<template>

<div class="GiftManage">

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

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

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

<span>礼品管理</span>

</div>

<div class="text item">

<!-- 弹出编辑模态框 -->

<el-dialog title="编辑礼品信息"

:visible.sync="dialogFormVisible"

width="580px"

top="5vh"

modal-append-to-body

>

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

<!--礼品分类 -->

<el-form-item label="礼品分类" prop="giftClassify">

<el-select v-model="editFormData.giftClassify" placeholder="请选择礼品分类">

<el-option

v-for="item in giftClassifyOptions"

:key="item.value"

:label="item.label"

:value="item.value">

</el-option>

</el-select>

</el-form-item>

<!-- 礼品名称 -->

<el-form-item label="礼品名称" prop="giftName">

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

</el-form-item>

<!-- 礼品码 -->

<el-form-item class="giftcode" label="礼品码" prop="giftCode">

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

<el-button type="danger" @click="generateGiftCode">生成礼品码</el-button>

</el-form-item>

<!-- 礼品所需积分 -->

<el-form-item label="所需积分" prop="costScore">

<el-input-number

v-model="editFormData.costScore"

:min="0"

label="描述文字">

</el-input-number> 积分

</el-form-item>

<!-- 礼品进价 -->

<el-form-item label="礼品进价" prop="price">

<el-input-number

v-model="editFormData.price"

:min="0"

label="描述文字">

</el-input-number> 元

</el-form-item>

<!-- 入库数量 -->

<el-form-item label="入库数量" prop="stockNum">

<el-input-number

v-model="editFormData.stockNum"

:min="1"

label="描述文字">

</el-input-number>

</el-form-item>

<!--礼品单位 -->

<el-form-item label="礼品单位" prop="giftUnit">

<el-select v-model="editFormData.giftUnit" placeholder="请选择礼品单位">

<el-option

v-for="item in giftUnitOptions"

:key="item.value"

:label="item.label"

:value="item.value">

</el-option>

</el-select>

</el-form-item>

<!-- 是否促销 -->

<el-form-item label="是否促销" prop="isSale">

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

<el-radio label="促销"></el-radio>

<el-radio label="不促销"></el-radio>

</el-radio-group>

</el-form-item>

<!-- 礼品介绍 -->

<el-form-item label="礼品介绍" prop="giftDesc">

<el-input type="textarea" v-model="editFormData.giftDesc"></el-input>

</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>

<!-- 筛选器 -->

<div class="filterBox">

<!-- 分类 -->

<el-select v-model="classify" placeholder="请选择分类" size="small">

<el-option

v-for="item in giftClassifyOptions"

:key="item.value"

:label="item.label"

:value="item.value">

</el-option>

</el-select>

<!-- 关键字 -->

关键字：

<el-input type="text" v-model="keyword" auto-complete="off" size="small" placeholder="礼品名称，礼品码"></el-input>

<el-button type="success" @click="getListByPages" style="margin-left:10px" size="small">查询</el-button>

</div>

<!-- 表格 -->

<el-table

ref="multipleTable"

:data="giftTableData"

tooltip-effect="dark"

style="width: 100%"

@selection-change="handleSelectionChange"

header-align="center"

>

<el-table-column

type="selection"

align="center"

width="55">

</el-table-column>

<!-- 礼品名称 -->

<el-table-column

prop="giftname"

label="礼品名称"

align="center"

>

</el-table-column>

<!-- 礼品分类 -->

<el-table-column

prop="giftclassify"

label="礼品分类"

align="center"

>

</el-table-column>

<!--礼品码 -->

<el-table-column

prop="giftcode"

align="center"

label="礼品码"

show-overflow-tooltip>

</el-table-column>

<!--礼品所需积分 -->

<el-table-column

prop="costscore"

align="center"

label="礼品所需积分"

show-overflow-tooltip>

</el-table-column>

<!--礼品进价 -->

<el-table-column

align="center"

prop="price"

label="礼品进价"

show-overflow-tooltip>

</el-table-column>

<!--入库数量 -->

<el-table-column

prop="stocknum"

align="center"

label="库存"

show-overflow-tooltip>

</el-table-column>

<!--礼品单位 -->

<el-table-column

prop="giftunit"

align="center"

label="礼品单位"

show-overflow-tooltip>

</el-table-column>

<!--是否促销 -->

<el-table-column

prop="issale"

align="center"

label="是否促销"

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="info" @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>

// 引入验证函数

import { feesReg } from "@/utils/validator/validator.js";

export default {

data(){

// 礼品码验证

const validateCode=(rule, value, callback) => {

if(!value){

callback(new Error("礼品码不能为空"));//输出错误提示

}else if(!feesReg(value)){

callback(new Error("礼品码必须为大于0的正整数"));//输出错误提示

}else{

callback();//成功回调

}

}

return{

// 表单回填数据

editFormData:{

giftClassify:"",

giftName:"",

giftCode: "",

costScore:"",

price:"",

stockNum:"",

giftUnit:"",

isSale:"",

giftDesc:""

},

// 验证规则

rules:{

// 礼品分类验证

giftClassify:[

{required:true,message: '请选择礼品分类', trigger: 'change'}

],

// 礼品名称验证

giftName:[

{required:true,message: '请输入礼品名称', trigger: 'blur'}

],

// 礼品码验证

giftCode:[

{required:true,validator: validateCode,trigger: 'input'}

]

} ,

// 商品分类

classify:"",

keyword:"",

giftTableData: [],

// 商品分类

giftClassifyOptions: [

{

value: '全部',

label: '全部'

},

{

value: '学习用品',

label: '学习用品'

}, {

value: '食品类',

label: '食品类'

}, {

value: '代金券',

label: '代金券'

}, {

value: '玩具类',

label: '玩具类'

}, {

value: '电子产品',

label: '电子产品'

}, {

value: '日用品',

label: '日用品'

}

],

// 商品单位

giftUnitOptions: [

{

value: '个',

label: '个'

}, {

value: '本',

label: '本'

}, {

value: '支',

label: '支'

}, {

value: '辆',

label: '辆'

}, {

value: '张',

label: '张'

}, {

value: '条',

label: '条'

}

],

// 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,

classify:this.classify,

keyword:this.keyword

}

// 向后端发送请求

this.request.get('/gift/giftmanage',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.giftTableData=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) {

// 获取勾选框的id数组

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

},

// 编辑回填事件

handleEdit(id) {

// 保存id

this.editId=id;

// 出现模态框

this.dialogFormVisible=true;

// 发送请求

this.request.get('/gift/editgift',{id})

.then(res=>{

// 回填数据

this.editFormData.giftClassify=res[0].giftclassify;

this.editFormData.giftName=res[0].giftname;

this.editFormData.giftCode=res[0].giftcode;

this.editFormData.costScore=res[0].costscore;

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

this.editFormData.stockNum=res[0].stocknum;

this.editFormData.giftUnit=res[0].giftunit;

this.editFormData.isSale=res[0].issale,

this.editFormData.giftDesc=res[0].giftdesc

})

.catch(err=>{

console.log(err);

})

},

// 保存数据

saveEdit(){

// 收集参数

let params = {

id:this.editId,

giftclassify:this.editFormData.giftClassify,

giftname: this.editFormData.giftName,

giftcode:this.editFormData.giftCode,

costscore:this.editFormData.costScore,

price:this.editFormData.price,

stocknum:this.editFormData.stockNum,

giftunit:this.editFormData.giftUnit,

issale:this.editFormData.isSale,

giftdesc:this.editFormData.giftDesc

};

// 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('/gift/savegift',params)

.then(res=>{

// 接收后端传回来的数据

let { code,msg } = res;

if(code === 0){

// 显示是否删除界面

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;

}

})

},

// 删除事件

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('/gift/giftdel',{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('/gift/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) {

// 获取每页显示数量

this.pageSize=val;

// 刷新页面

this.getListByPages();

},

handleCurrentChange(val) {

// 获取当前页

this.currentPage=val;

// 刷新页面

this.getListByPages();

},

// 生成礼品码

generateGiftCode(){

this.editFormData.giftCode = Math.ceil(Math.random()\*1000000000);

}

},

// 钩子函数

created(){

this.getListByPages()

}

}

</script>

<style lang="less">

@import "./GiftManage.less";

</style>