的内容，参数为{row,label,dic,$index} |
| 列表单的名称 | 表单自定义列的内容，参数为{row,label,dic,$index} |
| 列表单的名称+Header(nameHeader) | 表单分组头部自定义的内容 |
| 列表单的名称+Type(nameType) | 表单自定义列的内容，参数为{row,label,value} |
| menuForm | 表单操作按钮的自定义内容,参数为{row,label,dic,$index} |

## 基础



里面主要包含了change和click俩个回调方法，返回当前的值value和列的属性column,resetForm清空表单内容，但是默认值不会被清空,resetChange为清空方法的回调,如果想阻止依次排列，在组织属性后面加入row为true即可，配置mock为true，在对应的column配置mock,配置即可开启数据填充,具体看如下例子：

<**el-row** style="margin-bottom:20px">

<**el-radio-group** v-model="sizeValue">

<**el-radio** label="">默认</**el-radio**>

<**el-radio** label="small">small</**el-radio**>

<**el-radio** label="mini">mini</**el-radio**>

</**el-radio-group**>

</**el-row**>

<**uvue-form** ref="form" v-model="obj" :option="option" @reset-change="emptytChange" @submit="submit">

<**template** slot-scope="scope" slot="menuForm">

<**el-button** @click="tip">自定义按钮</**el-button**>

</**template**>

</**uvue-form**>

<**script**>

**var** DIC = {

VAILD: [{

label: '真',

value: 'true'

}, {

label: '假',

value: 'false'

}],

SEX: [{

label: '男',

value: 0

}, {

label: '女',

value: 1

}]

}

**export** **default** {

data() {

**return** {

obj:{},

sizeValue:''

}

},

computed: {

option(){

**return**{

size:**this**.sizeValue,

mock:true,

submitText: '完成',

column: [{

label: "用户名",

prop: "username",

tip: '这是信息提示',

span: 8,

maxlength: 3,

suffixIcon: 'el-icon-tickets',

prefixIcon: 'el-icon-tickets',

minlength: 2,

mock:{

type:'name',

en:true,

},

rules: [{

required: true,

message: "请输入用户名",

trigger: "blur"

}],

change:({value,column})=>{

**this**.obj.address=value;

**this**.$message.success('address change')

},

click:({value,column})=>{

**this**.$message.success('click')

}

},

{

label: "姓名",

prop: "name",

mock:{

type:'name'

},

span:8

},

{

label: "类型",

prop: "type",

type: "select",

dicData: DIC.VAILD,

span:6,

mock:{

type:'dic',

},

},

{

label: "权限",

prop: "grade",

span: 6,

type: "checkbox",

dicData: DIC.VAILD,

mock:{

type:'dic',

},

},

{

label: "开关",

prop: "switch",

span: 6,

type: "switch",

dicData: DIC.SEX,

mock:{

type:'dic'

},

hide: true,

row:true,

},

{

label: "性别",

prop: "sex",

span: 6,

type: "radio",

dicData: DIC.SEX,

mock:{

type:'dic'

},

valueDefault: 0,

change:({value,column})=>{

**this**.$message.success('change')

}

},

{

label: "数字",

prop: "number",

type: 'number',

span: 6,

precision:2,

mock:{

type:'number',

max:1,

min:2,

precision:2

},

valueDefault: 3,

minRows: 0,

maxRows: 3,

row:true,

},

{

label: "网站",

span: 12,

prop: "url",

prepend:'http://',

mock:{

type:'url',

header:false,

},

append:'com',

row:true,

},

{

label: "日期",

prop: "date",

type: "date",

span:8,

format:'yyyy-MM-dd',

valueFormat:'yyyy-MM-dd',

mock:{

type:'datetime',

format:'yyyy-MM-dd'

},

},

{

label: "日期时间",

prop: "datetime",

type: "datetime",

span:8,

format:'yyyy-MM-dd hh:mm:ss',

valueFormat:'yyyy-MM-dd hh:mm:ss',

mock:{

type:'datetime',

format:'yyyy-MM-dd hh:mm:ss',

now:true,

},

},

{

label: "时间",

prop: "time",

type: "time",

span:8,

format:'hh:mm:ss',

valueFormat:'hh:mm:ss',

mock:{

type:'datetime',

format:'hh:mm:ss'

},

},

{

label: "地址",

span: 24,

prop: "address",

mock:{

type:'county'

},

},{

label: "建议",

span: 24,

prop: "adit",

mock:{

type:'word',

min:10,

max:30

},

},{

label: "手机号",

mock:{

type:'phone'

},

span: 12,

prop: "phone",

}]

}

}

},

mounted(){

**this**.obj.username = 'smallwei'

**this**.obj.switch = 0;

**this**.obj.phone='17547400800';

},

methods:{

emptytChange(){

**this**.$message.success('清空方法回调');

},

submit () {

**this**.$message.success('当前数据'+JSON.stringify(**this**.obj));

},

tip(){

**this**.$message.success('自定义按钮');

}

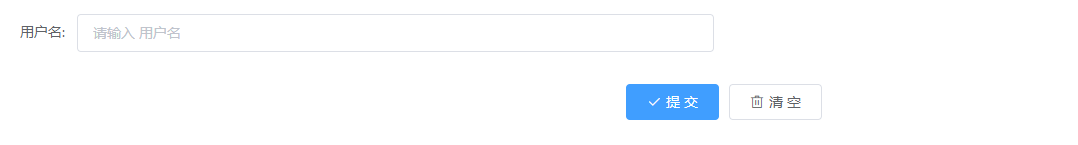
}

}

</**script**>

## 防重提交

提交后禁止表单和按钮重复提交：



submit中调用done方法和crud中使用一致

<**uvue-form** :option="option" v-model="form" @submit="handleSubmit"></**uvue-form**>

<**script**>

**export** **default** {

data(){

**return** {

form:{

text:'动态内容1'

},

option:{

column: [{

label: "用户名",

prop: "username"

}]

}

}

},

methods:{

handleSubmit(form,done){

**this**.$message.success('3s后关闭');

setTimeout(()=>{

done()

},3000)

}

}

}

</**script**>

## 日期



<**uvue-form** v-model="form" :option="option" @submit="submit"></**uvue-form**>

<**script**>

**export** **default** {

data() {

**return** {

form:{

datetime:'1398250549123'//时间戳例子

},

option:{

labelWidth:110,

column: [{

label: "日期范围",

prop: "daterange",

type: "daterange",

startPlaceholder: '日期开始范围自定义',

endPlaceholder: '日期结束范围自定义',

}, {

label: "时间范围",

value: 'timerange',

type: 'timerange',

startPlaceholder: '时间开始范围自定义',

endPlaceholder: '时间结束范围自定义',

}, {

label: "时间日期范围",

value: 'datetimerange',

type: 'datetimerange',

span:24,

row:true,

startPlaceholder: '时间日期开始范围自定义',

endPlaceholder: '时间日期结束范围自定义',

}, {

label: "日期",

prop: "date",

type: "date",

pickerOptions: {

disabledDate(time) {

**return** time.getTime() > Date.now();

},

shortcuts: [{

text: '今天',

onClick(picker) {

picker.$emit('pick', **new** Date());

}

}, {

text: '昨天',

onClick(picker) {

**const** date = **new** Date();

date.setTime(date.getTime() - 3600 \* 1000 \* 24);

picker.$emit('pick', date);

}

}, {

text: '一周前',

onClick(picker) {

**const** date = **new** Date();

date.setTime(date.getTime() - 3600 \* 1000 \* 24 \* 7);

picker.$emit('pick', date);

}

}]

}

}, {

label: "时间",

prop: "time",

type: "time",

}, {

label: "日期时间",

prop: "datetime",

type: "datetime",

format: "yyyy-MM-dd hh:mm:ss",

valueFormat: "timestamp",

}, {

label: "周",

prop: "week",

type: "week",

format: "yyyy 第 WW 周"

}, {

label: "月",

prop: "month",

type: "month",

}, {

label: "月范围",

prop: "monthrange",

type: "monthrange",

}, {

label: "年",

prop: "year",

type: "year",

}, {

label: "多个日期",

prop: "dates",

type: "dates",

format: "yyyy-MM-dd hh:mm:ss",

valueFormat: "yyyy-MM-dd hh:mm:ss",

}]

},

};

},

methods: {

submit(){

**this**.$message.success('当前数据'+JSON.stringify(**this**.form))

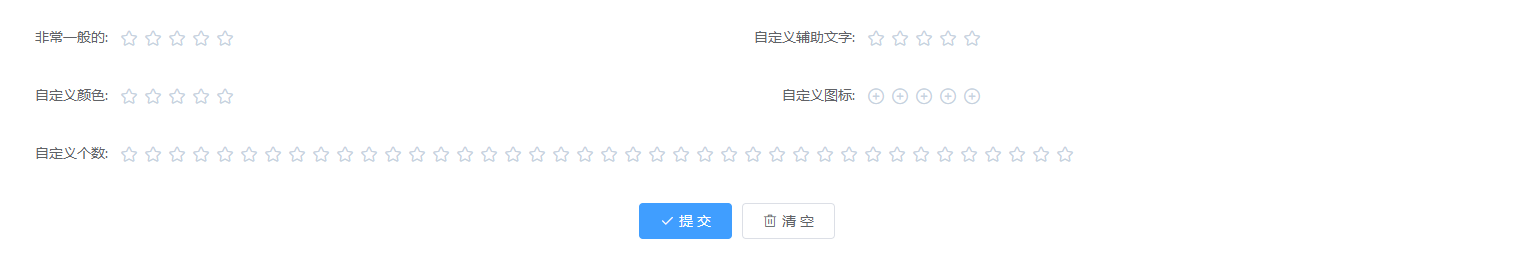
}

}

}

</**script**>

## 评价



<**uvue-form** v-model="form" :option="option" @submit="submit"></**uvue-form**>

<**script**>

**export** **default** {

data() {

**return** {

form:{},

option:{

labelWidth: 120,

column: [{

label: '非常一般的',

prop: 'score1',

type: 'rate',

}, {

label: '自定义辅助文字',

prop: 'score2',

type: 'rate',

showText: true,

texts: ['极差', '失望', '一般', '满意', '惊喜'],

}, {

label: '自定义颜色',

prop: 'score3',

type: 'rate',

colors: ['#99A9BF', '#F7BA2A', '#FF9900']

}, {

label: '自定义图标',

prop: 'score4',

type: 'rate',

iconClasses: ['el-icon-circle-close', 'el-icon-remove', 'el-icon-circle-plus'],

voidIconClass: "el-icon-circle-plus-outline",

colors: ['#99A9BF', '#F7BA2A', '#FF9900']

}, {

label: '自定义个数',

prop: 'score',

type: 'rate',

span: 24,

row: true,

max: 40

}]

}

};

},

methods: {

submit(){

**this**.$message.success('当前数据'+JSON.stringify(**this**.form))

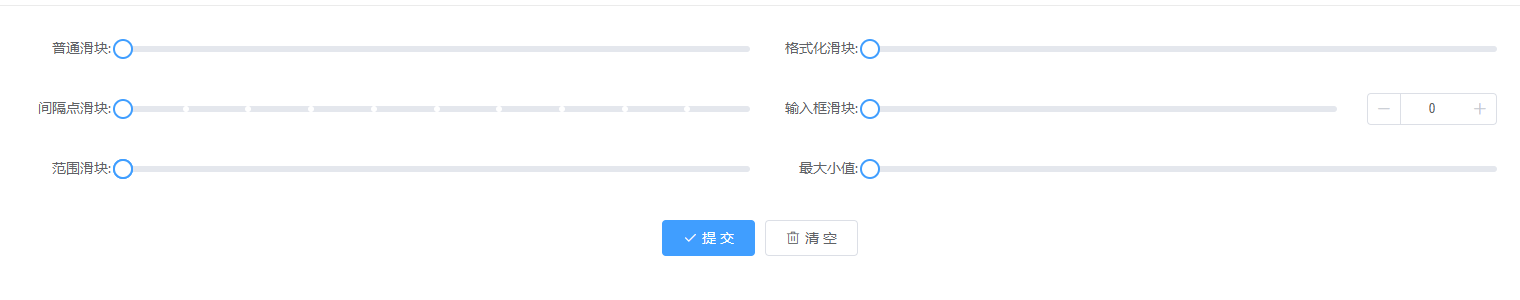
}

}

}

</**script**>

## 滑块



<**uvue-form** v-model="form" :option="option" @submit="submit"></**uvue-form**>

<**script**>

**export** **default** {

data() {

**return** {

form:{},

option:{

labelWidth: 100,

column: [{

label: "普通滑块",

prop: "pt",

max: 10,

type: 'slider',

}, {

label: "格式化滑块",

max: 10,

prop: "pt",

type: 'slider',

formatTooltip: **function**(val) {

**return** '格式化的值' + val;

}

}, {

label: "间隔点滑块",

prop: "pt",

type: 'slider',

step: 1,

max: 10,

showStops: true,

}, {

label: "输入框滑块",

prop: "pt",

max: 10,

showInput: true,

type: 'slider',

}, {

label: "范围滑块",

prop: "ptwf",

step: 1,

max: 10,

showStops: true,

range: true,

type: 'slider',

}, {

label: "最大小值",

prop: "pt",

min: 3,

max: 10,

type: 'slider',

}

]

}

};

},

methods: {

submit(){

**this**.$message.success('当前数据'+JSON.stringify(**this**.form))

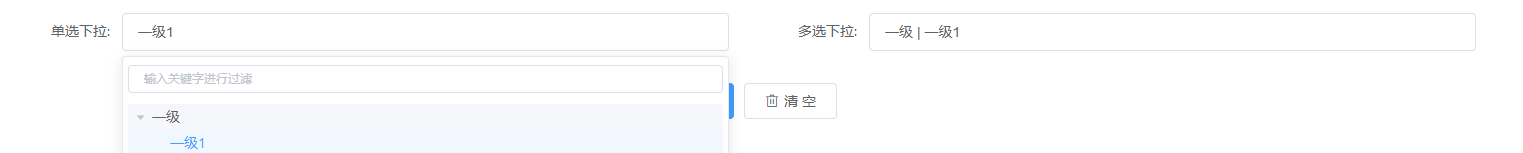
}

}

}

</**script**>

## 树形下拉框



<**uvue-form** v-model="form" :option="option" @submit="submit"></**uvue-form**>

<**script**>

**const** DIC= [{

label: '一级',

value: 0,

children: [

{

label: '一级1',

value: 1,

}, {

label: '一级2',

value: 2,

}

]

}]

**export** **default** {

data() {

**return** {

form:{

tree1:1,

tree2:[0,1]

},

option:{

labelWidth: 120,

column: [{

label: '单选下拉',

prop: 'tree1',

type: 'tree',

dicData:DIC

}, {

label: '多选下拉',

prop: 'tree2',

type: 'tree',

multiple:true,

dicData:DIC

}]

}

};

}

}

</**script**>

## 表单自定义



<**uvue-form** v-model="form" :option="option">

<**template** slot-scope="scope" slot="text">

<**div**>

<**el-tag**>测试内容1</**el-tag**>

<**el-input** v-model="form.text" placeholder="这里是自定的表单"></**el-input**>

</**div**>

</**template**>

</**uvue-form**>

<**script**>

**export** **default** {

data() {

**return** {

form:{

text:'测试内容'

},

option:{

labelWidth: 120,

column: [{

label: '自定义表单',

prop: 'text',

formslot:true

}]

}

};

}

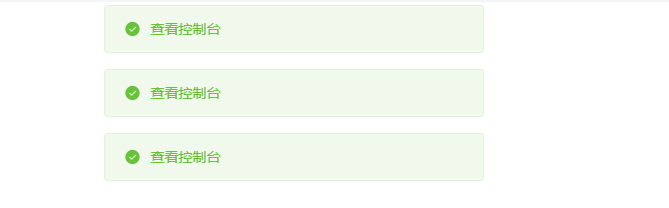
}

</**script**>

## 组件事件

* change事件
* click事件
* focus事件
* blur事件





<**uvue-form** :option="option" v-model="obj"></**uvue-form**>

<**script**>

**export** **default** {

data() {

**return** {

obj: {

name: '11'

},

data: [],

option: {

column: [{

label: '姓名',

prop: 'name',

change: () => {

**this**.$message.success('查看控制台')

console.log('值改变')

},

click: () => {

**this**.$message.success('查看控制台')

console.log('点击事件')

},

focus: () => {

**this**.$message.success('查看控制台')

console.log('获取焦点')

},

blur: () => {

**this**.$message.success('查看控制台')

console.log('失去焦点')

}

}]

}

}

}

}

</**script**>

## 数据类型



解决数据字典和字段类型不匹配问题,配置dataType属性(string / number)

<**uvue-form** :option="option" v-model="obj" > </**uvue-form**>

<**script**>

**let** baseUrl = 'https://xxxx/api/area'

**export** **default**{

data() {

**return** {

obj: {

province: 110000

},

option: {

column: [

{

label: '省份',

prop: 'province',

type: 'select',

dataType:'number',

props: {

label: 'name',

value: 'code'

},

dicUrl: `${baseUrl}/getProvince`,

rules: [

{

required: true,

message: '请选择省份',

trigger: 'blur'

}

]

}

]

}

}

},

}

</**script**>

## 字符长度提示



配置showWordLimit为true即可

<**uvue-form** :option="option" v-model="form" > </**uvue-form**>

<**script**>

**export** **default**{

data() {

**return** {

form: {

input:'欢迎使用uvue',

textarea:'欢迎使用uvue',

},

option: {

column: [

{

label:'单文本框',

prop:'input',

maxlength:10,

showWordLimit:true

}, {

label:'多文本框',

prop:'textarea',

type:'textarea',

minRows:10,

maxlength:200,

span:24,

showWordLimit:true

}

]

}

}

},

methods: {

submit() {

**this**.$message.success('当前数据' + JSON.stringify(**this**.form))

}

}

}

</**script**>

## 分组



用法普通的form组件分组用法一样，在group中配置结构体即可

<**uvue-form** :option="option" v-model="form" @submit="handleSubmit">

<**template** slot="group1Header">

<**h4**>自定义表头</**h4**>

</**template**>

</**uvue-form**>

<**script**>

**export** **default** {

data(){

**return** {

form:{

text1:'文本1',

text2:'文本2',

text3:'文本3',

},

option:{

group:[

{

icon:'el-icon-info',

label: '分组1',

prop: 'group1',

column: [{

label: '内容1',

prop: 'text1',

}]

},{

icon:'el-icon-info',

label: '分组2',

prop: 'group2',

column: [{

label: '选项卡2',

prop: 'text2',

}, {

label: '选项卡3',

prop: 'text3',

}]

}

]

}

}

},

methods:{

handleSubmit(form){

**this**.$message.success(JSON.stringify(**this**.form))

}

}

}

</**script**>

卡片展示



card组件可开启卡片分组效果，默认为false

<**uvue-form** :option="option1" v-model="form" @submit="handleSubmit"></**uvue-form**>

<**script**>

**export** **default** {

data(){

**return** {

form:{

text1:'文本1',

text2:'文本2',

text3:'文本3',

},

option1:{

card:true,

group:[

{

icon:'el-icon-info',

label: '分组1',

prop: 'group1',

column: [{

label: '内容1',

prop: 'text1',

}]

},{

icon:'el-icon-info',

label: '分组2',

prop: 'group2',

column: [{

label: '选项卡2',

prop: 'text2',

}, {

label: '选项卡3',

prop: 'text3',

}]

}

]

}

}

},

methods:{

handleSubmit(form){

**this**.$message.success(JSON.stringify(**this**.form))

}

}

}

</**script**>

## 搜索



配置type为search,appendClick为点击搜索的回调方法

<**uvue-form** :option="option" v-model="form" > </**uvue-form**>

<**script**>

**export** **default**{

data() {

**return** {

form:{},

option: {

labelWidth: 120,

column: [

{

label: '搜索内容',

prop: 'search',

type: 'search',

appendClick:()=>{

**this**.$message.success('搜索回调'+**this**.form.search);

}

}

]

}

}

}

}

</**script**>

## 颜色选择器



<**uvue-form** :option="option" v-model="form" > </**uvue-form**>

<**script**>

**export** **default**{

data() {

**return** {

form: {

color:'rgba(42, 20, 20, 1)',

},

option: {

labelWidth: 120,

column: [

{

label: '颜色选择器',

prop: 'color',

type: 'color',

}

]

}

}

}

}

</**script**>

## 图片选择器





配置type为icon-select，图标集合参考例子配置iconList属性

<**uvue-form** :option="option" v-model="form" > </**uvue-form**>

<**script**>

**export** **default**{

data() {

**return** {

form: {

icon:'el-icon-error'

},

option: {

labelWidth: 120,

column: [

{

label: '图标选择器',

prop: 'icon',

type: 'icon-select',

iconList: [{

label: '基本图标',

list: ['el-icon-info', 'el-icon-error', 'el-icon-error', 'el-icon-success', 'el-icon-warning', 'el-icon-question']

},{

label: '方向图标',

list: ['el-icon-info', 'el-icon-back', 'el-icon-arrow-left', 'el-icon-arrow-down', 'el-icon-arrow-right', 'el-icon-arrow-up']

}, {

label: '符号图标',

list: ['el-icon-plus', 'el-icon-minus', 'el-icon-close', 'el-icon-check']

},{

label: '阿里云图标',

list: ['iconfont icon-zhongyingwen', 'iconfont icon-rizhi1', 'iconfont icon-bug', 'iconfont icon-qq1', 'iconfont icon-weixin1']

},]

}

]

}

}

},

methods: {

submit() {

**this**.$message.success('当前数据' + JSON.stringify(**this**.form))

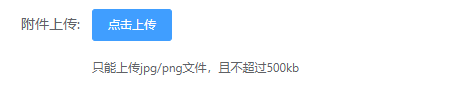
}

}

}

</**script**>

## 附件上传





<**uvue-form** :option="option" v-model="form" :upload-before="uploadBefore" :upload-after="uploadAfter"> </**uvue-form**>

<**script**>

**export** **default**{

data() {

**return** {

form: {

imgUrl:[{ "label": "uvue@226d5c1a\_logo.png", "value": "/images/logo-bg.jpg" }],

imgUrl3:'/images/logo-bg.jpg',

string:'/images/logo-bg.jpg,/images/logo-bg.jpg',

img:['/images/logo-bg.jpg','/images/logo-bg.jpg']

},

option: {

labelWidth: 120,

column: [

{

label: '附件上传',

prop: 'imgUrl',

type: 'upload',

loadText: '附件上传中，请稍等',

span: 24,

propsHttp: {

res: 'data'

},

tip: '只能上传jpg/png文件，且不超过500kb',

action: '/imgupload'

},

{

label: '水印头像',

prop: 'imgUrl3',

type: 'upload',

listType: 'picture-img',

span: 24,

propsHttp: {

res: 'data'

},

canvasOption: {

text: 'uvue',

ratio: 0.1

},

tip: '只能上传jpg/png用户头像，且不超过500kb',

action: '/imgupload'

},

{

label: '照片墙',

prop: 'imgUrl',

type: 'upload',

span: 24,

listType: 'picture-card',

tip: '只能上传jpg/png文件，且不超过500kb',

propsHttp: {

res: 'data'

},

action: '/imgupload'

},

{

label: '数组图片组',

prop: 'img',

dataType: 'array',

type: 'upload',

propsHttp: {

res: 'data.0'

},

span: 24,

listType: 'picture-card',

tip: '只能上传jpg/png文件，且不超过500kb',

action: '/imgupload'

},

{

label: '字符串图片组',

prop: 'string',

dataType: 'string',

type: 'upload',

propsHttp: {

res: 'data'

},

span: 24,

listType: 'picture-card',

tip: '只能上传jpg/png文件，且不超过500kb',

action: '/imgupload'

},

{

label: '拖拽上传',

prop: 'imgUrl',

type: 'upload',

span: 24,

drag: true,

propsHttp: {

res: 'data'

},

tip: '只能上传jpg/png文件，且不超过500kb',

action: '/imgupload'

},

{

label: '缩略图上传',

prop: 'imgUrl',

type: 'upload',

limit: 3,

span: 24,

propsHttp: {

res: 'data'

},

listType: 'picture',

tip: '只能上传jpg/png文件，且不超过500kb',

action: '/imgupload'

}

]

}

}

},

methods: {

uploadBefore(file, done, loading) {

console.log(file)

done()

**this**.$message.success('上传前的方法')

},

uploadAfter(res, done, loading) {

console.log(res)

done()

**this**.$message.success('上传后的方法')

},

submit() {

**this**.$message.success('当前数据' + JSON.stringify(**this**.form))

}

}

}

</**script**>

* Variables

| **参数** | **说明** | **类型** | **可选值** | **默认值** |
| --- | --- | --- | --- | --- |
| upload-before | 图片上传前的回调,会暂停图片上传function(file,done)，done用于继续图片上传，loading用于中断操作 | Function | — | — |
| upload-after | 图片上传后的回调,function(res,done)，done用于结束操作，loading用于中断操作 | Function | — | — |

* Prop attribute

如果dataType为arrat和string时，不存在该参数的设置

| **参数** | **说明** | **类型** | **可选值** | **默认值** |
| --- | --- | --- | --- | --- |
| label | 数据对象的图片地址 | String | — | label |
| value | 数据对象的图片名称 | String | — | value |

* propsHttp Attribute

如果dataType为arrat和string时，不存在name字段的配置

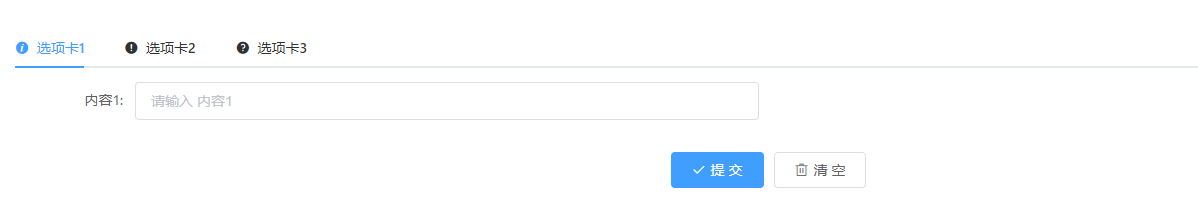
| **参数** | **说明** | **类型** | **可选值** | **默认值** |
| --- | --- | --- | --- | --- |
| url | 上传成功返回结构体的图片地址 | String | — | url |
| name | 上传成功返回结构体的图片的姓名 | String | — | name |
| res | 返回结构体的层次，例如返回(data:{url:'',name:''}),则res配置为data | String | — | - |
| fileName | 上传文件流时的名称 | String | — | file |

* canvasOption Attribute

| **参数** | **说明** | **类型** | **可选值** | **默认值** |
| --- | --- | --- | --- | --- |
| text | 字体的文字 | String | — | uway.cn |
| fontFamily | 字体类型 | String | — | microsoft yahei |
| color | 字体的颜色 | String | — | #999 |
| fontSize | 字体的大小 | String | — | 16 |
| opacity | 文字的透明度 | String | — | 100 |
| bottom | 文字距离图片底部的距离 | String | — | 10 |
| right | 文字距离图片右边的距离 | String | — | 10 |
| ratio | 压缩图片比率 | String | 0-1(可以是小数) | 1 |

## 选项卡

* 普通用法



集成了对form组件的封装，只需要在option中没一个选项卡column中配置option的form组件结构体即可，当然你觉的如果这个结构体太庞大的话，可以如下例子采用分组引入js，提交时候submit中返回当前选项卡的内容，如果你想获取全部选项卡对象，则给组件加一个v-model

<**uvue-tabs** :option="option" v-model="form" @submit="handleSubmit"></**uvue-tabs**>

<**script**>

**var** tab1Form ={

column:[{

label:'内容1',

prop:'text1'

}]

}

**var** tab2Form ={

column:[{

label:'内容2',

prop:'text2'

}]

}

**var** tab3Form ={

column:[{

label:'内容3',

prop:'text3'

}]

}

**export** **default** {

data(){

**return** {

form:{},

option:{

column: [{

icon:'el-icon-info',

label: '选项卡1',

prop: 'tab1',

option:tab1Form,

}, {

icon:'el-icon-warning',

label: '选项卡2',

prop: 'tab2',

option:tab2Form,

}, {

icon:'el-icon-question',

label: '选项卡3',

prop: 'tab3',

option:tab3Form,

}]

}

}

},

methods:{

handleSubmit(form){

**this**.$message.success(JSON.stringify(form))

}

}

}

</**script**>

* 分组



按照form的分组结构配置每一个选项卡即可

<**uvue-tabs** :option="option0" v-model="form" @submit="handleSubmit"></**uvue-tabs**>

<**script**>

**var** tab4Form ={

group:[

{

icon:'el-icon-info',

label: '分组1',

column:[{

label:'内容1',

prop:'text1'

}]

},

{

icon:'el-icon-info',

label: '分组2',

column:[{

label:'内容2',

prop:'text2'

}]

}

]

}

**var** tab5Form ={

group:[

{

icon:'el-icon-info',

label: '分组3',

column:[{

label:'内容3',

prop:'text3'

}]

},{

icon:'el-icon-info',

label: '分组4',

column:[{

label:'内容4',

prop:'text4'

}]

}

]

}

**export** **default** {

data(){

**return** {

form:{},

option0:{

column: [{

icon:'el-icon-info',

label: '选项卡1',

prop: 'tab1',

option:tab4Form,

}, {

icon:'el-icon-warning',

label: '选项卡2',

prop: 'tab2',

option:tab5Form,

}]

}

}

},

methods:{

handleSubmit(form){

**this**.$message.success(JSON.stringify(form))

}

}

}

</**script**>

## 多级联动



cascaderItem为需要联动的子选择框prop值,注意删除key之间的空格

<**uvue-form** :option="option" v-model="form" @submit="handleSubmit"></**uvue-form**>

<**script**>

**var** baseUrl = 'https://xxxx/api/area'

**export** **default** {

data(){

**return** {

form:{

province: '110000',

city: '110100',

area: '110101'

},

option:{

column: [{

label: '省份',

prop: 'province',

type: 'select',

props: {

label: 'name',

value: 'code'

},

cascaderItem: ['city', 'area'],

dicUrl: `${baseUrl}/getProvince`,

rules: [

{

required: true,

message: '请选择省份',

trigger: 'blur'

}

]

},

{

label: '城市',

prop: 'city',

type: 'select',

props: {

label: 'name',

value: 'code'

},

row: true,

dicFlag: false,

dicUrl: `${baseUrl}/getCity/{ {key}}`,

rules: [

{

required: true,

message: '请选择城市',

trigger: 'blur'

}

]

},

{

label: '地区',

prop: 'area',

type: 'select',

props: {

label: 'name',

value: 'code'

},

dicFlag: false,

dicUrl: `${baseUrl}/getArea/{ {key}}`,

rules: [

{

required: true,

message: '请选择地区',

trigger: 'blur'

}

]

}]

}

}

},

methods:{

handleSubmit(form){

**this**.$message.success(JSON.stringify(**this**.form))

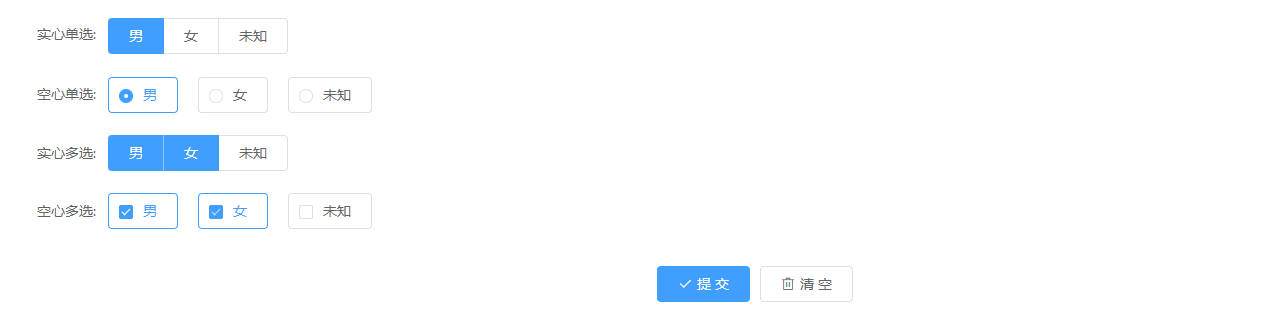
}

}

}

</**script**>

## 单选多选类型



自由组合button和border可以达到不同的样式效果

<**uvue-form** :option="option" v-model="obj"></**uvue-form**>

<**script**>

**const** DIC=[{

label:'男',

value:0

},{

label:'女',

value:1

},{

label:'未知',

value:''

}]

**export** **default** {

data() {

**return** {

obj: {

radio1:0,

radio2:0,

checkbox1:[0,1],

checkbox2:[0,1]

},

option: {

column: [

{

label: '实心单选',

prop: 'radio1',

type: 'radio',

span:24,

button:true,

dicData:DIC

},{

label: '空心单选',

prop: 'radio2',

type: 'radio',

span:24,

border:true,

dicData:DIC

},{

label: '实心多选',

prop: 'checkbox1',

type: 'checkbox',

span:24,

button:true,

dicData:DIC

},{

label: '空心多选',

prop: 'checkbox2',

span:24,

type: 'checkbox',

border:true,

dicData:DIC

}

]

}

}

}

}

</**script**>

## 选择框

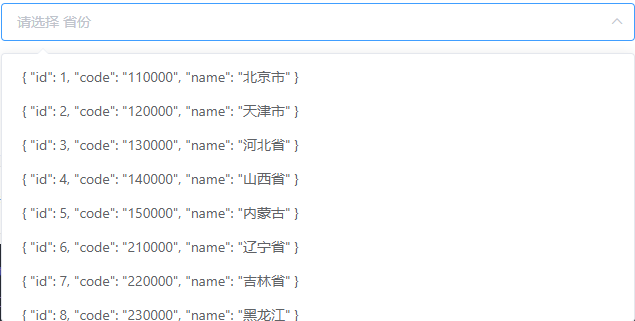
* Attributes

| **参数** | **说明** | **类型** | **可选值** | **默认值** |
| --- | --- | --- | --- | --- |
| typeslot | 子组件自定义 | Boolean | true/false | false |

* Scoped Lot

| **name** | **说明** |
| --- | --- |
| 列表单的名称+Type(nameType) | 表单自定义列的内容，参数为{item,label,value} |

* 普通用法



配置typeslot卡槽开启即可自定义下拉框的内容,typeformat配置回显的内容,但是你提交的值还是value并不会改变

<**uvue-form** :option="option" v-model="form" @submit="handleSubmit">

<**template** slot="provinceType" slot-scope="{item,value,label}">

<**span**></**span**>

</**template**>

</**uvue-form**>

<**script**>

**var** baseUrl = 'https://xxxx/api/area'

**export** **default** {

data() {

**return** {

form: {},

option: {

column: [

{

label: '省份',

prop: 'province',

type: 'select',

props: {

label: 'name',

value: 'code'

},

typeslot: true,

dicUrl: `${baseUrl}/getProvince`,

tpyeformat(item, label, value) {

**return** `值:${item[label]}-名:${item[value]}`

},

rules: [

{

required: true,

message: '请选择省份',

trigger: 'blur'

}

]

}

]

}

}

},

methods:{

handleSubmit(form){

**this**.$message.success(JSON.stringify(**this**.form))

}

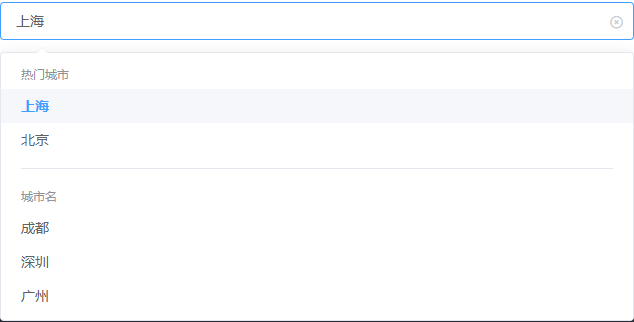
}

}

</**script**>

* 分组

配置group为true即可开启分组模式



<**uvue-form** :option="option" v-model="obj"></**uvue-form**>

<**script**>

**export** **default** {

data() {

**return** {

obj: {

select:'Shanghai'

},

option: {

column: [

{

label: '分组',

prop: 'select',

type: 'select',

group: true,

dicData: [{

label: '热门城市',

groups: [{

value: 'Shanghai',

label: '上海'

}, {

value: 'Beijing',

label: '北京'

}]

}, {

label: '城市名',

groups: [{

value: 'Chengdu',

label: '成都'

}, {

value: 'Shenzhen',

label: '深圳'

}, {

value: 'Guangzhou',

label: '广州'

}, {

value: 'Dalian',

label: '大连'

}]

}]

}

]

}

}

}

}

</**script**>

# 表格组件

## 文档

* Variables

| **参数** | **说明** | **类型** | **可选值** | **默认值** |
| --- | --- | --- | --- | --- |
| before-open | 打开前的回调，会暂停Dialog的  打开function(done,type)，done  用于关闭Dialog,type为当前窗口  的类型 | Function | — | — |
| before-close | 关闭前的回调，会暂停Dialog的  关闭function(done,type)，done  用于关闭Dialog,type为当前窗口  的类型 | Function | — | — |
| cell-class-name | 单元格的className的回调方法，  也可以使用字符串为所有单元格设  置一个固定的className,function(  {row,column,rowIndex,columnIndex})  /String | Function | - | - |
| data | 显示的数据 | Array | — | — |
| header-cell-class-name | header-cell-class-name表头单元格的  className的回调方法，也可以使用  字符串为所有表头单元格设置一个固  定的className,function({row,column,  rowIndex,columnIndex})/String | Function | - | - |
| option | 组件配置属性，详情见下面Option属性 | Object | — | — |
| page | 分页配置选项，具体看下表 | Object | — | — |
| row-class-name | 行的className的回调方法，也可以使用字符串为所有行设置一个固定的className,function({row,rowIndex}) | Function | - | - |
| span-method | 合并行或列的计算方法Function({row,column,rowIndex,columnIndex}) | Function | - | - |
| summary-method | 自定义的合计计算方法Function({columns,data}) | Function | - | - |
| table-loading | 表格等待框的控制 | Boolean | true/false | false |
| upload-before | 图片上传前的回调,会暂停图片上  传function(file,done)，done用于  继续图片上传，loading用于中断  操作 | Function | — | — |
| upload-after | 图片上传后的回调,function(res,done)，done用  于结束操作，loading用于中断  操作 | Function | — | — |

* Option Attributes

| **参数** | **说明** | **类型** | **可选值** | **默认值** |
| --- | --- | --- | --- | --- |
| addBtn | 添加按钮 | Boolean | true/false | true |
| addRowBtn | 表格行内添加按钮 | Boolean | true/false | false |
| align | 表格列齐方式 | String | left/center/right | left |
| border | 表格边框 | Boolean | true/false | false |
| calcHeight | 表格高度差（主要用于减去其他部分让表格高度自适应） | Number | — | auto |
| cancelBtnTitle | 弹出框取消按钮标题 | String | — | 取消 |
| columnBtn | 列显隐按钮 | Boolean | true/false | true |
| dataType | 数据的类型转换也就是数组和用逗号隔开的字符串（适用于checkbox/select多选/tree多选/img/array/upload） | String | number/string | - |
| cellBtn | 表格单元格可编辑（当column中有搜索的属性中有cell为true的属性启用，只对type为select和input有作用) | Boolean | true/false | true |
| dateBtn | 日期组件按钮 | Boolean | true/false | false |
| cancelBtn | 行编辑取消按钮 | Boolean | true/false | true |
| dateDefault | 日期控件默认的值 | Boolean | true/false | false |
| dicData | 传入本次需要的静态字典（在column中dicData写对象key值即可加载） | Object | - | - |
| dicMethod | 传入字典的请求方式 | Stirng | get/post | get |
| dicQuery | 传入字典的请求参数 | Object | - | - |
| dicUrl | 字典的网络请求接口（例如配置/xxx/xx/{\{key}},这样的格式，在column中dicData自动匹配prop字段名） | String | - | - |
| delBtn | 行内删除按钮 | Boolean | true/false | true |
| defaultSort | 表格的排序字段{prop:'date',order:'descending'}prop默认排序字段，order排序方式 | Object | — | - |
| dialogFullscreen | 是否为全屏Dialog | Boolean | true/false | false |
| dialogEscape | 是否可以通过按下ESC关闭Dialog | Boolean | true/false | true |
| dialogClickModal | 是否可以通过点击modal关闭Dialog | Boolean | true/false | true |
| dialogCloseBtn | 是否显示关闭按钮 | Boolean | true/false | true |
| dialogModal | 是否需要遮罩层 | Boolean | true/false | true |
| dialogTop | 弹出表单的弹窗具体顶部的距离 | String/Number | - | 35 |
| dialogType | 弹出表单的类型 | String | dialog / drawer | dialog |
| dialogWidth | 弹出表单的弹窗宽度 | String/Number | - | 60% |
| dialogHeight | 弹出表单的弹窗高度 | String/Number | - | 400 |
| defaultExpandAll | 是否默认展开所有行，"expand"为true的时候有效 | Boolean | true/false | false |
| expandRowKeys | 可以通过该属性设置目前的展开行，需要设置 rowKey 属性才能使用，该属性为展开行的 keys 数组。 | Array | - | - |
| editBtn | 行内编辑按钮 | Boolean | true/false | true |
| empty-text | 空数据时显示的文本内容，也可以通过slot="empty"设置 | String | - | 暂无数据 |
| expand | 是否展开折叠行 | Boolean | true/false | false |
| expandWidth | 是否展开折叠行宽度 | Number | - | 50 |
| expandFixed | 是否展开折叠行冻结 | Boolean | true/false | true |
| excelBtn | 打印按钮 | Boolean | true/false | false |
| filterBtn | 自定义过滤按钮 | Boolean | true/false | false |
| formWidth | 表单的宽度 | String/Number | - | 100% |
| height | 表格高度 | Number | — | auto |
| header | 头部显隐 | Boolean | true/false | true |
| index | 是否显示表格序号（根据分页会自动计算，比如每页10行，到了第二页就会从11开始记数） | Boolean | true/false | false |
| indexLabel | 序号的标题 | String | — | # |
| indexWidth | 序号的宽度 | Number | — | 50 |
| indexFixed | 序号的是否冻结 | Boolean | true / false | true |
| rowKey | 行数据的 Key的主键，用于其他相关操作 | String | — | id |
| indeterminate | 设置 indeterminate 状态，只负责样式控制 如果选择一半，不会出现半选 | Boolean | — | false |
| labelWidth | 弹出表单的label宽度 | Number | - | 110 |
| maxHeight | 表格最大高度 | Number | — | auto |
| menu | 是否显示操作菜单栏 | Boolean | true/false | true |
| menuWidth | 操作菜单栏的宽度 | Number | - | 240 |
| menuXsWidth | 手机端时操作菜单栏的宽度 | Number | - | 100 |
| menuAlign | 菜单栏对齐方式 | String | left/center/right | left |
| menuType | 操作栏菜单按钮类型 | String | button/icon/text/menu | text |
| menuBtnTitle | 菜单按钮的文字 | String | - | 功能 |
| pageSize | 每页显示多少条 | String | — | — |
| pageSizes | 分页的数组分段 | Array | — | [10,20,30,50,100] |
| printBtn | 打印按钮 | Boolean | true/false | false |
| refreshBtn | 刷新按钮 | Boolean | true/false | true |
| saveBtn | 保存按钮 | Boolean | true/false | true |
| updateBtn | 更新按钮 | Boolean | true/false | true |
| cancalBtn | 取消按钮 | Boolean | true/false | true |
| saveBtnTitle | 弹出新增按钮标题 | String | — | 新增 |
| selection | 行可勾选 | Boolean | true/false | false |
| selectionWidth | 行可勾选的宽度 | Number | - | 50 |
| selectionFixed | 行可勾选是否冻结 | Boolean | true/false | true |
| searchBtn | 搜索显隐按钮（当column中有搜索的属性，或则searchsolt为true时自定义搜索启动起作用） | Boolean | true/false | true |
| selectable | 仅对 selection为true 的列有效，类型为 Function，Function 的返回值用来决定这一行的 CheckBox 是否可以勾选 | Function(row, index) | — | — |
| reserveSelection | 仅对 selection为true 的列有效，类型为 Boolean，为 true 则会在数据更新之后保留之前选中的数据（需指定 rowKey） | Boolean | true/false | true |
| selectClearBtn | 清空选中按钮（当selection为true起作用） | Boolean | true/false | true |
| showHeader | 是否显示表格的表头 | Boolean | true/false | true |
| showSummary | 是否在表尾显示合计行 | Boolean | true/false | false |
| size | 控件大小 | String | medium/small/mini | medium |
| sumColumnList | 表格合计需要配置的字段 | Array | - | - |
| stripe | 是否显示表格的斑马条纹 | Boolean | true/false | false |
| tip | 弹窗编辑文字提示 | String | - | - |
| tipPlacement | 弹窗编辑文字提示展示方向 | String | top-start/top/top-end/left-start/leftleft-end/right-start/right/right-endbottom-start/bottom/bottom-end | - |
| title | 表格标题 | String | — | 表格标题 |
| checkStrictly | 不遵循父子规则 | Boolean | true/false | true |
| updateBtnTitle | 弹出框更新按钮标题 | String | — | 修改 |
| viewBtn | 查看按钮 | Boolean | true/false | false |
| width | 表格宽度 | Number | — | 100% |

* Props Attributes

| **参数** | **说明** | **类型** | **可选值** | **默认值** |
| --- | --- | --- | --- | --- |
| label | 字典的名称属性值 | String | — | — |
| value | 字典的值属性值 | String | — | — |
| children | 字典的子属性值 | String | — | — |
| res | 网络字典返回的数据格式 | String | — | — |

* Type Attributes

这是type属性可以配置的组件，当然你也可以自定义，参考第三方组件导入

| **参数** | **类型** |
| --- | --- |
| array | 数组框 |
| color | 颜色选择框 |
| cascader | 级联框 |
| checkbox | 多选框 |
| date | 日期框 |
| datetime | 日期时间框 |
| daterange | 日期范围 |
| datetimerange | 日期时间范围 |
| dates | 多个日期 |
| dynamic | 子表单 |
| icon | 图标选择框 |
| input | 输入框 |
| img | 图片框 |
| month | 月 |
| password | 密码框 |
| radio | 单选框 |
| select | 选择框 |
| switch | 开关框 |
| slider | 滑动框 |
| textarea | 文本框 |
| rate | 评价框 |
| time | 时间框 |
| timerange | 时间范围 |
| tree | 树框 |
| url | 超链 |
| week | 周 |
| year | 年 |

* Format Attributes

使用format指定输入框的格式；使用valueFormat指定绑定值的格式。

默认情况下，组件接受并返回Date对象。以下为可用的格式化字串，以UTC2017年1月2日03:04:05为例(请注意大小写)：

| **格式** | **含义** | **备注** | **举例** |
| --- | --- | --- | --- |
| yyyy | 年 |  | 2017 |
| M | 月 | 不补0 | 1 |
| MM | 月 |  | 01 |
| W | 周 | 仅周选择器的format可用；不补0 | 1 |
| WW | 周 | 仅周选择器的format可用 | 01 |
| d | 日 | 不补0 | 2 |
| dd | 日 |  | 02 |
| H | 小时 | 24小时制；不补0 | 3 |
| HH | 小时 | 24小时制 | 03 |
| h | 小时 | 12小时制，须和A或a使用；不补0 | 3 |
| hh | 小时 | 12小时制，须和A或a使用 | 03 |
| m | 分钟 | 不补0 | 4 |
| mm | 分钟 |  | 04 |
| s | 秒 | 不补0 | 5 |
| ss | 秒 |  | 05 |
| A | AM/PM | 仅format可用，大写 | AM |
| a | am/pm | 仅format可用，小写 | am |
| timestamp | JS时间戳 | 仅value-format可用；组件绑定值为number类型 | 1483326245000 |

* Column Attributes

| **参数** | **说明** | **类型** | **可选值** | **默认值** |
| --- | --- | --- | --- | --- |
| accept | 为upload时文件类型 | String/Array | — | - |
| addDisabled | 表单新增时是否禁止 | Boolean | true/false | false |
| addDisplay | 表单新增时是否可见 | Boolean | true/false | true |
| align | 对其方式 | String | left/center/right | left |
| append | 后面的辅助文字（当type为input）） | String/Number | - | - |
| cell | 行可编辑时，当前列是否可编辑 | Boolean | true/false | false |
| clearable | 表单清空 | Boolean | true/false | false |
| disabled | 全部是否禁止 | Boolean | true/false | false |
| editDisabled | 表单编辑时是否禁止 | Boolean | true/false | false |
| editDisplay | 表单编辑是否可见 | Boolean | true/false | true |
| endPlaceholder | 日期范围结束占位符 | String | - | - |
| filesize | 为upload时文件的大小 | Number | - | - |
| filterMultiple | 数据过滤的选项是否多选 | Boolean | true/false | true |
| filters | 数据过滤的选项，数组格式，数组中的元素需要有text和value属性。 | Array[{text,value}] | - | - |
| filterMethod | 数据过滤使用的方法，如果是多选的筛选项，对每一条数据会执行多次，任意一次返回true就会显示。 | Function(value,row,column) | - | - |
| fixed | 冻结列 | Boolean | true/false | false |
| filterable | 是否可以输入搜索。 | Boolean | true/false | false |
| formatter | 用来格式化内容 | Function(row,value,label,column) | - | - |
| formHeight | 表单行高度 | Number | - | auto |
| format | 显示值时间格式（当type为date/time/datetime/daterangetimerange/datetimerange/week/month/year/dates | - | - | - |
| formslot | 表单自定义 | Boolean | true/false | false |
| typeslot | 表单组件自定义 | Boolean | true/false | false |
| formWidth | 表单行高度 | Number/String | - | 50% |
| hide | 隐藏列 | Boolean | true/false | true |
| label | 列名称 | String | — | - |
| maxRows | 最大行（当type为textarea） | Number | - | 4 |
| minRows | 最小行（当type为textarea） | Number | - | 2 |
| minWidth | 列最小宽度 | String | — | auto |
| multiple | 多选（当type为select/tree时） | Boolean | true/false | false |
| overHidden | 超出隐藏 | Boolean | true/false | false |
| placeholder | 辅助语 | String | — | 请选择/请输入+label |
| prepend | 前面的辅助文字（当type为input） | String/Number | - | - |
| precision | 数字框输入精度（当type为number时） | Number | - | 2 |
| prop | 列字段 | String | — | - |
| readonly | 只读 | Boolean | true/false | false |
| rules | 表单规则,参考ele表单规则配置 | Object | - | - |
| size | 表单大小 | String | medium/small/mini | medium |
| showColumn | 是否加入动态现隐列 | Boolean | true/false | true |
| sortable | 排序 | Boolean | true/false | false |
| slot | 列自定义 | Boolean | true/false | false |
| span | 表单栅列 | Number | - | 12 |
| gutter | 项之间的间 | Number | - | 20 |
| startPlaceholder | 日期范围开始占位符 | String | - | - |
| type | 类型 | String | input/select/radio/checkbox/textarea/cascader/date/time/datetime/daterange/timerange/datetimerange/week/month/year/dates/password/switch/tree | input |
| valueDefault | 表单的默认值 | Number/String | - | - |
| valueFormat | 真实值的时间格式（当type为date/time/datetime/daterangetimerange/datetimerange/week/month/year/dates） | - | - | - |
| display | 全部是否可见 | Boolean | true/false | true |
| viewDisplay | 表单查看是否可见 | Boolean | true/false | true |
| tags | select组件多选时是否超出后合并标签 | Boolean | true/false | false |
| width | 列宽度 | String | — | auto |

* Search Attributes

| **参数** | **说明** | **类型** | **可选值** | **默认值** |
| --- | --- | --- | --- | --- |
| searchDefault | 搜索表单的默认值 | Number/String | - | - |
| searchPlaceholder | 搜索框的辅助文字 | String | - | label |
| searchSpan | 搜索框栅列 | Number | - | 8 |
| searchGutter | 搜索框的间距 | String | - | 20 |
| searchLabelWidth | 搜索框的文字宽度 | Number | - | 80 |
| searchClearable | 搜索框的清除按钮 | Boolean | true/false | false |
| searchMmultiple | 搜索表单的是否多选 | Boolean | true/false | false |
| searchFilterable | 是否可以输入搜索。 | Boolean | true/false | false |
| searchFilterMethod | 数据过滤使用的方法，如果是多选的筛选项，对每一条数据会执行多次，任意一次返回true就会显示。 | Function(value,row,column) | - | - |
| searchShow | 首次加载是否显示搜索 | Boolean | true/false | true |
| searchTipPlacement | 搜索文字提示展示方向 | String | top-start/top/top-end/left-start/leftleft-end/right-start/right/right-endbottom-start/bottom/bottom-end | - |
| searchTip | 弹窗编辑文字提示 | String | - | - |

* Events

| **事件名** | **说明** | **参数** |
| --- | --- | --- |
| current-change | 切换页面时触发该事件 | page |
| current-row-change | 当表格的当前行发生变化的时候会触发该事件，如果要高亮当前行，请打开表格的highlightCurrentRow属性为true | currentRow,oldCurrentRow |
| select | 单个选择回调 | selection, row |
| selectAll | 勾选全选 | selection |
| date-change | 表格日期回调方法 | date |
| filter-change | 过滤回调函数 | params |
| cell-mouse-enter | 当单元格 hover 进入时会触发该事件 | row, column, cell, event |
| cell-mouse-leave | 当单元格 hover 退出时会触发该事件 | row, column, cell, event |
| cell-click | 当某个单元格被点击时会触发该事件 | row, column, cell, event |
| cell-dblclick | 当某个单元格被双击击时会触发该事件 | row, column, cell, event |
| row-update | 编辑数据后确定触发该事件 | row,index,done,loading |
| row-save | 新增数据后点击确定触发该事件 | row,done,loading |
| row-del | 行数据删除时触发该事件 | row,index |
| row-click | 当某一行被点击时会触发该事件 | row,event,column |
| row-dblclick | 当某一行被双击时会触发该事件 | row,column |
| refresh-change | 点击刷新按钮触发该事件 | page |
| size-change | 页面每页显示的条数触发该事件 | pageSize |
| sort-change | 调用排序后触发该事件 | list |
| search-change | 点击搜索后触发该事件 | params |
| search-reset | 清空搜索回调方法 | - |
| selection-change | 当选择项发生变化时会触发该事件 | selection |
| toggleRowExpansion | 用于可展开表格，切换某一行的展开状态，如果使用了第二个参数，则是设置这一行展开与否（expanded 为 true 则展开 | row, expanded |

* Methods

|  |  |  |
| --- | --- | --- |
| **方法名** | **说明** | **参数** |
| init | 初始化参数（比如服务端加载option，需要调用该内置函数初始化） |  |
| clearValidate | 清空表格钟表单验证。 |  |
| rowAdd | 打开表单新增窗口 |  |
| rowSave | 表单保存调用 |  |
| rowUpdate | 表单更新调用 |  |
| rowEdit | 打开表单编辑窗口 | row,index |
| rowCell | 打开行编辑 | row,index |
| rowPrint | 打印表格 |  |
| rowExcel | 导出表格 |  |
| resetForm | 清空表单数据 |  |
| setCurrentRow | 用于单选表格，设定某一行为选中行，如果调用时不加参数，则会取消目前高亮行的选中状态。 | row |
| selectClear | 用于多选表格，清空用户的选择。 |  |
| searchReset | 清空搜索栏目的值 |  |
| toggleSelection | 用于多选表格，切换所有行的选中状态。 |  |

* Scope Slot

| **name** | **说明** |
| --- | --- |
| 列的名称 | 列自定义列的内容，参数为{row,label,dic,$index} |
| 列表单的名称+Form(nameForm) | 表单自定义列的内容，参数为{row,label,dic,$index} |
| 列表单的名称+Type(nameType) | 表单自定义列的内容，参数为{row,label,value} |
| empty | 暂无数据的自定义卡槽 |
| expand | 折叠板的自定义内容,参数为{row,label,dic,$index} |
| menu | 菜单的操作按钮自定义内容,参数为{row,label,dic,$index} |
| menuForm | 表单操作按钮的自定义内容,参数为{row,label,dic,$index} |
| menuLeft | 表格头部左侧内容 |
| menuRight | 表格头部右侧内容 |
| menuBtn | 操作栏目下拉菜单自定义(要用el-dropdown-item组件包裹起来),参数为{row,label,dic,$index} |
| search | 搜索栏目自定义内容 |
| searchMenu | 搜索栏目菜单自定义内容 |

## 基础

普通用法：



<**el-row** style="margin-bottom:20px;font-size:15px">

<**el-col** :span="4">显示边框: <**el-switch** size="small" v-model="showBorder"> </**el-switch**></**el-col**>

<**el-col** :span="4">显示斑马纹: <**el-switch** size="small" v-model="showStripe"> </**el-switch**></**el-col**>

<**el-col** :span="4">显示索引: <**el-switch** size="small" v-model="showIndex"> </**el-switch**></**el-col**>

<**el-col** :span="4">显示多选框: <**el-switch** size="small" v-model="showCheckbox"> </**el-switch**></**el-col**>

<**el-col** :span="4">显示表头: <**el-switch** size="small" v-model="showHeader"> </**el-switch**></**el-col**>

<**el-col** :span="4">显示分页: <**el-switch** size="small" v-model="showPage"> </**el-switch**></**el-col**>

</**el-row**>

<**el-row** style="margin-bottom:20px">

<**el-radio-group** v-model="sizeValue">

<**el-radio** label="">默认</**el-radio**>

<**el-radio** label="small">small</**el-radio**>

<**el-radio** label="mini">mini</**el-radio**>

</**el-radio-group**>

</**el-row**>

<**uvue-crud** :data="data" :option="option0"></**uvue-crud**>

<**script**>

**export** **default** {

data() {

**return** {

obj:{},

data: [

{

name:'张三',

sex:'男'

}, {

name:'李四',

sex:'女'

}, {

name:'王五',

sex:'女'

}, {

name:'赵六',

sex:'男'

}

],

showBorder: false,

showStripe: false,

showHeader: true,

showIndex: true,

showCheckbox: false,

showPage:false,

sizeValue:''

}

},

computed: {

option0(){

**return**{

title:'表格的标题',

border:**this**.showBorder,

stripe:**this**.showStripe,

showHeader:**this**.showHeader,

index:**this**.showIndex,

size:**this**.sizeValue,

selection:**this**.showCheckbox,

page:**this**.showPage,

align:'center',

menuAlign:'center',

column:[

{

label:'姓名',

prop:'name'

},

{

label:'性别',

prop:'sex'

}

]

}

}

}

}

</**script**>

丰富表格



合并菜单



配置menuType为menu表格的操作栏目菜单合并，menuBtn卡槽为自定义卡槽,delBtn和editBtn会消失,dateBtn控件的dateDefault为true时首次进来会加载回调方法,

<**uvue-crud** :data="data" :option="option1" v-model="obj" @date-change="dateChange">

<**template** slot-scope="scope" slot="menuBtn">

<**el-dropdown-item** divided @click.native="tip">自定义按钮</**el-dropdown-item**>

</**template**>

<**template** slot-scope="scope" slot="menu">

<**el-button** size="small" @click.native="tip">自定义按钮</**el-button**>

</**template**>

</**uvue-crud**>

<**script**>

**export** **default** {

data() {

**return** {

obj:{},

data: [

{

name:'张三',

sex:'男',

date:'1994-02-23 00:00:00'

}, {

name:'李四',

sex:'女',

date:'1994-02-23 00:00:00'

}, {

name:'王五',

sex:'女',

date:'1994-02-23 00:00:00'

}, {

name:'赵六',

sex:'男',

date:'1994-02-23 00:00:00'

}

],

option1:{

title:'表格的标题',

menuType:'menu',

page:false,

align:'center',

dateBtn:true,

dateDefault:true,

menuAlign:'center',

column:[

{

label:'姓名',

prop:'name'

},

{

label:'性别',

prop:'sex'

},{

label: "日期",

prop: "date",

type: "date",

format: "yyyy-MM-dd hh:mm:ss",

valueFormat: "yyyy-MM-dd hh:mm:ss",

}

]

}

}

},

methods: {

dateChange(date){

**this**.$message.success(date);

},

tip(){

**this**.$message.success('自定义按钮');

}

}

}

</**script**>

文本菜单



配置menuType为text时表格操作栏为文本按钮

<**uvue-crud** :data="data" :option="option2" v-model="obj" @date-change="dateChange">

<**template** slot-scope="scope" slot="menu">

<**el-button** type="text" size="small" @click.native="tip">自定义按钮</**el-button**>

</**template**>

</**uvue-crud**>

<**script**>

**export** **default** {

data() {

**return** {

obj:{},

data: [

{

name:'张三',

sex:'男',

date:'1994-02-23 00:00:00'

}, {

name:'李四',

sex:'女',

date:'1994-02-23 00:00:00'

}, {

name:'王五',

sex:'女',

date:'1994-02-23 00:00:00'

}, {

name:'赵六',

sex:'男',

date:'1994-02-23 00:00:00'

}

],

option2:{

title:'表格的标题',

menuType:'text',

page:false,

align:'center',

dateBtn:true,

dateDefault:true,

menuAlign:'center',

column:[

{

label:'姓名',

prop:'name'

},

{

label:'性别',

prop:'sex'

},{

label: "日期",

prop: "date",

type: "date",

format: "yyyy-MM-dd hh:mm:ss",

valueFormat: "yyyy-MM-dd hh:mm:ss",

}

]

}

}

},

methods: {

dateChange(date){

**this**.$message.success(date);

},

tip(){

**this**.$message.success('自定义按钮');

}

}

}

</**script**>

图标菜单



配置menuType为icon时表格操作栏为图标按钮

<**uvue-crud** :data="data" :option="option3" v-model="obj" @date-change="dateChange">

<**template** slot-scope="scope" slot="menu">

<**el-button** size="small" @click.native="tip" icon="el-icon-share"></**el-button**>

</**template**>

</**uvue-crud**>

<**script**>

**export** **default** {

data() {

**return** {

obj:{},

data: [

{

name:'张三',

sex:'男',

date:'1994-02-23 00:00:00'

}, {

name:'李四',

sex:'女',

date:'1994-02-23 00:00:00'

}, {

name:'王五',

sex:'女',

date:'1994-02-23 00:00:00'

}, {

name:'赵六',

sex:'男',

date:'1994-02-23 00:00:00'

}

],

option3:{

title:'表格的标题',

menuType:'icon',

page:false,

align:'center',

dateBtn:true,

dateDefault:true,

menuAlign:'center',

column:[

{

label:'姓名',

prop:'name'

},

{

label:'性别',

prop:'sex'

},{

label: "日期",

prop: "date",

type: "date",

format: "yyyy-MM-dd hh:mm:ss",

valueFormat: "yyyy-MM-dd hh:mm:ss",

}

]

}

}

},

methods: {

dateChange(date){

**this**.$message.success(date);

},

tip(){

**this**.$message.success('自定义按钮');

}

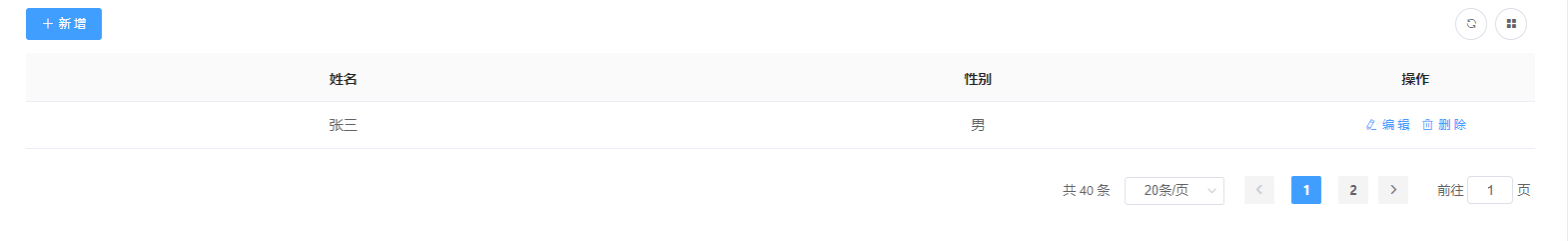
}

}

</**script**>

## 分页

普通分页



首次加载调用on-load方法加载数据，返回page分页对象信息,赋值page的total总条数即可

<**uvue-crud**

:data="data"

:option="option"

:page="page"

@on-load="onLoad"

></**uvue-crud**>

<**script**>

**export** **default** {

data() {

**return** {

page: {

pageSize: 20

},

data: [],

option: {

align: 'center',

menuAlign: 'center',

column: [

{

label: '姓名',

prop: 'name'

},

{

label: '性别',

prop: 'sex'

}

]

}

}

},

methods: {

onLoad(page) {

**this**.$message.success('分页信息:' + JSON.stringify(page))

**this**.page.total = 40

//模拟分页

**if** (page.currentPage === 1) {

**this**.data = [

{

name: '张三',

sex: '男'

}

]

} **else** **if** (page.currentPage == 2) {

**this**.data = [

{

name: '李四',

sex: '女'

}

]

}

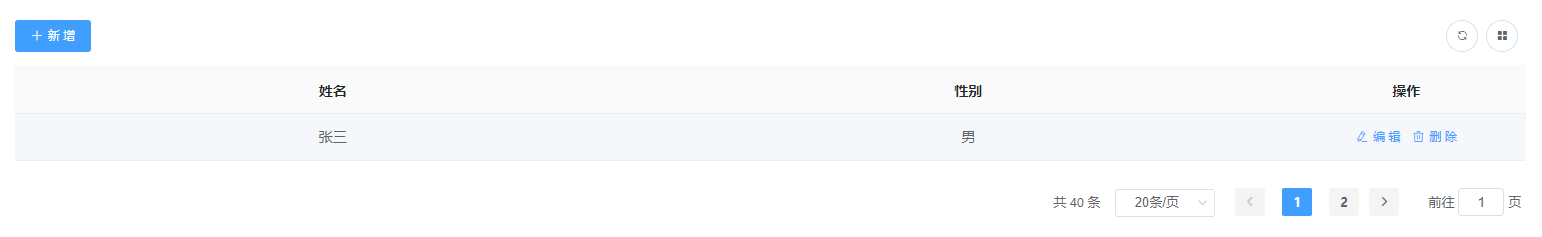
}

}

}

</**script**>

自定义分页：



实际的用法要后台配置，将后台返回值赋值给 page 中的属性， page就是分页对象注入，page 对象中的total为总页数，pageSizes为分页信息，currentPage为当前第几页，pageSize每一页加载多少条数据，点击页码会调用current-change方法回调当前页数，返回当前第几页，点击每页多少条会调size-change方法回调

<**uvue-crud**

:data="data"

:option="option"

:page="page"

@size-change="sizeChange"

@current-change="currentChange"

></**uvue-crud**>

<**script**>

**export** **default** {

data() {

**return** {

page: {

//pageSizes: [10, 20, 30, 40],默认

currentPage: 1,

total: 0,

pageSize: 10

},

data: [

{

name: '张三',

sex: '男'

},

{

name: '李四',

sex: '女'

}

],

option: {

align: 'center',

menuAlign: 'center',

column: [

{

label: '姓名',

prop: 'name'

},

{

label: '性别',

prop: 'sex'

}

]

}

}

},

created() {

**this**.getList()

},

methods: {

getList() {

**this**.page.total = 40

**if** (**this**.page.currentPage === 1) {

**this**.data = [

{

name: '张三',

sex: '男'

}

]

} **else** **if** (**this**.page.currentPage == 2) {

**this**.data = [

{

name: '李四',

sex: '女'

}

]

}

},

sizeChange(val) {

**this**.page.currentPage = 1

**this**.page.pageSize = val

**this**.getList()

**this**.$message.success('行数' + val)

},

currentChange(val) {

**this**.page.currentPage = val

**this**.getList()

**this**.$message.success('页码' + val)

}

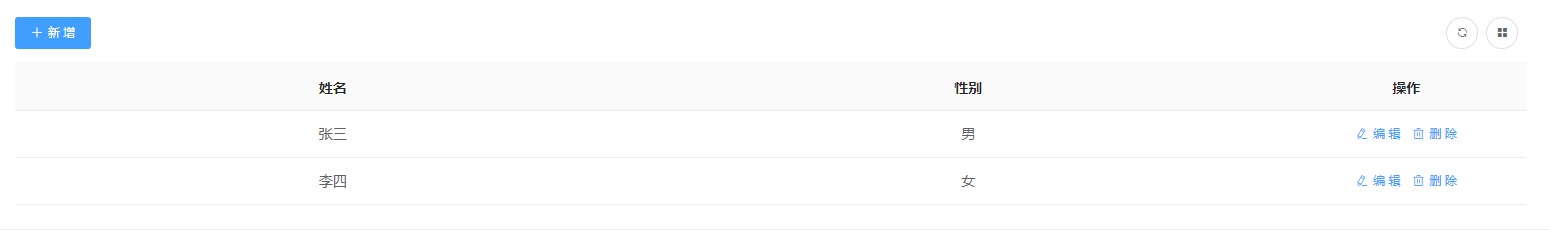
}

}

</**script**>

## 验证

普通用法：



配置验证字段的rules的数据对象即可，具体参考element中form表单校验的配置规则官方规则

<**uvue-crud** :data="data" :option="option" @row-save="rowSave" @row-update="rowUpdate"></**uvue-crud**>

<**script**>

**export** **default** {

data() {

**return** {

data: [

{

name:'张三',

sex:'男'

}, {

name:'李四',

sex:'女'

}

],

option:{

page:false,

align:'center',

menuAlign:'center',

column:[

{

label:'姓名',

prop:'name',

rules: [{

required: true,

message: "请输入姓名",

trigger: "blur"

}]

}, {

label:'性别',

prop:'sex',

rules: [{

required: true,

message: "请输入性别",

trigger: "blur"

}]

}

]

},

};

},

methods: {

rowSave(form,done){

**this**.$message.success('新增数据'+ JSON.stringify(form));

done();

}

rowUpdate(form,index,done){

**this**.$message.success('编辑数据'+ JSON.stringify(form)+'数据序号'+index);

done();

},

}

}

</**script**>

自定义验证：

<**uvue-crud** :data="data" v-model="obj" :option="option1" @row-save="rowSave" @row-update="rowUpdate"></**uvue-crud**>

<**script**>

**export** **default** {

data() {

**var** validatePass = **function**(rule, value, callback){

**if** (value === '') {

callback(**new** Error('请输入密码'));

} **else** {

callback();

}

};

**var** validatePass2 = **function**(rule, value, callback) {

**if** (value === '') {

callback(**new** Error('请再次输入密码'));

} **else** **if** (value !== **this**.obj.password) {

callback(**new** Error('两次输入密码不一致!'));

} **else** {

callback();

}

};

**return** {

obj:{},

data: [

{

name:'张三',

sex:'男'

}, {

name:'李四',

sex:'女'

}

],

option0:{

page:false,

align:'center',

menuAlign:'center',

column:[

{

label:'姓名',

prop:'name',

rules: [{

required: true,

message: "请输入姓名",

trigger: "blur"

}]

}, {

label:'性别',

prop:'sex',

rules: [{

required: true,

message: "请输入性别",

trigger: "blur"

}]

}

]

}, {

label:'密码',

prop:'password',

hide:true,

rules: [{ validator: validatePass, trigger: 'blur' }]

}, {

label:'确认密码',

prop:'oldpassword',

hide:true,

rules: [{ validator: validatePass2, trigger: 'blur' }]

}

};

},

methods: {

rowSave(form,done){

**this**.$message.success('新增数据'+ JSON.stringify(form));

done();

}

rowUpdate(form,index,done){

**this**.$message.success('编辑数据'+ JSON.stringify(form)+'数据序号'+index);

done();

},

}

}

</**script**>

## 方法

普通用法：



点击不用的按钮回调不用的方法，点击删除回调row-del方法,保存新数据回调rowSave方法,修改数据回调row-update方法,刷新回调refresh方法

<**uvue-crud** :data="data" :option="option"

@row-save="rowSave"

@row-update="rowUpdate"

@row-del="rowDel"

@refresh-change="refresh"></**uvue-crud**>

<**script**>

**export** **default** {

data() {

**return** {

data: [

{

name:'张三',

sex:'男'

}, {

name:'李四',

sex:'女'

}

],

option:{

page:false,

align:'center',

menuAlign:'center',

menuWidth:400,

viewBtn:true,

column:[

{

label:'姓名',

prop:'name'

}, {

label:'性别',

prop:'sex'

}

]

},

};

},

methods: {

rowSave(form,done,loading){

**this**.$message.success('模拟网络请求')

setTimeout(()=>{

**this**.$message.success('关闭按钮等待');

loading();

},1000)

setTimeout(()=>{

**this**.$message.success('新增数据'+ JSON.stringify(form));

done();

},2000)

},

refresh(val){

**this**.$message.success('刷新回调,当前分页对象'+ JSON.stringify(val));

},

rowDel(form,index){

**this**.$message.success('删除数据'+ JSON.stringify(form));

},

rowUpdate(form,index,done,loading){

**this**.$message.success('模拟网络请求')

setTimeout(()=>{

**this**.$message.success('关闭按钮等待');

loading();

},1000)

setTimeout(()=>{

**this**.$message.success('编辑数据'+ JSON.stringify(form)+'数据序号'+index);

done();

},2000)

},

}

}

</**script**>

表格中可显示的列字段进行排序，设置属性sortable接受一个Boolean的属性，设置为true即可开启排序，回调sort-change方法返回排序后的数据

<**uvue-crud** :data="data" :option="option" @sort-change="sortChange"></**uvue-crud**>

<**script**>

**export** **default** {

data() {

**return** {

data: [

{

sortable:true,

name:'张三',

sex:'男'

}, {

name:'李四',

sex:'女'

}

],

option:{

border:true,

page:false,

align:'center',

menuAlign:'center',

column:[

{

label:'姓名',

prop:'name'

}, {

label:'性别',

prop:'sex'

}

]

},

};

},

methods: {

sortChange(val){

**this**.$message.success('排序'+ JSON.stringify(val));

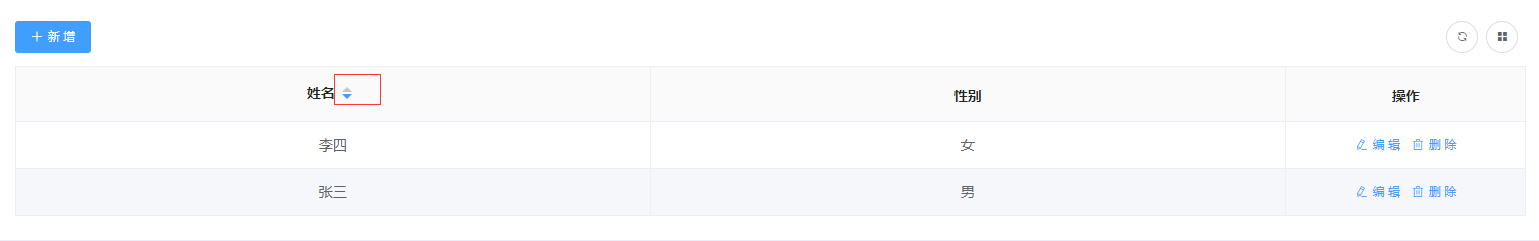
}

}

}

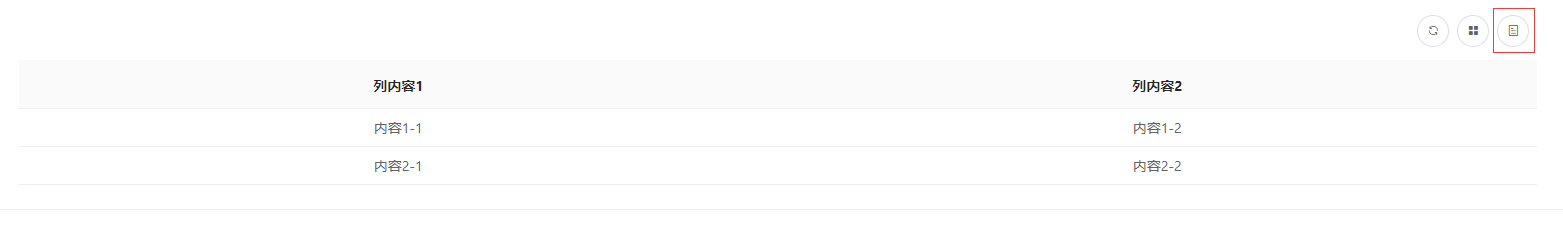
</**script**>

## 排序



## 筛选

普通用法：





filterBtn默认为true，可以自定义过滤条件，根据filter-change函数回调

<**uvue-crud** :option="option" :data="data" @filter-change="filterChange"></**uvue-crud**>

<**script**>

**export** **default** {

data(){

**return** {

data:[{

text1:'内容1-1',

text2:'内容1-2'

},{

text1:'内容2-1',

text2:'内容2-2'

}],

option:{

filterBtn:true,

align:'center',

addBtn:false,

menu:false,

column: [{

label: '列内容1',

prop: 'text1',

}, {

label: '列内容2',

prop: 'text2',

}]

}

}

},

methods:{

filterChange(result) {

**this**.$message.success('过滤器' + JSON.stringify(result))

},

}

}

</**script**>

## 字典

普通用法：



本地字典只要配置dicData为一个Array数组即可，便会自动加载字典到对应的组件中，注意返回的字典中value类型和数据的类型必须要对应，比如都是字符串或则都是数字。

<**uvue-crud** :data="data" :option="option" ></**uvue-crud**>

<**script**>

**export** **default** {

data() {

**return** {

data: [

{

name:'张三',

sex:'男',

grade:1,

cascader:[0,1],

checkbox:[0,1],

tree:0,

}, {

name:'李四',

sex:'女',

grade:0,

cascader:[0,2],

checkbox:[0,1],

tree:1

}

],

option:{

page:false,

align:'center',

menuAlign:'center',

column:[

{

label:'姓名',

prop:'name',

}, {

label:'性别',

prop:'sex'

},{

label:'权限',

prop:'grade',

type:'select',

dicData:[

{

label:'有权限',

value:1

},{

label:'无权限',

value:0

},{

label:'禁止项',

disabled:true,

value:-1

}

]

},{

label:'级别',

prop:'cascader',

type:'cascader',

dicData:[{

label:'一级',

value:0,

children:[

{

label:'一级1',

value:1,

},{

label:'一级2',

value:2,

}

]

}],

},

{

width:120,

label:'多选',

prop:'checkbox',

type:'checkbox',

span:24,

dicData:[{

label:'有权限',

value:1

},{

label:'无权限',

value:0

},{

label:'禁止项',

disabled:true,

value:-1

}]

},

{

label:'树型',

prop:'tree',

type:'tree',

dicData:[{

label:'一级',

value:0,

children:[

{

label:'一级1',

value:1,

},{

label:'一级2',

value:2,

}

]

}]

}

]

}

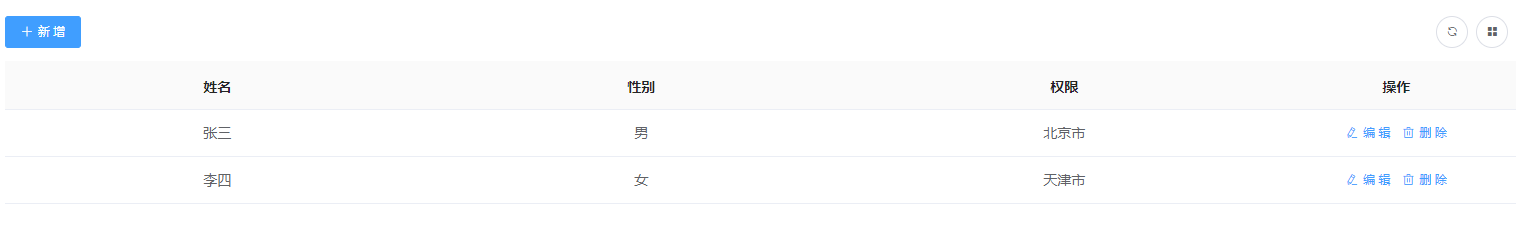
}

}

}

</**script**>

后台接口字典：



如果你的后台字典不是本地字典，支持给每一列属性单独配置网络字典

<**uvue-crud** :data="data2" :option="option3" ></**uvue-crud**>

<**script**>

**export** **default** {

data() {

**return** {

data2: [

{

name:'张三',

sex:'男',

grade:'110000'

}, {

name:'李四',

sex:'女',

grade:'120000'

}

],

option3:{

page:false,

align:'center',

menuAlign:'center',

column:[

{

label:'姓名',

prop:'name',

}, {

label:'性别',

prop:'sex'

},{

label:'权限',

prop:'grade',

type:'select',

props: {

label: 'name',

value: 'code'

},

dicUrl:'https://xxxx/api/area/getProvince'

}

]

}

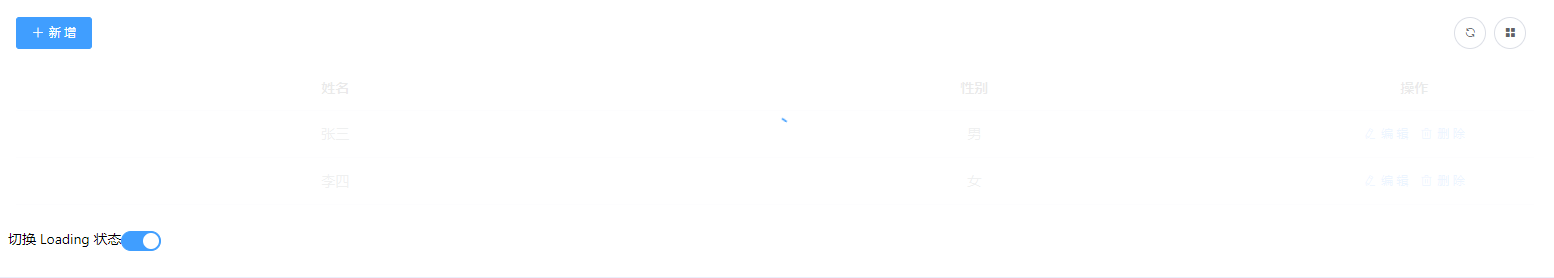
}

}

}

</**script**>

## 等待



组件中loading接受一个Boolean的属性，来控制组件的等待框显隐藏

<**uvue-crud** :data="data" :option="option" :table-loading="loading"></**uvue-crud**>

<**div** style="margin-top:20px;">

切换 Loading 状态<**el-switch** size="small" v-model="loading"> </**el-switch**>

</**div**>

<**script**>

**export** **default** {

data() {

**return** {

loading:true,

data: [

{

name:'张三',

sex:'男'

}, {

name:'李四',

sex:'女'

}

],

option:{

page:false,

align:'center',

menuAlign:'center',

column:[

{

label:'姓名',

prop:'name'

}, {

label:'性别',

prop:'sex'

}

]

},

};

},

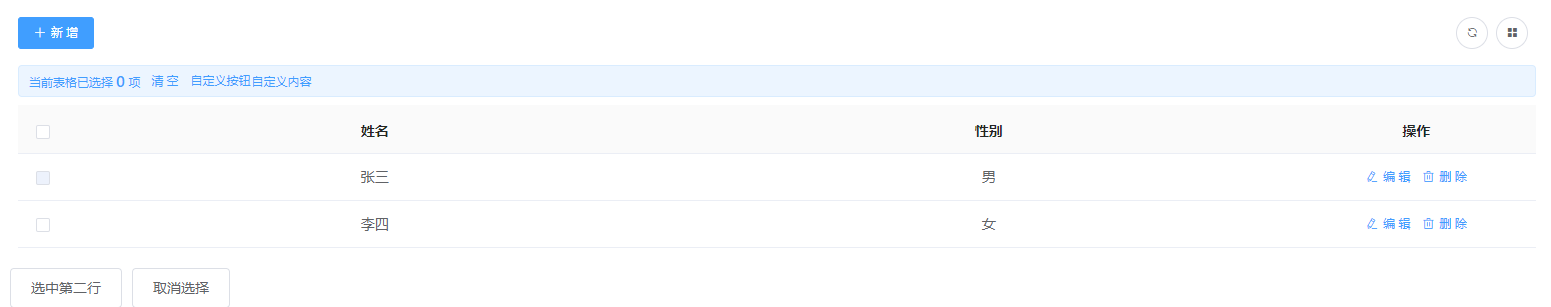
created(){

}

}

</**script**>

## 多选



设selection属性为true即可；勾选的同时会回调selectionChange方法返回当前选中的数据,setCurrent方法设置选中的行,selectable函数决定该行是否可以勾选

<**uvue-crud** ref="crud" :data="data" :option="option4" @selection-change="selectionChange">

<**template** slot="tip">

<**el-button** type="text" size="small">

自定义按钮

</**el-button**>

<**span**>自定义内容</**span**>

</**template**>

</**uvue-crud**>

<**div** style="margin-top: 20px">

<**el-button** @click="toggleSelection([data[1]])">选中第二行</**el-button**>

<**el-button** @click="toggleSelection()">取消选择</**el-button**>

</**div**>

<**script**>

**export** **default** {

data() {

**return** {

data: [

{

name:'张三',

sex:'男'

}, {

name:'李四',

sex:'女'

}

],

option4:{

selection: true,

selectable:(row,index)=>{

**return** index===1;

},

page:false,

align:'center',

menuAlign:'center',

column:[

{

label:'姓名',

prop:'name'

}, {

label:'性别',

prop:'sex'

}

]

},

};

},

methods: {

selectionChange(list){

**this**.$message.success('选中的数据'+ JSON.stringify(list));

},

toggleSelection(val){

**this**.$refs.crud.toggleSelection(val);

}

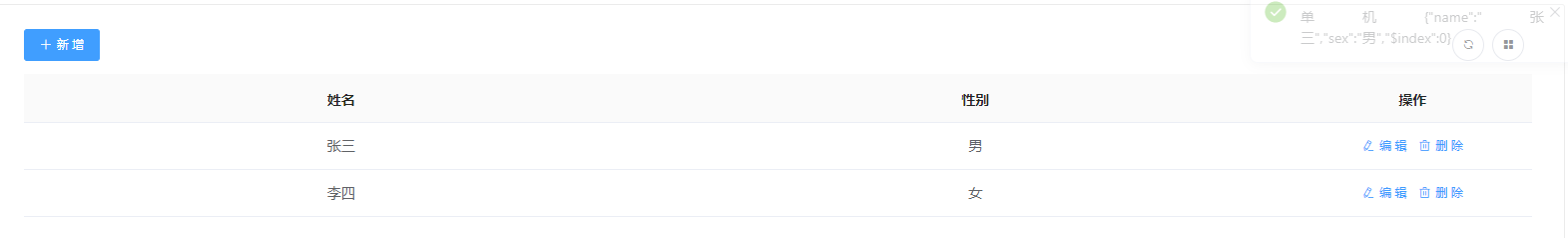
}

}

</**script**>

## 行操作

行单击：



单机一行数据时回调row-click方法,同时返回当前行的row数据,event当前的操作对象,column当前列的属性

<**uvue-crud** :data="data" :option="option" @row-click="handleRowClick"></**uvue-crud**>

<**script**>

**export** **default** {

data() {

**return** {

data: [

{

name:'张三',

sex:'男'

}, {

name:'李四',

sex:'女'

}

],

option:{

page:false,

align:'center',

menuAlign:'center',

column:[

{

label:'姓名',

prop:'name'

}, {

label:'性别',

prop:'sex'

}

]

}

};

},

methods: {

handleRowClick (row, event, column) {

**this**.$notify({

showClose: true,

message: '单机'+JSON.stringify(row),

type: 'success',

});

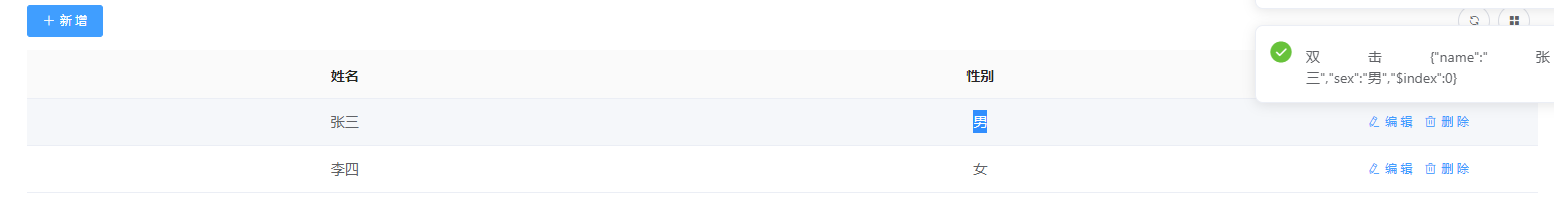
},

}

}

</**script**>

行双击：



双击一行数据时回调row-dblclick方法,同时返回当前行的row数据,event当前的操作对象,column当前列的属性

<**uvue-crud** :data="data" :option="option" @row-dblclick="handleRowDBLClick"></**uvue-crud**>

<**script**>

**export** **default** {

data() {

**return** {

data: [

{

name:'张三',

sex:'男'

}, {

name:'李四',

sex:'女'

}

],

option:{

page:false,

align:'center',

menuAlign:'center',

column:[

{

label:'姓名',

prop:'name'

}, {

label:'性别',

prop:'sex'

}

]

}

}

},

methods: {

handleRowDBLClick (row, event) {

**this**.$notify({

showClose: true,

message: '双击'+JSON.stringify(row),

type: 'success',

});

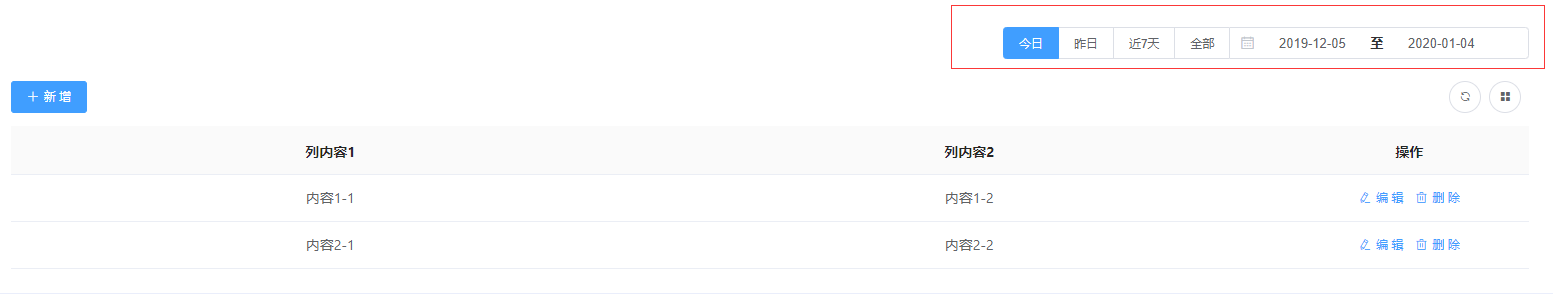
},

}

}

</**script**>

## 日期组件



配置dateDefault为false可以取消默认选择

<**uvue-crud** ref="crud" @date-change="dateChange":option="option" :data="data" @sortable-change="sortableChange"></**uvue-crud**>

<**script**>

**export** **default** {

data(){

**return** {

data:[{

text1:'内容1-1',

text2:'内容1-2'

},{

text1:'内容2-1',

text2:'内容2-2'

},{

text1:'内容3-1',

text2:'内容3-2'

}],

option:{

dateBtn:true,

align:'center',

column: [{

label: '列内容1',

prop: 'text1',

}, {

label: '列内容2',

prop: 'text2',

}]

}

}

},

methods:{

dateChange(date){

**this**.$message.success(date);

}

}

}

</**script**>

## 自定义表头



调用renderHeader函数即可自定义表头，具体操作看如下例子

<**uvue-crud** ref="crud" :option="option" :data="data"></**uvue-crud**>

<**script**>

**export** **default** {

data() {

**return** {

option: {

column: [{

label: '姓名',

prop: 'name',

renderHeader: (h, { column, $index }) => {

**return** h('div',[

h('el-tag', column.label),

h('spam', '自定义表头'),

])

},

}, {

label: '年龄',

prop: 'sex'

}]

},

data: []

}

}

}

</**script**>

## 复杂表头



<**uvue-crud** :option="option" :data="data" ></**uvue-crud**>

<**script**>

**export** **default** {

data() {

**return** {

option: {

border: true,

index: true,

expandLevel: 3,

headerAlign: 'center',

align: 'center',

tree: true,

labelWidth: 100,

column: [{

label: '姓名',

prop: 'name',

width:140,

}, {

label: '性别1',

prop: 'sex',

},

{

label: '自定义图标',

prop: 'icon',

type: "icon-select",

iconList: [{

label: '基本图表',

list: ['el-icon-time', 'el-icon-bell', 'el-icon-star-on']

}]

}, {

label: '复杂表头',

children: [{

label: '信息',

children: [{

label: '年龄',

prop: 'age'

}, {

label: '手机号',

prop: 'phone',

}]

}, {

label: '级别',

prop: 'grade',

type: 'select',

dicData: [{

label: '测试',

value: 1

}]

}]

}, {

label: '测试',

prop: 'test',

},

{

label: '手机信息1',

prop: 'phone1'

}],

},

data: [{

name: '张三',

age: 14,

grade: 1,

phone1: '17547400800',

phone: '17547400800',

icon: 'el-icon-time',

test: 1,

sex: '男'

}, {

name: '王五',

age: 10,

grade: 1,

test: 1,

sex: '女',

icon: 'el-icon-star-on',

phone1: '17547400800',

phone: '17547400800'

}]

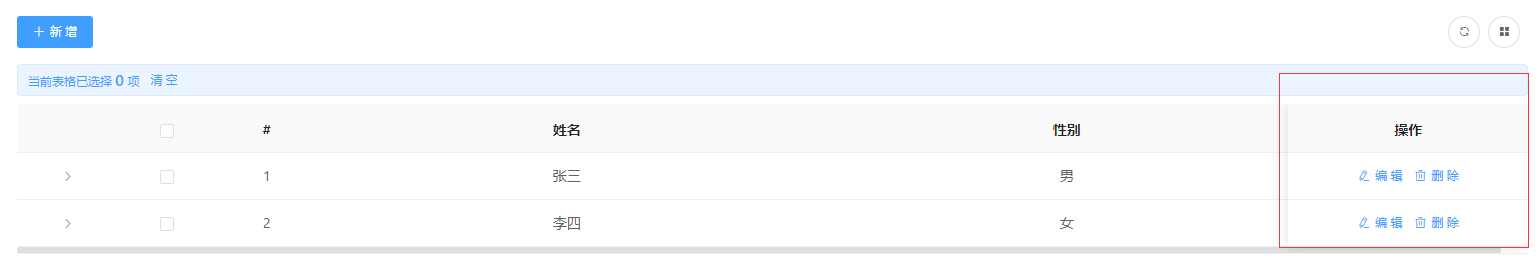
}

}

}

</**script**>

## 冻结列



配置indexFixed,selectionFixed,expandFixed可以配置序号，多选，面板是否为冻结,当然你也可以配置他们的宽度indexWdth,selectionWidth,expandWidth。

<**uvue-crud** :data="data" :option="option" ></**uvue-crud**>

<**script**>

**export** **default** {

data() {

**return** {

data: [

{

id:1,

name:'张三',

sex:'男'

}, {

id:2,

name:'李四',

sex:'女'

}

],

option:{

index:true,

indexFixed:false,

indexWidth:100,

selection:true,

selectionWidth:100,

selectionFixed:false,

expand:true,

expandWidth:100,

expandFixed:false,

align:'center',

menuAlign:'center',

column:[

{

width:500,

label:'姓名',

prop:'name'

}, {

width:500,

label:'性别',

prop:'sex'

}

]

},

};

}

}

</**script**>

## 行/单元格样式



对开开放了cell-style和row-style方法

<**uvue-crud** :data="data" :option="option" :cell-style="cellStyle" :row-style="rowStyle"></**uvue-crud**>

<**script**>

**export** **default** {

data() {

**return** {

data: [

{

name:'张三',

money:3000

}, {

name:'李四',

sex:false,

money:4000

}, {

name:'王五',

sex:false,

money:2000

}

],

option:{

column:[

{

label:'姓名',

prop:'name'

},{

label:'工资',

prop:'money'

}

]

}

}

},

methods:{

rowStyle({row,column,rowIndex}){

**if**(rowIndex%2===0){

**return** {

backgroundColor:'#eee',

color:'#fff'

}

}

},

cellStyle({row,column,rowIndex,columnIndex}){

**if**(columnIndex==1){

**if**(row.money<=3000){

**return** {

color:'green',

fontWeight:'bold',

fontSize:'20'

}

}**else**{

**return** {

color:'red',

fontWeight:'bold',

fontSize:'20'

}

}

}

}

}

}

</**script**>

## 空状态



emptyText属性可以配置空状态时的提示语

<**uvue-crud** ref="crud" :option="option" :data="data"></**uvue-crud**>

<**script**>

**export** **default** {

data() {

**return** {

option: {

emptyText: '自定义暂无数据提示语',

column: [{

label: '姓名',

prop: 'name'

}, {

label: '年龄',

prop: 'sex'

}]

},

data: []

}

}

}

</**script**>

## 导出



excelBtn设置为true即可开启导出功能，如果配置了selection属性，则需要勾选导出的数据，否则为全部,如果配置了title属性，则为导出文件的名称

<**uvue-crud** :option="option" :data="data"></**uvue-crud**>

<**script**>

**export** **default** {

data(){

**return** {

data:[{

text1:'内容1-1',

text2:'内容1-2'

},{

text1:'内容2-1',

text2:'内容2-2'

}],

option:{

title:'表格的标题',

excelBtn:true,

addBtn:false,

menu:false,

align:'center',

column: [{

label: '列内容1',

prop: 'text1',

}, {

label: '列内容2',

prop: 'text2',

}]

}

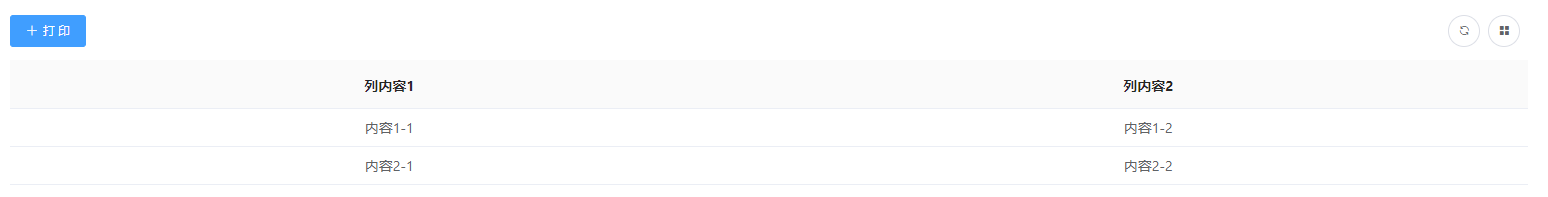
}

}

}

</**script**>

## 打印



printBtn设置为true即可开启打印功能，如果配置了selection属性，则需要勾选打印的数据，否则为全部

<**uvue-crud** :option="option" :data="data"></**uvue-crud**>

<**script**>

**export** **default** {

data(){

**return** {

data:[{

text1:'内容1-1',

text2:'内容1-2'

},{

text1:'内容2-1',

text2:'内容2-2'

}],

option:{

align:'center',

printBtn:true,

addBtn:false,

menu:false,

column: [{

label: '列内容1',

prop: 'text1',

}, {

label: '列内容2',

prop: 'text2',

}]

}

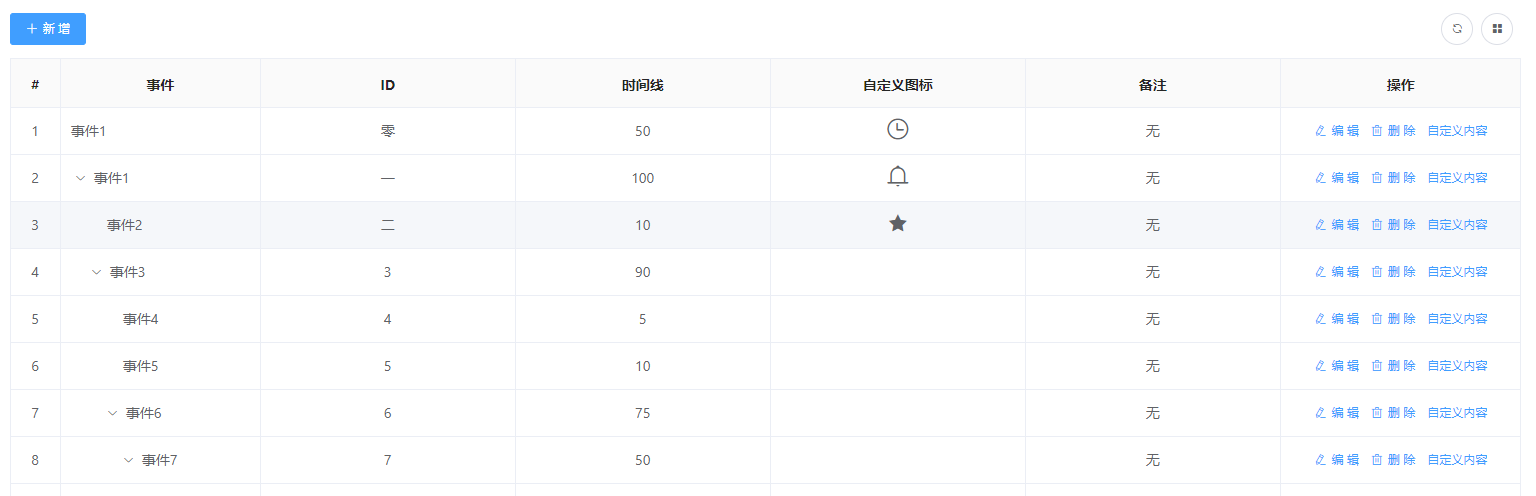
}

}

}

</**script**>

## 表格树



defaultExpandAll属性是否全部展开,其他用法和普通crud一致,一定要配置rowKey主键Id

<**uvue-crud** :option="option" :data="data" >

<**template** slot="icon" slot-scope="scope">

<**i** :class="scope.row.icon" style="font-size:24px"></**i**>

</**template**>

<**template** slot="menu" slot-scope="scope">

<**el-button** :size="scope.size" :type="scope.type">自定义内容</**el-button**>

</**template**>

</**uvue-crud**>

<**script**>

**export** **default** {

data(){

**return** {

data: [

{

id: 0,

event: '事件1',

timeLine: 50,

icon: 'el-icon-time',

comment: '无'

},

{

id: 1,

event: '事件1',

icon: 'el-icon-bell',

timeLine: 100,

comment: '无',

children: [

{

id: 2,

event: '事件2',

icon: 'el-icon-star-on',

timeLine: 10,

comment: '无'

},

{

id: 3,

event: '事件3',

timeLine: 90,

comment: '无',

children: [

{

id: 4,

event: '事件4',

timeLine: 5,

comment: '无'

},

{

id: 5,

event: '事件5',

timeLine: 10,

comment: '无'

},

{

id: 6,

event: '事件6',

timeLine: 75,

comment: '无',

children: [

{

id: 7,

event: '事件7',

timeLine: 50,

comment: '无',

children: [

{

id: 71,

event: '事件71',

timeLine: 25,

comment: 'xx'

},

{

id: 72,

event: '事件72',

timeLine: 5,

comment: 'xx'

},

{

id: 73,

event: '事件73',

timeLine: 20,

comment: 'xx'

}

]

},

{

id: 8,

event: '事件8',

timeLine: 25,

comment: '无'

}

]

}

]

}

]

}

],

option: {

headerAlign: 'center',

align: 'center',

border: true,

index: true,

defaultExpandAll:true,

column: [

{

label: '事件',

prop: 'event',

align: 'left',

width: 200

},

{

label: 'ID',

prop: 'id',

dicData: [{

label: '零',

value: 0

}, {

label: '一',

value: 1

}, {

label: '二',

value: 2,

}]

},

{

label: '时间线',

prop: 'timeLine'

},

{

label: '自定义图标',

prop: 'icon',

slot: true,

},

{

label: '备注',

prop: 'comment'

}

],

}

}

}

}

</**script**>

## 左树右表



这是tree组件和crud组件的结合使用，主要是用nodeClick回调刷新crud数据

<**el-container**>

<**el-aside** width="200px">

<**uvue-tree** :option="treeOption" :data="treeData" @node-click="nodeClick"></**uvue-tree**>

</**el-aside**>

<**el-main**>

<**uvue-crud**

:data="data"

:option="option"

v-model="obj"

></**uvue-crud**>

</**el-main**>

</**el-container**>

<**script**>

**export** **default** {

data() {

**return** {

obj: {},

treeData:[

{

id:0,

label:'一级部门',

value:0,

children:[

{

id:1,

label:'一级部门1',

value:0,

}

]

}],

treeOption:{

nodeKey:'id',

addBtn:false,

menu:false,

size:'small',

formOption:{

labelWidth:100,

column:[{

label:'自定义项',

prop:'test'

}],

},

props:{

labelText:'标题',

label:'label',

value:'value',

children:'children'

}

},

data:[],

data0: [

{

name: '张三',

sex: '男'

},

{

name: '李四',

sex: '女'

}

],

data1: [

{

name: '张三1',

sex: '男'

},

{

name: '李四2',

sex: '女'

}

],

option: {

column: [

{

label: '姓名',

prop: 'name'

},

{

label: '性别',

prop: 'sex'

}

]

}

}

},

created(){

**this**.data=**this**.data0;

},

methods: {

nodeClick(data){

**if**(data.id==0){

**this**.data=**this**.data0;

}**else** **if**(data.id==1){

**this**.data=**this**.data1;

}

**this**.$message.success(JSON.stringify(data))

}

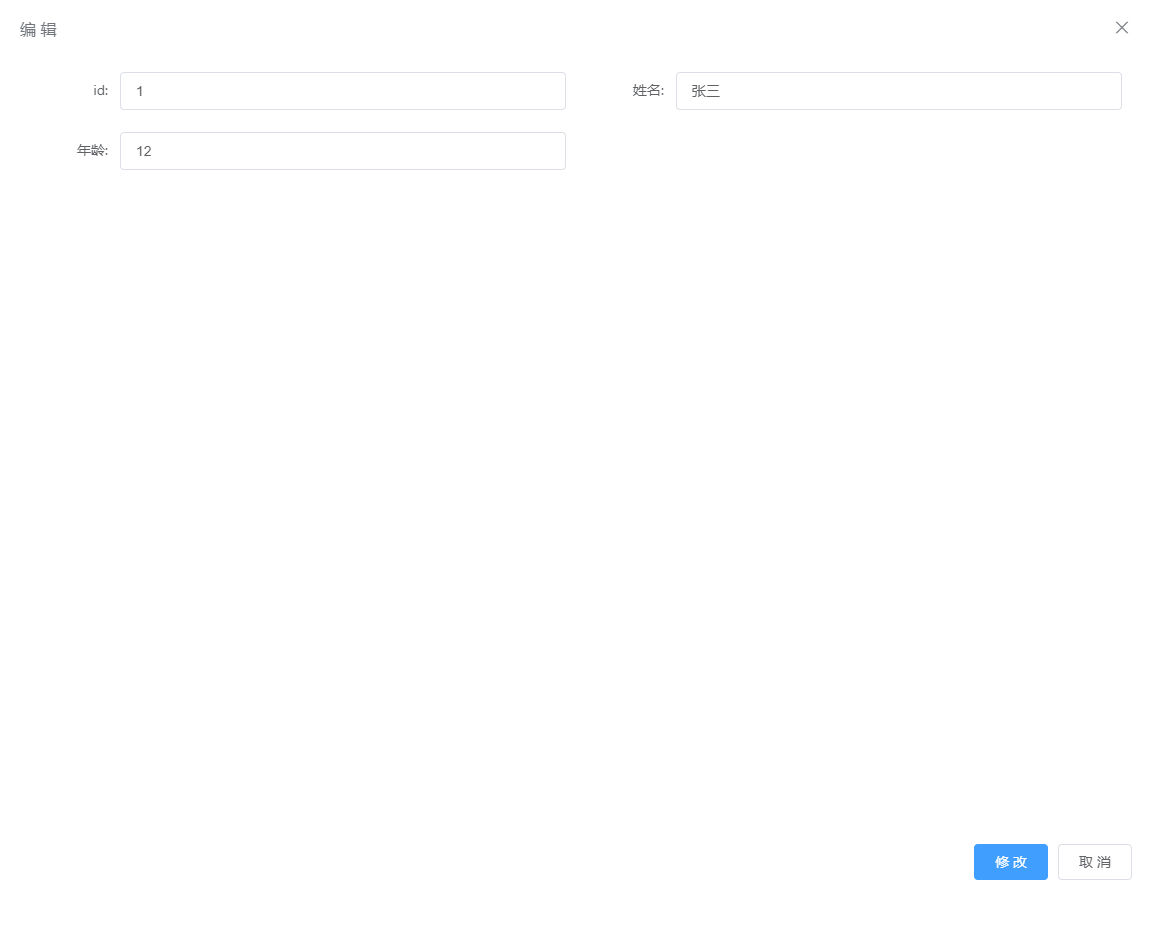
}

}

</**script**>

## 抽屉弹框





配置dialogType为弹窗的方式,dialogDirection为弹窗的位置

<**el-radio-group** v-model="direction">

<**el-radio** label="ltr">从左往右开</**el-radio**>

<**el-radio** label="rtl">从右往左开</**el-radio**>

<**el-radio** label="ttb">从上往下开</**el-radio**>

<**el-radio** label="btt">从下往上开</**el-radio**>

</**el-radio-group**>

<**uvue-crud** :option="option" :data="list"></**uvue-crud**>

<**script**>

**export** **default** {

watch:{

direction(value){

**this**.option.dialogDirection=value;

}

},

data(){

**return** {

direction:'rtl',

list:[{

id:1,

name:'张三',

sex:12

},{

id:2,

name:'李四',

sex:12

}],

option:{

dialogDirection:'rtl',

dialogType:'drawer',

column:[{

label:'id',

prop:'id'

},{

label:'姓名',

prop:'name'

},{

label:'年龄',

prop:'sex'

}]

}

}

}

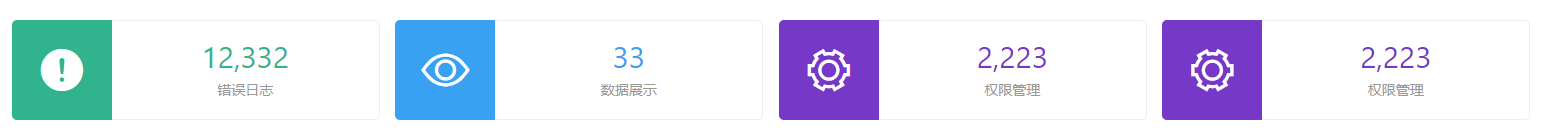
}

}

</**script**>

# 数据展示

## DataBox



<**uvue-data-box** :option="option"></**uvue-data-box**>

<**script**>

**export** **default** {

data(){

**return** {

option: {

span:6,

data: [

{

click: **function** (item) {

alert(JSON.stringify(item));

},

title: '错误日志',

count: 12332,

icon: 'el-icon-warning',

color: 'rgb(49, 180, 141)',

href:'https://uway.cn',

target:'\_blank'

},

{

click: **function** (item) {

alert(JSON.stringify(item));

},

title: '数据展示',

count: 33,

icon: 'el-icon-view',

color: 'rgb(56, 161, 242)',

href:'https://uway.cn',

target:'\_blank'

},

{

click: **function** (item) {

alert(JSON.stringify(item));

},

title: '权限管理',

count: 2223,

icon: 'el-icon-setting',

color: 'rgb(117, 56, 199)',

href:'https://uway.cn',

target:'\_blank'

},

{

click: **function** (item) {

alert(JSON.stringify(item));

},

title: '权限管理',

count: 2223,

icon: 'el-icon-setting',

color: 'rgb(117, 56, 199)',

href:'https://uway.cn',

target:'\_blank'

},

]

},

}

}

}

</**script**>

## DataDisplay



<**uvue-data-display** :option="option"></**uvue-data-display**>

<**script**>

**export** **default** {

data(){

**return** {

option: {

span:6,

data: [

{

click: **function** (item) {

alert(JSON.stringify(item));

},

count: 100,

title: '日活跃数',

href:'https://uway.cn',

target:'\_blank'

},

{

click: **function** (item) {

alert(JSON.stringify(item));

},

count: '3,000',

title: '月活跃数',

href:'https://uway.cn',

target:'\_blank'

},

{

click: **function** (item) {

alert(JSON.stringify(item));

},

count: '20,000',

title: '年活跃数',

href:'https://uway.cn',

target:'\_blank'

},

{

click: **function** (item) {

alert(JSON.stringify(item));

},

count: '40,000',

title: '周活跃数',

href:'https://uway.cn',

target:'\_blank'

}

]

},

}

}

}

</**script**>

## DataIcons



<**uvue-data-icons** :option="option"></**uvue-data-icons**>

<**script**>

**export** **default** {

data(){

**return** {

option: {

span:4,

data: [

{

click: **function** (item) {

alert(JSON.stringify(item));

},

title: '今日注册',

count: 12678,

icon: 'el-icon-tickets',

href:'https://uway.cn',

target:'\_blank'

},

{

click: **function** (item) {

alert(JSON.stringify(item));

},

title: '今日登录',

count: 22139,

icon: 'el-icon-success',

href:'https://uway.cn',

target:'\_blank'

},

{

click: **function** (item) {

alert(JSON.stringify(item));

},

title: '今日订阅',

count: 35623,

icon: 'el-icon-info',

href:'https://uway.cn',

target:'\_blank'

},

{

click: **function** (item) {

alert(JSON.stringify(item));

},

title: '今日评论',

count: 16826,

icon: 'el-icon-message',

href:'https://uway.cn',

target:'\_blank'

},

{

click: **function** (item) {

alert(JSON.stringify(item));

},

title: '今日评论',

count: 16826,

icon: 'el-icon-message',

href:'https://uway.cn',

target:'\_blank'

},

{

click: **function** (item) {

alert(JSON.stringify(item));

},

title: '今日评论',

count: 16826,

icon: 'el-icon-message',

href:'https://uway.cn',

target:'\_blank'

}

]

},

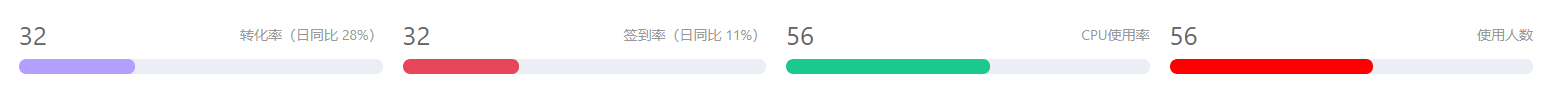
}

}

}

</**script**>

## DataProgress



<**uvue-data-progress** :option="option"></**uvue-data-progress**>

<**script**>

**export** **default** {

data(){

**return** {

option: {

span:6,

data: [

{

click: **function** (item) {

alert(JSON.stringify(item));

},

title: '转化率（日同比 28%）',

color: 'rgb(178, 159, 255)',

count: 32,

href:'https://uway.cn',

target:'\_blank'

},

{

click: **function** (item) {

alert(JSON.stringify(item));

},

title: '签到率（日同比 11%）',

color:'rgb(230, 71, 88)',

count: 32,

href:'https://uway.cn',

target:'\_blank'

},

{

click: **function** (item) {

alert(JSON.stringify(item));

},

title: 'CPU使用率',

color:'rgb(27, 201, 142)',

count: 56,

href:'https://uway.cn',

target:'\_blank'

},

{

click: **function** (item) {

alert(JSON.stringify(item));

},

title: '使用人数',

color:'red',

count: 56,

href:'https://uway.cn',

target:'\_blank'

}

]

},

}

}

}

</**script**>

## DataRotate



<**uvue-data-rotate** :option="option"></**uvue-data-rotate**>

<**script**>

**export** **default** {

data(){

**return** {

option: {

span: 8,

data: [

{

click: **function** (item) {

alert(JSON.stringify(item));

},

count: '150',

title: '新订单',

icon: 'el-icon-warning',

color: 'rgb(49, 180, 141)'

}, {

click: **function** (item) {

alert(JSON.stringify(item));

},

count: '53%',

title: '跳出率',

icon: 'el-icon-view',

color: '#00a65a'

}, {

click: **function** (item) {

alert(JSON.stringify(item));

},

count: '44',

title: '用户注册数',

icon: 'el-icon-setting',

color: '#f39c12'

}

]

},

}

}

}

</**script**>

## DataTabs



<**uvue-data-tabs** :option="option"></**uvue-data-tabs**>

<**script**>

**export** **default** {

data(){

**return** {

option: {

data: [

{

click: **function** (item) {

alert(JSON.stringify(item));

},

title: '分类统计',

subtitle: '实时',

count: 7993,

allcount: 10222,

text: '当前分类总记录数',

color: 'rgb(27, 201, 142)',

key: '类'

},

{

click: **function** (item) {

alert(JSON.stringify(item));

},

title: '附件统计',

subtitle: '实时',

count: 3112,

allcount: 10222,

text: '当前上传的附件数',

color: 'rgb(230, 71, 88)',

key: '附'

},

{

click: **function** (item) {

alert(JSON.stringify(item));

},

title: '文章统计',

subtitle: '实时',

count: 908,

allcount: 10222,

text: '评论次数',

color: 'rgb(178, 159, 255)',

key: '评'

}

]

}

}

}

}

</**script**>

## DataPanel



<**uvue-data-panel** :option="option"></**uvue-data-panel**>

<**script**>

**export** **default** {

data(){

**return** {

option: {

span:8,

data: [

{

click: **function** (item) {

alert(JSON.stringify(item));

},

title: 'New Visits',

count: '102,400',

icon: 'el-icon-message',

color: '#00a7d0',

},

{

click: **function** (item) {

alert(JSON.stringify(item));

},

title: 'Messages',

count: '81,212',

icon: 'el-icon-info',

color: 'rgb(27, 201, 142)',

},

{

click: **function** (item) {

alert(JSON.stringify(item));

},

title: 'Purchases',

count: '9,280',

icon: 'el-icon-success',

color: 'rgb(230, 71, 88)',

}

]

}

}

}

}

</**script**>