

【组件】SvTable-列表

SvTable表格

基于ElementUI的Table表格组件进行二次封装，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\SvTable\index.vue

示例代码：src\views\admin\demo\SvTableDemo.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
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
query	列表查询条件	Object	否	{}
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	—
tableWidth	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	—	—	—

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
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，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	—	—

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 // ...
          }
        ]
      }
    }
  }
}
```

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。

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。

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>
  <div class="app-container">
    <div class="filter-container">
      <!-- -->
      <sv-input
        class="filter-item"
        v-model="listQuery.name"
        placeholder=""
        style="width: 200px;"
        @handleSave="handleFilter"
        @keyup.enter.native="handleFilter"
      >
    </sv-input>
  </div>
</template>
```

```

<!-- -->
<el-input
  class="filter-item"
  v-model="listQuery.username"
  placeholder=""
  style="width: 200px;"
  @keyup.enter.native="handleFilter"
>
</el-input>
<!-- -->
<el-select
  class="filter-item"
  v-model="listQuery.sex"
  placeholder=""
  style="width: 200px;"
  @keyup.enter.native="handleFilter"
>
  <el-option v-for="item in sexOptions" :key="item.id" :label="item.labelDefault" :value="item.value">
  </el-option>
</el-select>
<el-button class="filter-item" type="primary" @click="handleFilter"></el-button>
<el-button class="filter-item" type="primary" @click="handleReset"></el-button>
<!-- -->
<div>
  <!-- -->
  <el-button class="filter-item" type="primary" @click="columnsSettings"></el-button>
  <el-button class="filter-item" type="primary" @click="resetColumnSettings"></el-button>
  <el-button class="filter-item" type="primary" @click="exportCurrentPage"></el-button>
  <el-button class="filter-item" type="primary" @click="exportCurrentPage"></el-button>
  <!-- -->
  <span style="font-size: 18px; color: #909399;"> | </span>
  <el-button class="filter-item" type="primary" @click="handleAdd"></el-button>
  <el-button class="filter-item" type="primary" @click="handleEdit"></el-button>
  <el-button v-if="btnDelete" class="filter-item" type="danger" @click="handleDelete"></el-button>
  <el-button class="filter-item" type="primary"></el-button>
  <el-button class="filter-item" type="info"></el-button>
</div>
</div>
<!-- tableIndex -->
<!-- singleSelectselection@selection-change -->
<!-- api -->
<!-- refresh -->
<!-- data -->
<!-- fieldList -->
<!-- listQuery -->
<!-- handle -->
<!-- btnHideSet -->
<!-- btnDisableSet -->
<!-- cell-style -->
<!-- row-style:stripe="false" -->
<!-- exportApiexportApi -->
<!-- maxExportRows10000 -->
<!-- exportFileName -->
<!-- module -->
<sv-table
  ref="SvTable"
  :tableIndex="true"
  :singleSelect="true"
  :api="pageQuery"
  :refresh="tableInfo.refresh"
  :data.sync="tableInfo.data"
  :fieldList="tableInfo.fieldList"
  :query="listQuery"
  :handle="tableInfo.handle"
  :btnHideSet="btnHideSet"
  :btnDisableSet="btnDisableSet"
  :cell-style="tableCellStyle"
  :row-style="tableRowStyle"
  @selection-change="selectionChange"
  :maxExportRows="20000"
  exportFileName=""
  module="SvTableDemo"
>
  <!-- fieldListtypeslotcolprop -->
  <template v-slot:col-name="{ row }">
    <a style="color: #409eff" @click="handleView(row)">{{ row.name }}</a>
  </template>
  <!-- btevent -->
  <template v-slot:bt-removeGroupRelate="{ data }">
    <el-button
      type="text"
      :disabled="data.item.event === 'removeGroupRelate' && data.row.id === '1'"
      @click="handleRemoveGroupRelate(data.row)"
      style="color: red"
    >

```

```

    >

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

    </el-button>
  </template>
</sv-table>
</div>
</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 // ...
          },
          //
          {
            label: '',
            prop: 'username',
            width: 120, // minWidth
            showOverflowTooltip: true
          },
          //
          {
            label: '',
            prop: 'groupNames',
            minWidth: 100, // minWidth
            showOverflowTooltip: true
          },
          //
          {
            prop: 'sex',
            label: '',
            dictType: 'comm_sex', //
            width: 100 // minWidth
          },
          //
          {
            type: 'tag',
            prop: 'isSuperAdmin',
            label: '',
            width: 120, // minWidth
            // tagOptionkeytypename
            tagOptions: {
              0: {
                type: 'primary',
                name: ''
              },
              1: {
                type: 'warning',
                name: ''
              }
            }
          },
          //
          {
            type: 'tag',
            prop: 'isDisabled',
            label: '',
            width: 100, // minWidth
            // tagOptionkeytypename

```

```

        tagOptions: {
          0: {
            type: 'success',
            name: ''
          },
          1: {
            type: 'danger',
            name: ''
          }
        },
        //
        {
          prop: 'description',
          label: '',
          minWidth: 100, // width
          showOverflowTooltip: true
        }
      ],
      //
      handle: {
        //
        width: 340, // minWidth
        // align: center, // center
        // fixed: 'right', //
        btList: [
          {
            type: 'text',
            label: '',
            event: 'relateGroup', //
            show: true //
          },
          {
            type: 'text',
            label: '',
            event: 'relateDepart', //
            show: true //
          },
          {
            slot: true, //
            event: 'removeGroupRelate',
            show: true
          },
          {
            slot: true, //
            event: 'removeDepartRelate',
            show: true
          }
        ]
      }
    },
    currentRow: undefined, //
    //
    listQuery: {
      name: undefined,
      username: undefined,
      isDeleted: 0
    },
    //
    btnDelete: false
  }
},
computed: {
  ...mapGetters(['elements']),
  //
  sexOptions() {
    return this.$store.getters.dicts.comm_sex
  }
},
created() {
  //
  this.btnDelete = this.elements['userManager:btn_del']
},
mounted() {
  //
  this.getList()
},
methods: {
  getList() {
    this.tableInfo.refresh = Math.random()
  },
  //
  selectionChange(row) {
    this.currentRow = row
  }
}

```



```

    },
    //
    handleFilter() {
        this.getList()
    },
    //
    handleReset() {
        this.listQuery = {
            name: undefined,
            username: undefined,
            isDeleted: 0
        }
    },
    //
    btnHideSet(event, row) {
        if (event === 'relateGroup' && row.id === '1') return true
    },
    //
    btnDisableSet(event, row) {
        if (event === 'relateDepart' && row.id === '1') return true
    },
    //
    tableCellStyle({ row, column }) {
        if (row.id === '1' && column.property === 'username') {
            return { background: '#67C23A' }
        }
        return {}
    },
    // ***stripefalse
    tableRowStyle({ row }) {
        if (row.id === '1') {
            return { background: '#F56C6C' }
        }
        return {}
    },
    // module
    columnsSettings() {
        this.$refs.SvTable.columnsSettings()
    },
    //
    resetColumnSettings() {
        this.$refs.SvTable.resetColumnSettings()
    },
    //
    // maxExportRows10000
    // exportApi
    exportExcel() {
        this.$refs.SvTable.exportExcel()
    },
    //
    exportCurrentPage() {
        this.$refs.SvTable.exportCurrentPage()
    },
    /**
     * status
     * 1. router/index.js
     * 2. path: 'SvTableEditDemo/:status/:pageId/:id?/:title?id?'
     * 3. pageId$openTagstatusidtitle
     */
    //
    handleView(row) {
        //
        this.currentRow = row
        if (!this.currentRow) return this.svMessage({ type: 'warning', title: '' })
        //
        // namerouter/index.jsname
        // title
        // status(status)
        // idid$openTagididUserIdpkName: 'userId'userId: this.currentRow.id
        this.$openTag(this, {
            name: 'SvTableEditDemo',
            title: `-${this.currentRow.name}`,
            status: 'view',
            id: this.currentRow.id
        })
    },
    //
    handleAdd() {
        //
        // namerouter/index.jsname
        // title
        // status(status)
        this.$openTag(this, {
            name: 'SvTableEditDemo',
            title: '',

```

```

        status: 'create'
    })
},
//
handleEdit() {
    if (!this.currentRow) return this.svMessage({ type: 'warning', title: '' })
    //
    // namerouter/index.jsname
    // title
    // status(status)
    // idid$openTagididuserIdpkName: 'userId'userId: this.currentRow.id
    this.$openTag(this, {
        name: 'SvTableEditDemo',
        title: `-${this.currentRow.name}`,
        status: 'edit',
        id: this.currentRow.id
    })
},
//
handleDelete() {
    if (!this.currentRow) return this.svMessage({ type: 'warning', title: '' })
    this.$confirm('', '', {
        confirmButtonText: '',
        cancelButtonText: '',
        type: 'warning'
    })
    .then(() => {
        //
    })
    .catch(() => {
        console.log('')
    })
},
//
handleRemoveGroupRelate(row) {
    console.log(row)
}
}
}
</script>

```