

【组件】SvSearchTable-带高级查询列表

SvSearchTable表格

基于SvTable组件进行二次封装，用于展示多条结构类似的数据，可对数据进行排序或其他自定义操作。

示例说明

示例网址：[Savor-Enterprise](#)

账号：admin 密码：123456

源代码：src\components\SvSearchTable\index.vue

示例代码：src\views\admin\demo\SvSearchTableDemo.vue

使用说明

使用方式：

已经注册成全局组件SvTable，直接通过<sv-table ref="SvTable"><sv-table>的方式在template里面引用，通过fieldList配置表格中的属性。

表格属性：

参数	说明	类型	是否必填	默认值
moduleName	模块名称用于保存当前页的查询方案，可以与module为同一个	String	是	—
labelWidth	查询条件标签宽度	String	否	60px
extraDicts	自定义的数据字典	Object	否	—
stripe	是否为斑马纹 table，会和row-style、cell-style有样式冲突	Boolean	否	true
tableIndex	索引	Boolean	否	true
singleSelect	单选，对应的回调事件是@selection-change	Boolean	否	false
selection	多选，对应的回调事件是@selection-change	Boolean	否	false
checkedList	选中行数据	Array	否	[]
api	获取数据的接口	Function	是	—
data	列表数据	Array	是	—
fieldList	表格字段配置，详细参考表1-fieldList配置表	Array	是	—
pk	主键	String	否	id
pager	是否分页	Boolean	否	true
refresh	刷新标识	Number	否	—
initCurrentPage	重置当前页	Number	否	—
module	是否开启列设置保存功能，必须保证唯一	String	否	—
handle	操作栏配置，详细参考表2-handle配置表	Object	否	—
btnHideSet	操作列的按钮隐藏功能	Function	否	false
btnDisableSet	操作列的按钮禁用功能	Function	否	false
exportApi	导出接口，不用api可以直接自定义接口	Funtion	否	—
maxExportRows	导出最大行数	Funtion	否	—
exportFileName	导出文件名	String	否	—
customTableHeight	自定义table高度	String/Number	否	—
tabelWidth	表格宽度，默认100%	String	否	—
className	自定义类名	String	否	—
el-table的其他属性	支持el-table的其他属性	—	—	—

自定义的数据字典extraDicts示例

```
extraDicts: {
  isSuperAdmin: [
    {
      value: 1,
      labelDefault: ''
    },
    {
      value: 0,
      labelDefault: ''
    }
  ]
}
```

默认选中行checkedList示例

```
<!-- { orderId: '111', orderCode: 'DD202003180001', orderName: '' } -->
<sv-table pk="orderId" :check-list="['111']"></sv-table>
<!-- { id: '111', orderCode: 'DD202003180001', orderName: '' } -->
<sv-table :check-list="['111']"></sv-table>
```

表1-fieldList配置表

参数	说明	类型	可选值	默认值
inputType	查询条件输入框类型	String	参考表1-1inputType类型配置表	—
multiple	默认精确查询多个, 适用于inputType为input和select的类型	Boolean	true/false	false
tableAlias	字段对应的后台表别名	String	—	—
operation	连接符	String	参考表1-2operation类型配置表	—
initValue	初始值 (默认值)	String/Number/Array/Boolean	参考表1-3initValue类型配置表	—
inputHidden	在表上方的查询条件中不展示	Boolean	true/false	false
type	列展示内容类型	String	slot/tag/image/link	—
label	显示的列标题	String	—	—
prop	对应列或查询条件内容的字段名	String	—	—
align	对齐方式	String	left/right/center	center
width	对应列的宽度	String/Number	—	—
minWidth	对应列的最小宽度, 与 width 的区别是 width 是固定的, minWidth 会把剩余宽度按比例分配给设置了 minWidth 的列	String	—	100px
showOverflowTooltip	当内容过长被隐藏时显示 tooltip	Boolean	—	false
fixed	列是否固定在左侧或者右侧, true 表示固定在左侧	String/Boolean	true, left, right	—
hidden	列是否隐藏	Boolean	—	—
dictType	列展示的内容如果是字典值, 在此字段指定对应的字典类型	String	—	—
tagOptions	type为tag时可用。不设置tagOptions时, tag展示的是字段值; 设置tagOptions时, 可以进行tag展示值的映射, 详见示例	Object	—	—
path	类型为link时必须填, 设置的是跳转的路由	String	—	—
params	类型为link时使用, 路由参数, 数组元素为行数据的属性名, 参数值取自行数据	Array	—	—
customParams	类型为link时使用, 路由参数, 由父组件设置参数值	Object	—	—
dynamicRouteParams	类型为link时使用, 路由的动态部分, 数组元素为行数据的属性名或者固定值, 数据元素顺序同动态路由顺序一致	Array	—	—
children	嵌套表格	Array	—	—

表1-1 inputType类型配置表

序号	参数	说明	operation默认值
1	input	输入框类型	默认为eq, multiple为true时默认为in
2	number	计数器类型	默认为eq
3	select	选择器类型	默认为eq, multiple为true时默认为in
4	date	日期类型	默认为eq
5	datetime	时间日期类型	默认为eq
6	daterange	日期范围类型	默认为between

表1-2operation类型配置表

序号	参数	说明	语义
1	eq	等于	=
2	ne	不等于	<>
3	gt	大于	>
4	ge	大于等于	>=
5	lt	小于	<
6	le	小于等于	<=
7	like	模糊查询	'%value%'
8	likeLeft	左模糊查询	'%value'
9	likeRight	右模糊查询	'value%'
10	in	精确查询多个，对应valueData示例"a, b, c"	in
11	between	围查询，对应valueData示例"2020-08-18 21:59:01, 2020-08-18 22:02:59"	between s and e

表1-3initValue类型配置表

序号	inputType	multiple operation initValue类型 示例			
1	input	false	in	Array	['aaa', 'bbb']
2	input	true	其他	String	'aaa'
3	number	—	—	Number	1
4	select	false	in	Array	['woman', 'man']
5	select	true	其他	String /Number	'woman'
6	date	—	—	Date	this.moment(new Date()).format('YYYY-MM-DD')
7	datetime	—	—	Date	this.moment(new Date()).format('YYYY-MM-DD')
8	dateRange	—	—	Array	[this.moment(new Date()).format('YYYY-MM-DD HH:mm:ss'), this.moment(new Date()).format('YYYY-MM-DD HH:mm:ss')]
9	datetimeRange	—	—	Array	[this.moment(new Date()).format('YYYY-MM-DD HH:mm:ss'), this.moment(new Date()).format('YYYY-MM-DD HH:mm:ss')]

type: slot类型示例

```
<template>
  <sv-table ref="SvTable" :data.sync="tableInfo.data" :fieldList="tableInfo.fieldList">
    <!-- fieldListtypeslotcolprop -->
    <template v-slot:col-name="{ row }">
      <a style="color: #409eff" @click="handleView(row)">{{ row.name }}</a>
    </template>
  </sv-table>
</template>
```

```

<script>
import { mapGetters } from 'vuex'
import { pageQuery } from '@/views/admin/demo/api'
export default {
  name: 'SvTableDemo',
  data() {
    return {
      pageQuery,
      tableInfo: {
        refresh: 1,
        data: [],
        fieldList: [
          //
          {
            type: 'slot', // linkchildrentagimagefile
            label: '',
            prop: 'name',
            width: 120, // minWidth
            showOverflowTooltip: true // ...
          }
        ]
      }
    }
  }
}
</script>

```

type: tagOptions示例:

```

tagOptions: {
  0: {
    type: 'primary',
    name: ''
  },
  1: {
    type: 'warning',
    name: ''
  }
}

```

0、1为字段值，type是el-tag的type，name是tag展示值。

type: file类型示例:

```

{
  type: 'file',
  label: '',
  prop: 'attachment',
  storageType: 'minio' // minio
}

```

展示文件列表，可展示多个文件。

type: link类型示例:

```

{
  type: 'link',
  prop: 'code',
  label: '',
  path: '/baseManager/userForm',
  params: ['code'],
  customParams: {
    formStatus: 'view'
  },
  dynamicRouteParams: ['id', '2']
}

```

如果行数据是 {id: '1', code: 'DD202003180001', description: '采购订单'}, 最终生成的url如下:

<http://localhost:9527/baseManager/userForm/1/2?code=DD202003180001&formStatus=view>

- path是路由，在router.js中定义
- '/1/2'部分由dynamicRouteParams生成，'/baseManager/userForm/1/2'是动态路由的应用
- 'code=DD202003180001'部分由params生成，是url携带的参数，参数值来源于行数据，通过this.\$route.query获取
- 'formStatus=view'部分由customParams生成，也是url携带的参数，参数值自定义，通过this.\$route.query获取

```
//
{
  type: 'link',
  prop: 'code',
  label: '',
  params: ['code'],
  tagInfo: {
    path: 'org/struInnerUser/components/InnerUserForm',
    title: '-{1}',
    status: 'view',
    pkName: 'bizUserId'
  },
  titleParams: ['userName']
}
```

最终子组件可接收到的prop有bizUserId、status、code

列中带children示例

```
//
{
  prop: 'description',
  label: '',
  minWidth: 100,
  children: [
    {
      prop: 'description1',
      label: '1'
    },
    {
      prop: 'description2',
      label: '2'
    }
  ]
}
```

表2-handle配置表

参数	说明	类型	可选值	默认值
fixed	操作列是否固定在左侧或者右侧，true 表示固定在左侧	String/Boolean	true, left, right	—
align	对齐方式	String	left/right/center	center
label	显示的操作列标题	String	—	操作
width	对应操作列的宽度	String/Number	—	—
min-width	对应操作列的最小宽度	String/Number	—	—
btList	按钮配置，表3-btList配置表	Array	—	—

表3-btList配置表

参数	说明	类型	可选值	默认值
slot	插槽，自定义内容	template	—	—
event	事件名称	String	—	—
show	是否显示按钮	Boolean	—	—
type	按钮类型	String	primary / success / warning / danger / info / text	—
icon	图标	String	—	—
disabled	是否禁用按钮	Boolean	—	—

动态禁用或隐藏按钮示例

```
<template>
  <div class="app-container">
    <!-- btnHideSetbtnDisableSet -->
    <sv-table ref="SvTable" :btnHideSet="btnHideSet" :btnDisableSet="btnDisableSet"> </sv-table>
  </div>
</template>
```

```
<script>
export default {
  methods: {
    //
    btnHideSet(event, row) {
      if (event === 'relateGroup' && row.id === '1') return true
    },
    //
    btnDisableSet(event, row) {
      if (event === 'relateDepart' && row.id === '1') return true
    }
  }
}
</script>
```

按钮slot插槽自定义示例

```
<template>
  <sv-table ref="SvTable" :btnHideSet="btnHideSet" :btnDisableSet="btnDisableSet">
    <!-- btevent -->
    <template v-slot:bt-removeGroupRelate="{ data }">
      <el-button type="text" :disabled="data.item.event === 'removeGroupRelate' && data.row.id === '1'">

    </el-button>
    </template>
    <!-- btevent -->
    <template v-slot:bt-removeDepartRelate="{ data }">
      <el-button type="text" :disabled="data.item.event === 'removeDepartRelate' && data.row.id === '1'">

    </el-button>
    </template>
  </sv-table>
</template>

<script>
import { pageQuery } from '@views/admin/demo/api'
export default {
  name: 'SvTableDemo',
  data() {
    return {
      pageQuery,
      tableInfo: {
        refresh: 1,
        data: [],
        fieldList: [],
        //
        handle: {
          //
          width: 340, // minWidth
          // align: center, // center
          // fixed: 'right', //
          btList: [
            {
              type: 'text',
              label: '',
              event: 'relateGroup', //
              show: true //
            }
          ]
        }
      }
    }
  }
}
</script>
```

表格事件

事件名	说明	参数
selection-change	和singleSelect或者selection配合使用，选择行数据	rows/row (Array/Object)
row-dbclick	双击选中行事件	row
handle-click	操作栏按钮点击事件	event, row, index

表格方法

方法名	说明	参数
exportExcel	excel导出列表所有内容，根据列名生成表头，导出内容同列表内容（列表所有内容），默认最大导出10000行	—

exportCurrentPage	excel导出列表当前页，根据列名生成表头，导出内容同列表内容（导出列表当前页）	—
columnsSettings	打开列设置	—
resetColumnSettings	重置列设置（初始化列设置），和columnsSettings一起使用	—

表格方法调用示例

```
//
this.$refs.refName.exportExcel()
//
this.$refs.refName.exportCurrentPage()
//
this.$refs.refName.columnsSettings()
//
this.$refs.refName.resetColumnSettings()
```

基础示例完整版

```
<template>
  <!-- tableIndex -->
  <!-- singleSelectselection@selection-change -->
  <!-- api -->
  <!-- data -->
  <!-- fieldList -->
  <!-- extraDicts -->
  <!-- moduleNamemodule -->
  <!-- module -->
  <sv-search-table
    ref="SvSearchTable"
    :tableIndex="true"
    :singleSelect="true"
    :api="customPage"
    :data.sync="tableInfo.data"
    :field-list="tableInfo.fieldList"
    :extraDicts="tableInfo.extraDicts"
    moduleName="SvSearchTableDemo"
    module="SvSearchTableDemo"
    @selection-change="selectionChange"
  >
    <!-- -->
    <el-button type="primary" class="filter-item" @click="columnsSettings"></el-button>
    <el-button type="primary" class="filter-item" @click="resetColumnSettings"></el-button>
    <el-button type="primary" class="filter-item" @click="exportExcel"></el-button>
    <el-button type="primary" class="filter-item" @click="exportCurrentPage"></el-button>
    <span style="font-size: 18px; color: #909399;"> | </span>
    <el-button type="primary" class="filter-item" @click="handleAdd"></el-button>
    <el-button type="primary" class="filter-item" @click="handleEdit"></el-button>
    <el-button type="primary" class="filter-item" @click="handleView"></el-button>
    <el-button type="danger" class="filter-item" @click="handleDelete"></el-button>
    <!-- fieldListtypeslotcolprop -->
    <template v-slot:col-name="{ row }">
      <a style="color: #409eff" @click="handleView(row)">{{ row.name }}</a>
    </template>
  </sv-search-table>
</template>

<script>
import { customPage } from '@views/admin/demo/api'
import { mapGetters } from 'vuex'
export default {
  name: 'SvSearchTableDemo',
  data() {
    return {
      customPage,
      tableInfo: {
        data: [],
        fieldList: [
          {
            inputType: 'input', //
            tableAlias: 'b', //
            type: 'slot', // linkchildrentagimagefile
            prop: 'name',
            label: '',
            width: 120, // minWidth
            showOverflowTooltip: true // ...
          },
          {
            inputType: 'input',
            prop: 'username',
            label: '',
            width: 120,
```

```

        showOverflowTooltip: true
    },
    {
        inputType: 'select',
        operation: 'in', // eq
        prop: 'sex',
        label: '',
        dictType: 'comm_sex', //
        initValue: ['man'], //
        width: 120,
        showOverflowTooltip: true
    },
    //
    {
        inputType: 'input',
        label: '',
        prop: 'groupNames',
        minWidth: 100, // minWidth
        showOverflowTooltip: true
    },
    {
        inputType: 'select',
        operation: 'in',
        dictType: 'isSuperAdmin', //
        type: 'tag',
        tagOptions: {
            // tag
            0: {
                type: 'primary',
                name: ''
            },
            1: {
                type: 'warning',
                name: ''
            }
        },
        prop: 'isSuperAdmin',
        label: '',
        width: 120,
        showOverflowTooltip: true
    },
    {
        inputType: 'number',
        inputHidden: true, //
        prop: 'description',
        label: '',
        minWidth: 100,
        showOverflowTooltip: true
    },
    {
        inputType: 'date',
        prop: 'crtTime',
        label: '',
        width: 160,
        initValue: this.moment(new Date()).format('YYYY-MM-DD'), //
        showOverflowTooltip: true
    },
    {
        inputType: 'datetimerange',
        prop: 'updTime',
        label: '',
        width: 160,
        //
        initValue: [
            this.moment(new Date()).format('YYYY-MM-DD HH:mm:ss'),
            this.moment(new Date()).format('YYYY-MM-DD HH:mm:ss')
        ],
        showOverflowTooltip: true
    }
],
//
extraDicts: {
    isSuperAdmin: [
        {
            value: 1,
            labelDefault: ''
        },
        {
            value: 0,
            labelDefault: ''
        }
    ]
}
},

```



```

        currentUser: undefined,
        //
        userManagerBtnAdd: false,
        userManagerBtnEdit: false,
        userManagerBtnDel: false
    },
    computed: {
        // vuex
        ...mapGetters(['elements'])
    },
    created() {
        //
        this.userManagerBtnAdd = this.elements['userManager:btn_add']
        this.userManagerBtnEdit = this.elements['userManager:btn_edit']
        this.userManagerBtnDel = this.elements['userManager:btn_del']
    },
    methods: {
        //
        selectionChange(row) {
            this.currentUser = row
        },
        //
        handleView(row) {
            //
            if (row) this.currentUser = row
            if (!this.currentUser) return this.$message({ type: 'warning', title: '' })
            //
            // namerouter/index.jsname
            // title
            // status(status)
            // idid$openTagididuserIdpkName: 'userId'userId: this.currentUser.id
            this.$openTag(this, {
                name: 'SvTableEditDemo',
                title: `-${this.currentUser.name}`,
                status: 'view',
                id: this.currentUser.id
            })
        },
        //
        handleAdd() {
            //
            // namerouter/index.jsname
            // title
            // status(status)
            this.$openTag(this, {
                name: 'SvTableEditDemo',
                title: '',
                status: 'create'
            })
        },
        //
        handleEdit() {
            if (!this.currentUser) return this.$message({ type: 'warning', title: '' })
            //
            // namerouter/index.jsname
            // title
            // status(status)
            // idid$openTagididuserIdpkName: 'userId'userId: this.currentUser.id
            this.$openTag(this, {
                name: 'SvTableEditDemo',
                title: `-${this.currentUser.name}`,
                status: 'edit',
                id: this.currentUser.id
            })
        },
        //
        handleDelete() {
            if (!this.currentUser) return this.$message({ type: 'warning', title: '' })
            this.$confirm('', '', {
                confirmButtonText: '',
                cancelButtonText: '',
                type: 'warning'
            })
            .then(() => {
                //
            })
            .catch(() => {
                console.log('')
            })
        },
        // module
        columnsSettings() {
            this.$refs.SvTable.columnsSettings()
        }
    }
}

```

```
    },  
    //  
    resetColumnSettings() {  
        this.$refs.SvTable.resetColumnSettings()  
    },  
    // maxExportRows10000  
    // exportApi  
    exportExcel() {  
        this.$refs.SvTable.exportExcel()  
    },  
    //  
    exportCurrentPage() {  
        this.$refs.SvTable.exportCurrentPage()  
    }  
}  
}
```

</script>