<template>

<div class="CourseAdd">

<!-- 面板 -->

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

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

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

<span>新增课程</span>

</div>

<div class="text item">

<el-form

:model="courseAddForm"

status-icon

:rules="rules"

ref="courseAddForm"

label-width="100px"

class="demo-ruleForm"

size="small"

style="width:400px"

>

<!-- 课程名称 -->

<el-form-item label="课程名称" prop="courseName">

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

</el-form-item>

<!-- 总课时数 -->

<el-form-item label="总课时数" prop="coursePeriod">

<el-input-number

v-model="courseAddForm.coursePeriod"

:min="0"

label="描述文字">

</el-input-number>

</el-form-item>

<!-- 课时时长 -->

<el-form-item label="课时总时长" prop="CourseLength">

<el-select v-model="courseAddForm.CourseLength" placeholder="请选择总课时时长">

<el-option label="30分钟" value="30分钟"></el-option>

<el-option label="45分钟" value="45分钟"></el-option>

<el-option label="60分钟" value="60分钟"></el-option>

<el-option label="70分钟" value="70分钟"></el-option>

<el-option label="90分钟" value="90分钟"></el-option>

<el-option label="120分钟" value="120分钟"></el-option>

</el-select>

</el-form-item>

<!-- 单次课时费 -->

<el-form-item label="单次课时费" prop="courseFees" >

<el-input class="fees" type="text" v-model="courseAddForm.courseFees" auto-complete="off"></el-input>

</el-form-item>

<!-- 课程对象 -->

<el-form-item label="课程对象" prop="fitStudent">

<el-select v-model="courseAddForm.fitStudent" multiple placeholder="请选择课程适合对象">

<el-option

v-for="item in fitStudentOptions"

:key="item.value"

:label="item.label"

:value="item.value">

</el-option>

</el-select>

</el-form-item>

<!-- 启用该课程 -->

<el-form-item label="启用该课程" prop="initiateMode ">

<el-switch

v-model="courseAddForm.initiateMode"

active-color="#13ce66"

inactive-color="#ff4949"

>

</el-switch>

</el-form-item>

<!-- 课程介绍 -->

<el-form-item label="课程介绍" prop="courseDesc">

<el-input type="textarea" v-model="courseAddForm.courseDesc"></el-input>

</el-form-item>

<!-- 按钮组 -->

<el-form-item>

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

<el-button @click="resetForm('courseAddForm')">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 { feesReg } from "@/utils/validator/validator.js"

export default {

// 注入reloa依赖

inject:['reload'],

data(){

// 课时费用验证

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

if(!value){

callback(new Error("课时费用不能为空"));//输出错误提示

}else if(!feesReg(value)){

callback(new Error("课时费用必须为大于0的正整数"));//输出错误提示

}else{

callback();//成功回调

}

}

return{

// 表单数据

courseAddForm:{

courseName:"",

coursePeriod: "",

courseFees:"",

CourseLength:"",

fitStudent:[],

courseDesc:"",

initiateMode:false

},

// 验证规则

rules:{

// 课程名称验证

courseName:[

{required:true,message: '请输入课程名称', trigger: 'blur'}

],

// 课时数验证

coursePeriod:[

{required:true,message: '请选择课时数',trigger: 'change'}

],

// 课时费用验证

courseFees:[

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

],

// 课时长度验证

CourseLength:[

{required:true,message: '请选择课时长度',trigger: 'change'}

],

// 课程适合学生对象验证

fitStudent:[

{required:true,message: '请选择课程适合对象',trigger: 'change'}

],

// 课程描述验证

courseDesc:[

{required:true,message: '请输入课程描述',trigger: 'blur'}

]

} ,

// 课程选择

fitStudentOptions: [{

value: '小学一年级',

label: '小学一年级'

}, {

value: '小学二年级',

label: '小学二年级'

}, {

value: '小学三年级',

label: '小学三年级'

}, {

value: '小学四年级',

label: '小学四年级'

}, {

value: '小学五年级',

label: '小学五年级'

}, {

value: '小学六年级',

label: '小学六年级'

}],

}

},

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 = {

courseName:this.courseAddForm.courseName,

coursePeriod: this.courseAddForm.coursePeriod+"课时",

courseFees:this.courseAddForm.courseFees+"元/课时",

CourseLength:this.courseAddForm.CourseLength,

fitStudent:this.courseAddForm.fitStudent.join('/'),

courseDesc:this.courseAddForm.courseDesc,

initiateMode:this.courseAddForm.initiateMode

};

// 向后台发送数据

this.request.post('/course/courseadd',params)

.then(res=>{

// 接收传过来的参数

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/coursemanage");

})

}else{

this.$message.error(msg);

}

})

.catch(err=>{

console.log(err);

})

} else {

return false;

}

});

},

}

}

</script>

<style lang="less">

@import "./CourseAdd.less";

</style>