<template>

<div class="ClassManage">

<el-card class="box-card">

<div slot="header" class="clearfix title">

<span class="iconfont icon-guanli"></span>

<span>班级管理</span>

</div>

<div class="text item">

<!-- 弹出编辑模态框 -->

<el-dialog title="编辑班级信息"

:visible.sync="dialogFormVisible"

width="520px"

top="5vh"

modal-append-to-body

>

<el-form :model="editFormData" size="small" ref="editFormData" :rules="rules">

<!-- 班级名称 -->

<el-form-item label="班级名称" prop="className">

<el-input type="text" v-model="editFormData.className" auto-complete="off"></el-input>

</el-form-item>

<!-- 相关课程 -->

<el-form-item label="相关课程" prop="classCourse">

<el-select v-model="editFormData.classCourse" placeholder="请选择相关课程">

<el-option

v-for="(item,index) in courseOptions"

:key="index"

:label="item.label"

:value="item.value">

</el-option>

</el-select>

</el-form-item>

<!--班型 -->

<el-form-item label="班级类型" prop="classSet">

<el-select v-model="editFormData.classSet" placeholder="请选择班型">

<el-option

v-for="(item,index) in classSetOptions"

:key="index"

:label="item.label"

:value="item.value">

</el-option>

</el-select>

</el-form-item>

<!--班级类别 -->

<el-form-item label="班级类别" prop="classType">

<el-radio-group v-model="editFormData.classType">

<el-radio label="春季班"></el-radio>

<el-radio label="暑假班"></el-radio>

<el-radio label="秋季班"></el-radio>

<el-radio label="寒假班"></el-radio>

</el-radio-group>

</el-form-item>

<!-- 周课次数 -->

<el-form-item label="周课次数" prop="classCourses">

<el-input-number

v-model="editFormData.classCourses"

:min="1"

:max="10"

label="描述文字">

</el-input-number>

</el-form-item>

<!-- 每周上课日期 -->

<el-form-item label="上课日期" prop="classDate">

<el-select v-model="editFormData.classDate" multiple placeholder="请选择每周上课日期">

<el-option

v-for="(item,index) in weekOptions"

:key="index"

:label="item.label"

:value="item.value">

</el-option>

</el-select>

</el-form-item>

<!-- 每天上课时间 -->

<el-form-item label="上课时间" prop="classTime">

<el-time-select

v-model="editFormData.classTime"

:picker-options="{

start: '08:30',

step: '00:15',

end: '20:00'

}"

placeholder="选择上课时间">

</el-time-select>

</el-form-item>

<!-- 上课教室 -->

<el-form-item label="上课教室" prop="classroom">

<el-select v-model="editFormData.classroom" placeholder="请选择上课教室">

<el-option

v-for="(item,index) in classroomOptions"

:key="index"

:label="item.label"

:value="item.value">

</el-option>

</el-select>

</el-form-item>

<!-- 上课老师 -->

<el-form-item label="上课老师" prop="teacher">

<el-select v-model="editFormData.teacher" placeholder="请选择上课老师">

<el-option

v-for="(item,index) in teacherOptions"

:key="index"

:label="item.label"

:value="item.value">

</el-option>

</el-select>

</el-form-item>

</el-form>

<!-- 尾部 -->

<div slot="footer" class="dialog-footer">

<el-button @click="dialogFormVisible = false" size="small">取 消</el-button>

<el-button type="success" @click="saveEdit" size="small">确认修改</el-button>

</div>

</el-dialog>

<!-- 表格 -->

<el-table

ref="multipleTable"

:data="classTableData"

tooltip-effect="dark"

style="width: 100%"

@selection-change="handleSelectionChange"

header-align="center"

>

<el-table-column

type="selection"

align="center"

width="55">

</el-table-column>

<!-- 班级名称 -->

<el-table-column

prop="classname"

label="班级名称"

align="center"

>

</el-table-column>

<!-- 相关课程 -->

<el-table-column

prop="classcourse"

label="相关课程"

align="center"

>

</el-table-column>

<!--班型 -->

<el-table-column

prop="classset"

align="center"

label="班型"

show-overflow-tooltip>

</el-table-column>

<!--班级类别 -->

<el-table-column

prop="classtype"

align="center"

label="班级类别"

show-overflow-tooltip>

</el-table-column>

<!--周课次数 -->

<el-table-column

align="center"

prop="classcourses"

label="周课次数"

width="90"

show-overflow-tooltip>

</el-table-column>

<!--上课日期 -->

<el-table-column

prop="classdate"

align="center"

label="上课日期"

show-overflow-tooltip>

</el-table-column>

<!--上课时间 -->

<el-table-column

prop="classtime"

align="center"

label="上课时间"

show-overflow-tooltip>

</el-table-column>

<!--上课教室 -->

<el-table-column

prop="classroom"

align="center"

label="上课教室"

show-overflow-tooltip>

</el-table-column>

<!--上课老师 -->

<el-table-column

prop="teacher"

align="center"

label="上课老师"

show-overflow-tooltip>

</el-table-column>

<!-- 编辑 -->

<el-table-column label="操作" align="center" width="250">

<template slot-scope="scope">

<el-button

size="mini"

type="success"

@click="handleEdit(scope.row.id)">

<span class="el-icon-edit"></span>

编辑</el-button>

<el-button

size="mini"

type="danger"

@click="handleDelete(scope.row.id)">

<span class="el-icon-delete"></span>

删除</el-button>

</template>

</el-table-column>

</el-table>

<!-- 分页 -->

<el-pagination

@size-change="handleSizeChange"

@current-change="handleCurrentChange"

:current-page="currentPage"

:page-sizes="[3, 5, 10, 20]"

:page-size="pageSize"

layout="total, sizes,prev, pager, next, jumper"

:total="total">

</el-pagination>

<!-- 按钮组 -->

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

<el-button type="danger" @click="batchDel">批量删除</el-button>

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

</div>

</div>

</el-card>

</div>

</template>

<script>

export default {

data(){

return{

// 回填数据

editFormData:{

className:"",

classCourse: "",

classSet:"",

classType:"",

classCourses:"",

classDate:[],

classTime:"",

classroom:"",

teacher:""

},

// 验证规则

rules:{

// 班级名称验证

className:[

{required:true,message: '请填写班级名称', trigger: 'blur'}

],

// 相关课程验证

classCourse:[

{required:true,message: '请选择性别',trigger: 'change'}

],

// 班型验证

classSet:[

{required:true,message: '请选择班型', trigger: 'change'}

],

// 班级类型验证

classType:[

{required:true,message: '请选择班级类型',trigger: 'change'}

],

// 周课次数

classCourses:[

{required:true,message: '请选择周课次数',trigger: 'change'}

],

// 上课日期

classDate:[

{required:true,message: '请选择上课日期',trigger: 'change'}

],

// 教室验证

classroom:[

{required:true,message: '请选择上课教室',trigger: 'change'}

],

// 老师验证

teacher:[

{required:true,message: '请选择上课老师',trigger: 'change'}

],

// 上课时间

classTime:[

{required:true,message: '请选择上课时间',trigger: 'change'}

]

} ,

classTableData: [],

// 课程选择

courseOptions: [

{

value: '少儿英语',

label: '少儿英语'

}, {

value: '小学奥数',

label: '小学奥数'

}, {

value: '小学数学同步',

label: '小学数学同步'

}, {

value: '小学作文',

label: '小学作文'

}

],

// 班型选择

classSetOptions:[

{

label:"4~8人精品班",

value:"4~8人精品班"

},

{

label:"9~20人小班",

value:"9~20人小班"

},

{

label:"20~40人大班",

value:"20~40人大班"

},

],

// 星期选择

weekOptions:[

{

label:"星期日",

value:"星期日"

},

{

label:"星期一",

value:"星期一"

},

{

label:"星期二",

value:"星期二"

},

{

label:"星期三",

value:"星期三"

},

{

label:"星期四",

value:"星期四"

},

{

label:"星期五",

value:"星期五"

},

{

label:"星期六",

value:"星期六"

},

],

// 教室选择

classroomOptions:[

{

label:"教室一",

value:"教室一"

},

{

label:"教室二",

value:"教室二"

},

{

label:"教室三",

value:"教室三"

},

{

label:"教室四",

value:"教室四"

},

{

label:"教室五",

value:"教室五"

},

{

label:"教室六",

value:"教室六"

},

{

label:"教室七",

value:"教室七"

},

],

// 教师选择

teacherOptions:[

{

label:"张老师",

value:"张老师"

},

{

label:"王老师",

value:"王老师"

},

{

label:"李老师",

value:"李老师"

},

{

label:"马老师",

value:"马老师"

},

{

label:"陈老师",

value:"陈老师"

}

],

// 模态框是否可见

dialogFormVisible:false,

formLabelWidth:"80px",//模态框标签宽度

multipleSelectionId: [],

currentPage: 1,//当前页码

total:0,//总页数

pageSize:3,//每页显示数量

// 保存id

editId:""

}

},

methods:{

// 分页获取列表函数

getListByPages(){

// 收集参数

let params={

pageSize:this.pageSize,

currentPage:this.currentPage

}

// 向后端发送请求

this.request.get('/class/classmanage',params)

.then(res=>{

// 接收后端发送回来的数据

let {total,data}=res;

// 赋值给页面总条数

this.total=total;

// 赋值给表格数据

this.classTableData=data;

// 判断当前页中数据是否为空，或当前页是否为第一页

if(data.length===0&&this.currentPage!==1){

this.currentPage-=1;

// 调用自己

this.getListByPages();

}

})

.catch(err=>{

console.log(err);

})

},

toggleSelection(rows) {

if (rows) {

rows.forEach(row => {

this.$refs.multipleTable.toggleRowSelection(row);

});

} else {

this.$refs.multipleTable.clearSelection();

}

},

handleSelectionChange(val) {

this.multipleSelectionId = val.map(val=>val.id);

},

// 修改回填

handleEdit(id) {

// 保存id

this.editId=id;

// 出现模态框

this.dialogFormVisible=true;

// 发送请求

this.request.get('/class/editclass',{id})

.then(res=>{

// 处理上课日期字符串成数组

let classdate = res[0].classdate.split('/');

// 回填数据

this.editFormData.className= res[0].classname;

this.editFormData.classCourse= res[0].classcourse;

this.editFormData.classSet= res[0].classset;

this.editFormData.classType= res[0].classtype;

this.editFormData.classCourses= res[0].classcourses;

this.editFormData.classDate= classdate;

this.editFormData.classTime= res[0].classtime;

this.editFormData.classroom= res[0].classroom;

this.editFormData.teacher= res[0].teacher;

})

.catch(err=>{

console.log(err);

})

},

// 保存数据

saveEdit(){

// 收集参数

let params = {

id:this.editId,

classname:this.editFormData.className,

classcourse: this.editFormData.classCourse,

classset:this.editFormData.classSet,

classtype:this.editFormData.classType,

classcourses:this.editFormData.classCourses,

classdate:this.editFormData.classDate.join('/'),

classtime:this.editFormData.classTime,

classroom:this.editFormData.classroom,

teacher:this.editFormData.teacher,

};

// 判断：表单通过验证后发送请求到后台

this.$refs.editFormData.validate(valid=>{

if(valid){

// 发送请求

this.request.post('/class/saveclass',params)

.then(res=>{

// 接收后端传回来的数据

let { code,msg } = res;

if(code === 0){

// 显示是否修改界面

this.$confirm('是否确定修改该数据?', '提示', {

confirmButtonText: '确定',

cancelButtonText: '取消',

type: 'warning',

title:"温馨提示",

lockScroll:false,//将 body 滚动锁定

roundButton:true

}).then(()=>{

// 隐藏编辑框

this.dialogFormVisible = false;

// 弹出修改成功提示

this.$message({

type:"success",

message:msg

})

// 刷新列表

this.getListByPages();

}).catch(() => {

this.$message({

type: 'info',

message: '已取消修改'

});

});

}else if(code === 1){

// 弹出失败提示

this.$message.error(msg);

}

})

.catch(err=>{

console.log(err);

})

}else{

return false;

}

})

},

// 删除

handleDelete(id) {

// 显示是否删除界面

this.$confirm('此操作将删除该数据, 是否继续?', '提示', {

confirmButtonText: '确定',

cancelButtonText: '取消',

type: 'warning',

title:"温馨提示",

lockScroll:false,//将 body 滚动锁定

roundButton:true

}).then(()=>{

// 发送请求到后端

this.request.get('/class/classdel',{id})

.then(res=>{

// 接收后端发送回来的数据

let {code,msg}=res;

if(code===0){

this.$message({

type:"success",

message:msg

})

// 请求数据

this.getListByPages();

}else if(code===1){

this.$message.error(msg);

}

})

.catch(err=>{

console.log(err);

})

}).catch(() => {

this.$message({

type: 'info',

message: '已取消删除'

});

});

},

// 批量删除

batchDel(){

// 判断是否勾选了数据

if(this.multipleSelectionId.length>0){

// 弹出是否删除提示

this.$confirm('是否批量删除?', '提示', {

confirmButtonText: '确定',

cancelButtonText: '取消',

type: 'warning'

}).then(() => {

// 收集参数

let params = {

ids:this.multipleSelectionId

}

// 向后端发送参数

this.request.get('/class/batchdel',params)

.then(res=>{

// 接收传回来数据

let {code,msg}=res;

if(code===0){

this.$message({

type:"success",

message:msg

})

// 刷新页面

this.getListByPages();

}else if(code === 1){

this.$message.error(msg);

}

})

.catch(err=>{

console.log(err);

})

}).catch(() => {

this.$message({

type: 'info',

message: '已取消删除'

});

});

}else{

this.$message.error("你没有勾选任何选项！")

}

},

// 分页函数

handleSizeChange(val) {

// 赋值

this.pageSize=val;

this.getListByPages();

},

handleCurrentChange(val) {

// 赋值

this.currentPage=val;

this.getListByPages();

}

},

// 创建周期，钩子函数

created(){

this.getListByPages();

}

}

</script>

<style lang="less">

@import "./ClassManage.less";

</style>