



列表自定义列实现

功能说明：

页面自定义设置列表需要选择的列，设置组件集成的两种方法，一个是在列表外增加设置组件，一个是在列表表头增加设置组件
具体代码案例参照【常用示例-单表模型示例】 功能

功能预览：

名称：

年龄：

查询

重置

高级查询

展开

+ 新增

+ 创建单据

+ 一对多

导出

导入

已选择 0 项

清空

刷新

自定义列

<input type="checkbox"/>	#	姓名	关键词	打卡时间	性别	年龄	生日	邮箱	个人简介	操作
<input type="checkbox"/>	1	导入小虎			女	28				编辑 更多
<input type="checkbox"/>	2	导入zhangdaiscott			男		2019-01-03			编辑 更多
<input type="checkbox"/>	3	小麦			女			zhangdaiscott@163.com		编辑 更多
<input type="checkbox"/>	4	郭靖11a	777	2018-12-07 00:00:00						编辑 更多
<input type="checkbox"/>	5	杨康22a	奸臣							编辑 更多
<input type="checkbox"/>	6	报名时间								编辑 更多

名称：

年龄：

查询

重置

高级查询

展开

+ 新增

+ 创建单据

+ 一对多

导出

导入

已选择 0 项

清空

刷新

自定义列

自定义列

通过勾选来控制展示的列

☒ 姓名

☒ 关键词

☒ 打卡时间

☒ 性别

☒ 年龄

☒ 生日

☒ 邮箱

☒ 个人简介

自定义列

名称：

年龄：

查询

重置

高级查询

展开

+ 新增

+ 创建单据

+ 一对多

导出

导入

已选择 0 项

清空

刷新

自定义列

☒ 姓名

☒ 关键词

☒ 打卡时间

☒ 性别

☒ 年龄

☒ 生日

☒ 邮箱

☒ 个人简介

通过勾选来控制展示的列

自定义列

实现方法：

一. 增加初始化配置

1. data() 方法中配置

```
//表头
columns:[],
//列设置
settingColumns:[],
//列定义
defColumns: [{
  title: '#',
  dataIndex: '',
  key: 'rowIndex',
```

```

        width: 60,
        align: "center",
        customRender: function (t, r, index) {
            return parseInt(index) + 1;
        }
    },
    {
        title: '姓名',
        align: "center",
        dataIndex: 'name'
    },
    .....
    .....
]

```

说明:

columns: 列表展示的列, 初始为空。

settingColumns: 保存勾选的列设置

defColumns: 定义列表可以展示的列信息

2. 增加设置按钮, 两种实现方式任选其一即可

(1) 第一种在列表外增加设置按钮

```

<a-popover title="自定义列" trigger="click" placement="leftBottom">
  <template slot="content">
    <a-checkbox-group @change="onColSettingsChange" v-model="settingColumns" :defaultValue="
      <a-row>
        <template v-for="(item,index) in defColumns">
          <template v-if="item.key!='rowIndex'&& item.dataIndex!='action'">
            <a-col :span="12"><a-checkbox :value="item.dataIndex">{{ item.title }}</a-chec
          </template>
        </template>
      </a-row>
    </a-checkbox-group>
  </template>
  <a><a-icon type="setting" />自定义列</a>
</a-popover>

```

(2) 第二种在表头列中扩展按钮

在操作列定义中增加插槽设置

```

{
    title: '操作',
    dataIndex: 'action',
    align: "center",
    scopedSlots: {
        filterDropdown: 'filterDropdown',
        filterIcon: 'filterIcon',
        customRender: 'action'},
}

```

中增加插槽代码

```

<div slot="filterDropdown">
  <a-card>
    <a-checkbox-group @change="onColSettingsChange" v-model="settingColumns" :defaultValue="se
      <a-row>
        <template v-for="(item,index) in defColumns">
          <template v-if="item.key!='rowIndex'&& item.dataIndex!='action'">
            <a-col :span="12"><a-checkbox :value="item.dataIndex">{{ item.title }}</a-checkbox
          </template>
        </template>
      </a-row>
    </a-checkbox-group>
  </a-card>

```

3. 实现checkbox @change

```
//列设置更改事件
onColSettingsChange (checkedValues) {
  var key = this.$route.name+":colsettings";
  Vue.ls.set(key, checkedValues, 7 * 24 * 60 * 60 * 1000)
  this.settingColumns = checkedValues;
  const cols = this.defColumns.filter(item => {
    if(item.key == 'rowIndex' || item.dataIndex == 'action'){
      return true
    }
    if (this.settingColumns.includes(item.dataIndex)) {
      return true
    }
    return false
  })
  this.columns = cols;
},
```

4. 页面加载时实现列的初始化方法

```
initColumns(){
  //权限过滤（列权限控制时打开，修改第二个参数为授权码前缀）
  //this.defColumns = colAuthFilter(this.defColumns,'testdemo:');

  var key = this.$route.name+":colsettings";
  let colSettings= Vue.ls.get(key);
  if(colSettings==null || colSettings==undefined){
    let allSettingColumns = [];
    this.defColumns.forEach(function (item,i,array ) {
      allSettingColumns.push(item.dataIndex);
    })
    this.settingColumns = allSettingColumns;
    this.columns = this.defColumns;
  }else{
    this.settingColumns = colSettings;
    const cols = this.defColumns.filter(item => {
      if(item.key == 'rowIndex' || item.dataIndex == 'action'){
        return true;
      }
      if (colSettings.includes(item.dataIndex)) {
        return true;
      }
      return false;
    })
    this.columns = cols;
  }
}
```

created中调用:

```
created() {
  this.initColumns();
},
```