Table 组件使用

参数字段	参数说明	参数类型	使用
showTableHeader	是否展示表格头部，默认为true(展示)	Boolean	:showTableHeader="false"
CustomSetVisible	自定义列设置隐藏，默认为true(展示)	Boolean	-
SortVisible	排序隐藏，默认为true(展示)	Boolean	-
ScreenVisible	筛选隐藏，默认为true(展示)	Boolean	-

RefreshVisible	刷新隐藏，默认为true(展示)	Boolean	-
ColumnSetVisible	列设置隐藏，默认为true(展示)	Boolean	-
CellSetVisible	单元格设置隐藏，默认为true(展示)	Boolean	-
UserPolicyVisible	用户策略选择隐藏，默认为false(隐藏)	Boolean	-
UserPolicySetVisible	用户策略设置隐藏，	Boolean	-

	默认为false(隐藏)		
AddVisible	新增按钮隐藏，默认为false(隐藏)	Boolean	-
EditVisible	修改按钮隐藏，默认为false(隐藏)	Boolean	-
DelVisible	删除按钮隐藏，默认为true(显示)	Boolean	-
ImportVisible	导入按钮隐藏，默认为true(展示)	Boolean	-

ExportVisible	导出按钮隐藏，默认为true(展示)	Boolean	-
fpagination	分页条设置隐藏，默认为false(隐藏)	Boolean	-
DefaultFirstRow	默认选中第一行，默认为false(隐藏)	Boolean	-
SelectionColumnVisible	选择框展示，默认是true(展示)	Boolean	-
OperateColumnVisible	操作列隐藏，默认为fals	Boolean	操作列只有当 (EditVisible || DelVisible) && OperateColumnVisible 为 true 时，才会显示

	e(隐藏)		
MoreVisible	更多按钮隐藏，默认为true(展示)	Boolean	-
IndexColumnVisible	序号列隐藏，默认为true(展示)	Boolean	-
SelectionColumnVisible	选中行列隐藏，默认为true(展示)	Boolean	-
Stripe	斑马纹显示，默认为true(展示)	Boolean	-
showTableFooter	是否展示foot	Boolean	-

	er，默认为true(展示)		
showYhlTableCrad	是否展示crad，默认为true(展示)	Boolean	-
keyCode	组件Key	String	必传，区分表格
titleName	表格标题	String	titleName="表格组件布局应用表格标题"
componentKey	组件Key	String	componentKey = "STUDENTKEY"
fullImportAction	全量导入执行接口	String	必填，不传则导入全量按钮不存在
incrementImportAction	增量导入执行接口	String	必填，不传则导入增量按钮不存在
rowKey	行关键字段	String	rowKey=“id”
rowChildren	子行对象属性	String	同 element 组件的 tree-props 的 children 属性
ObjectType	数据源类型，必	String	-

	传，否则自定义列不可使用		

tableColumns	表格列头	Array	:tableColumns = tableColumns
			tableColumns: [
			{
			prop: 'name',
			label: '名字',
			dataType: 'CHARACTER', // NUMERICAL、DATE、DURATION
			width: '120', // 列宽
			align: "center", // center/left/right 不加 align 字段，默认 CHARACTER 左对齐，其他居中
			fixed: 0, // 是否固定 0/1
			sortBy: 1, // 排序
			showType: 'TEXT', // 必传
			fshow: 1, // 是否展示
			enumKey: "SexEnum",
			fscope:false, // 是否为操作列
			}
			]
			// 系统列-新增列：之前保存版本的配置中不存在新增的列
			// 当新增一个系统列，在表格中会默认不显示，在列设置把新增的一列勾选后，即可看到
			// 操作列的使用方法
			<template slot="column" slot-scope="scope">
			<el-button @click="rowClick1(scope.column, scope.row)">操作列按钮</el-button>
			</template>

tableData	表格数据	Array	tableData = [
			{
			age: 23,
			birthDate: 930499200000,
			name: '张三',
			pstudentId: "ff30c266-6afb-11ed-ae23-00505697f8a5",
			...
			}

			]
			// 控制任意行是否能勾选
			
			tableData = [
			{
			age: 23,
			birthDate: 930499200000,
			name: '张三',
			pstudentId:
			"ff30c266-6afb-11ed-ae23-00505697f8a5",
			selectable: false, // 非必传，默认能勾选（true）
			...
			}
			]

userPolicy	用户策略	Array	userPolicy = [{label: "默认视图", value:"69b8c4b1-7130-426e-938f-67e6ef975e89"}, ...] // 切换版本的时候会通过 ChangeUserPolicy(value) 将单个用户策略的 value 带出

customColumns	自定义列	Array	customColumns = [
			{
			
			id:"37ddcdad-58e7-48d7-916b-b475d6cdefcd",
			prop:"property2",
			label:"3",
			dataType:"NUMERICAL",
			width:"120",
			showType:"ICON", // TEXT：文本列，ICON：图表列， CHART：图表列
			expression:"1", // 公式
			expressionIcon:[
			
			{id:"686f8a708c0c11edb58a15b987c62785",expression:"",icon:"el-icon-check",color:"#FF4500"}
			],
			chart:"PROPORTIONBAR", // PROPORTIONBAR，PROPORTIONPIE，CHART
			align:"center"
			}

			]

enums	枚举集合	Array	[
			{
			key:"SexEnum",
			values:[{"label":"是","value":"YES"},{"label":"否","value":"NO"}]
			}
			]

mergeColumms	合并单元格列集合	Array	// 当 col02 列上下相邻两个单元格内容相同，则会上下合并
			// rowChildren 若有值，则不能合并
			mergeColumms: [
			{
			prop: 'col02'
			}
			]

tree	表格为树状结构时，默认展开层级	Array	tree=[1, 2] 展开一二层级
total	表格数据总行数	Number	:total = 0
tableIndent	展示树形数据时，树节点的缩进(默认为16)	Number	:tableIndent="60"

fullImportData	全量导入附带参数	Object	// 同 el-upload 组件的 data 属性 fullImportData: { type: 'FULL_IMPORT' }
incrementImportData	增量导入附带参数	Object	// 同 el-upload 组件的 data 属性 incrementImportData: { type: 'INCREMENTAL_IMPORT' }
requestHeaders	设置上传的请求头部	Object	同 el-upload 组件的 headers 属性

currentUserPolicy	当前选择的版本	Object	currentUserPolicy:{
			btnSet: {}, // 按钮配置
			cellSets: [], // 单元格设置
			columnset: [], // 列设置
			d_sorts:[],
			d_table: [],
			default: 'YES',
			global: 'NO',
			id: 'iufiquvbqiebviwvreve',
			name: '默认试图',
			objectType: 'STUDENT',
			screen:[],// 筛选
			sorts:[] //排序
			}

QueryComplate	查询方法	Function	QueryComplate(_pageNum, _pageSize, _screens, _sorts) {
			console.log({_pageNum, _pageSize, _screens, _sorts})
			{
			_pageNum: 1,
			_pageSize: 50,
			_screens: [// 筛选配置
			{
			property: "name",
			label: "名字",
			fieldType: "CHARACTER",
			connector: "and",

			symbol: "EQUAL",
			fixed:"YES", // YES：固定值，NO：字段
			value1: "张三",
			value2:"" //当是时间段时，value1是起始时间，value2 是结束时间
			}
			],
			_sorts: [
			{
			property: "name",
			label: "名字",
			fieldType: "CHARACTER",
			seqNum: 1,
			sortOrder: "asc" // 升序
			}
			]
			}
			}

UserPolicyComplate	用户版本保存	Function	UserPolicyComplate(_userPolicy) {
			// _userPolicy:单个组件的配置
			}

ChangeUserPolicy	切换版本	Function	// 表格版本切换
			ChangeUserPolicy(_values) {
			console.log('curUserPolicy=====>', _values)
			// _values：所选 userPolicy 对应版本的 value
			}

CustomColumnSave	自定义列保存	Function	CustomColumnSave(_data, _pageNum, _pageSize, _screens, _sorts) {
			console.log({_data, _pageNum, _pageSize, _screens, _sorts});
			{
			_data: {
			expressionId: "1fef7893-5740-4608-88b7-4fe56c8d7edb",
			attributeCode: "property1", //
			attributeName: "自定义列名称",
			dataType: "NUMERICAL",
			expression: "1",
			expressionIcon: [],
			columnType: "TEXT",

			chart: "",
			componentKey: "STUDENTSTABLEKEY"
			},
			_pageNum: 1,
			_pageSize: 100,
			_screens: [], // 筛选
			_sorts: [] // 排序
			}
			}
			// 保存成功之后，重新获取下版本的配置，并刷新下列表

CustomColumnDel	自定义删除	Function	CustomColumnDel(data) {
			console.log('CustomColumnDelData===>', data)
			// CustomColumnDelData===> 1fef7893-5740-4608-88b7-4fe56c8d7edb
			}

ExportTemplate	导出模板方法	Function	ExportTemplateFn() {
			let url = `plant/exportPlantFactoryExcel`;
			myExportTemplate(url).then(response => {
			console.log(response, '--------------response');
			});
			}

ExportData	导出数据模板方法	Function	ExportData() {
			let url = `plant/exportPlantData`;
			let params = {
			queryCriteriaParam: this.filterObj.queryCriteriaParam,
			sortParam: this.filterObj.sortParam
			};
			myExportData(url, params)
			}

AddData	新增方法	Function	AddDataFun(tableData) {
			console.log("新建提交的表单数据：", tableData);
			}

EditData	修改方法	Function	EditDataFun(tableData) { console.log("表格一行的数据数据：", tableData);

			}

DelRows	批量删除方法	Function	DelRowsFun(rows) {
			// rows：[]
			console.log('要删除的多行数据：',rows)
			}

DelRow	删除单行方法	Function	DelRowFun(row) {
			// row：{}
			console.log('要删除的单行数据：',row)
			}

RowClick	单击行方法	Function	RowClickFun(row) {
			// row：{}
			console.log('所点击的行数据：',row)
			}

ExpandChange	多层级展开、关闭行方法	Function	ExpandChange(row, key) {
			console.log({row, key}) // row:所点击的行数据，key：展开/闭合
			}

SelectionChange	勾选行触发方法	Function	SelectionChangeFun(selectArray) {
			console.log({selectArray}) // 所有已选择的行数据
			}

ImportChange	导入后更新方法	Function	ImportChange(file, fileList) {
			console.log({file, fileList})
			{
			file: {
			status: "fail",
			name: "index.html",
			size: 13150,
			percentage: 100,
			uid: 1672032848716,
			raw: {
			uid: 1672032848716
			}
			},
			fileList: []
			}

			}

LoadRowNode	点击行树节点加载数据方法	Function	LoadRowNode (tree, treeNode, resolve) {
			let res = [
			{
			col01: 41,
			col02: "4-11",
			col03: Math.random()*10000,
			col04: "2022-07-19 05:20:22",
			customMap: { costom01: 4 }
			},{
			col01: 42,
			col02: "4-12",
			col03: Math.random()*10000,
			col04: "2022-07-19 05:20:22",
			customMap: { costom01: 4 }
			}
			]
			// return res
			resolve(res)
			}

ButtonCollect	按钮选择性的收集到更多里	Function	ButtonCollect(BTNS) {
			// BTNS,所有按钮的 key,返回对应的 key,就会将返回的 key 放置在外面
			return ['exportOut', 'customOut', 'cellOut', 'addColumnOut', 'delOut']
			}