<template>

<div class="GiftAdd">

<!-- 面板 -->

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

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

<span class="el-icon-d-arrow-right"></span>

<span>新增礼品</span>

</div>

<div class="text item">

<el-form

:model="giftAddForm"

status-icon

:rules="rules"

ref="giftAddForm"

label-width="100px"

class="demo-ruleForm"

size="small"

style="width:500px"

>

<!--礼品分类 -->

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

<el-select v-model="giftAddForm.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="giftAddForm.giftName" auto-complete="off"></el-input>

</el-form-item>

<!-- 礼品码 -->

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

<el-input type="text" v-model="giftAddForm.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="giftAddForm.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="giftAddForm.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="giftAddForm.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="giftAddForm.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="giftAddForm.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="giftAddForm.giftDesc"></el-input>

</el-form-item>

<!-- 按钮组 -->

<el-form-item>

<el-button type="success" @click="submitForm('giftAddForm')">添加</el-button>

<el-button @click="resetForm('giftAddForm')">重置</el-button>

</el-form-item>

</el-form>

</div>

</el-card>

</div>

</template>

<script>

// 引入验证函数

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

export default {

// 注入reloa依赖

inject:['reload'],

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{

// 表单数据

giftAddForm:{

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'}

],

// 所需积分验证

costScore:[

{required:true,message: '请输入所需积分', trigger: 'blur'}

]

} ,

// 商品分类

giftClassifyOptions: [

{

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'

}

],

}

},

methods:{

// 重置按钮事件

resetForm(formName) {

this.$refs[formName].resetFields();

},

// 提交按钮事件

submitForm(formName) {

this.$refs[formName].validate((valid) => {

// 如果成功

if (valid) {

// 提交数据给后端

let params = {

giftclassify:this.giftAddForm.giftClassify,

giftname: this.giftAddForm.giftName,

giftcode:this.giftAddForm.giftCode,

costscore:this.giftAddForm.costScore,

price:this.giftAddForm.price,

stocknum:this.giftAddForm.stockNum,

giftunit:this.giftAddForm.giftUnit,

issale:this.giftAddForm.isSale,

giftdesc:this.giftAddForm.giftDesc

};

//向后端发送数据

this.request.post('/gift/giftadd',params)

.then(res=>{

// 接收数据

let { code , msg } = res;

if(code===0){

// 判断是否继续添加数据

this.$confirm('添加数据成功，是否继续添加数据?', '提示', {

confirmButtonText: '确定',

cancelButtonText: '取消',

type: 'success'

}).then(() => {

// 调用刷新方法

this.reload();

}).catch(() => {

// 跳转列表页面

this.$router.push("/home/giftmanage");

})

}else if(code===1){

this.$message.error(msg);

}

})

.catch(err=>{

console.log(err);

})

} else {

return false;

}

});

},

generateGiftCode(){

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

}

}

}

</script>

<style lang="less">

@import "./GiftAdd.less";

</style>