<template>

<div class="ClientAdd">

<!-- 面板 -->

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

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

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

<span>新增客户</span>

</div>

<div class="text item">

<el-form

:model="clientAddForm"

status-icon

:rules="rules"

ref="clientAddForm"

label-width="100px"

class="demo-ruleForm"

size="small"

style="width:400px"

>

<!-- 姓名 -->

<el-form-item label="客户姓名" prop="name">

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

</el-form-item>

<!-- 性别 -->

<el-form-item label="性别" prop="sex">

<el-radio-group v-model="clientAddForm.sex">

<el-radio label="男"></el-radio>

<el-radio label="女"></el-radio>

</el-radio-group>

</el-form-item>

<!-- 电话号码 -->

<el-form-item label="电话号码" prop="tel">

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

</el-form-item>

<!-- 意向课程 -->

<el-form-item label="意向课程" prop="courseData">

<el-select v-model="clientAddForm.courseData" multiple placeholder="请选择意向课程">

<el-option

v-for="item in courseOptions"

:key="item.value"

:label="item.coursename"

:value="item.coursename">

</el-option>

</el-select>

</el-form-item>

<!-- 销售状态 -->

<el-form-item label="销售状态" prop="salesStatus">

<el-select v-model="clientAddForm.salesStatus" placeholder="请选择销售状态">

<el-option

v-for="item in salesStatusOptions"

:key="item.value"

:label="item.label"

:value="item.value">

</el-option>

</el-select>

</el-form-item>

<!-- 跟单人员 -->

<el-form-item label="跟单人员" prop="salesMan">

<el-select v-model="clientAddForm.salesMan" placeholder="请选择跟单人员">

<el-option

v-for="item in salesManOptions"

:key="item.value"

:label="item.name"

:value="item.name">

</el-option>

</el-select>

</el-form-item>

<!-- 所在学校 -->

<el-form-item label="所在学校" prop="school">

<el-select v-model="clientAddForm.school" placeholder="请选择所在学校">

<el-option label="成华区实验小学" value="成华区实验小学"></el-option>

<el-option label="华阳实验小学" value="华阳实验小学"></el-option>

<el-option label="泡桐树小学" value="泡桐树小学"></el-option>

<el-option label="龙江路小学" value="龙江路小学"></el-option>

</el-select>

</el-form-item>

<!-- 所在年级 -->

<el-form-item label="所在年级" prop="classData">

<el-select v-model="clientAddForm.classData" placeholder="请选择所在年级">

<el-option label="一年级" value="一年级"></el-option>

<el-option label="二年级" value="二年级"></el-option>

<el-option label="三年级" value="三年级"></el-option>

<el-option label="四年级" value="四年级"></el-option>

<el-option label="五年级" value="五年级"></el-option>

<el-option label="六年级" value="六年级"></el-option>

</el-select>

</el-form-item>

<!-- 是否报班 -->

<el-form-item label="是否报班" prop="courseStatus">

<el-switch

v-model="clientAddForm.courseStatus"

active-color="#13ce66"

inactive-color="#ff4949"

@change="activeSalesStatus('clientAddForm.courseStatus')"

>

</el-switch>

</el-form-item>

<!-- 按钮组 -->

<el-form-item>

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

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

</el-form-item>

</el-form>

</div>

</el-card>

</div>

</template>

<script>

// 引入验证用户名函数

import { nameReg,telReg } from "@/utils/validator/validator.js"

export default {

// 注入reloa依赖

inject:['reload'],

data(){

// 姓名验证

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

if(!value){

callback(new Error("姓名不能为空"));//输出错误提示

}else if(!nameReg(value)){

callback(new Error("请输入中文"));//输出错误提示

}else{

callback();//成功回调

}

}

// 验证电话号码

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

if(!value){

callback(new Error("电话号码不能为空"));//输出错误提示

}else if(!telReg(value)){

callback(new Error("电话号码不合法"));//输出错误提示

}else{

callback();//成功回调

}

}

// 验证销售状态，如果已报班，将是否报班改为是

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

if(!value){

callback(new Error("请选择销售状态"));//输出错误提示

}else{

if(value){

if(value==="已报班"){

this.$refs.clientAddForm.model.courseStatus=true;

}else{

this.$refs.clientAddForm.model.courseStatus=false;

}

callback();//成功回调

}

}

}

return{

// 表单数据

clientAddForm:{

name:"",

sex: "",

salesStatus:"",

courseData:[],

tel:"",

school:"",

classData:"",

courseStatus:false,

salesMan:""

},

// 验证规则

rules:{

// 姓名验证

name:[

{required:true,validator:validateName, trigger: 'blur'}

],

// 性别验证

sex:[

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

],

// 电话号码验证

tel:[

{required:true,validator: validateTel, trigger: 'blur'}

],

// 销售状态

salesStatus:[

{required:true,validator: validateSales,trigger: 'change'}

],

// 课程选择

courseData:[

{required:true,message: '请选择意向课程',trigger: 'change'}

],

// 跟单人员

salesMan:[

{required:true,message: '请选择跟单人员',trigger: 'change'}

]

} ,

// 课程选择

courseOptions: [],

// 销售状态

salesStatusOptions: [

{

value: '已报班',

label: '已报班'

}, {

value: '意向强烈',

label: '意向强烈'

}, {

value: '意向一般',

label: '意向一般'

}, {

value: '下期班联系',

label: '下期班联系'

}, {

value: '勿扰',

label: '勿扰'

}

],

// 跟单人员

salesManOptions: [

{

value: '张三',

label: '张三'

}, {

value: '李四',

label: '李四'

}, {

value: '赵王',

label: '赵王'

}

],

}

},

methods:{

// 是否报班开关触发的事件，如果值为true，则将销售状态的值改为已报班

activeSalesStatus(val){

if(val){

this.clientAddForm.salesStatus="已报班";

}else if(!val){

this.clientAddForm.salesStatus="";

}

},

// 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 = {

name:this.clientAddForm.name,

sex: this.clientAddForm.sex,

salesStatus:this.clientAddForm.salesStatus,

courseData:this.clientAddForm.courseData.join('/'),

tel:this.clientAddForm.tel,

school:this.clientAddForm.school,

classData:this.clientAddForm.classData,

courseStatus:this.clientAddForm.courseStatus,

salesMan:this.clientAddForm.salesMan

};

//向后端发送数据

this.request.post('/client/clientadd',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/salesmanage");

})

}else if(code===1){

this.$message.error(msg);

}

})

.catch(err=>{

console.log(err);

})

} else {

return false;

}

});

},

},

//创建状态获取相关信息

created(){

// 获取课程相关信息

this.request.get('/class/couresdata')

.then(res=>{

// 接收后端传回来的数据

this.courseOptions=res;

})

.catch(err=>{

console.log(err)

})

// 获得销售人员相关信息

this.request.get('/staff/staffdata')

.then(res=>{

// 接收后台传送回来的数据

let { saleman }=res;

this.salesManOptions = saleman;

})

.catch(err=>{

console.log(err);

})

}

}

</script>

<style lang="less">

@import "./ClientAdd.less";

</style>