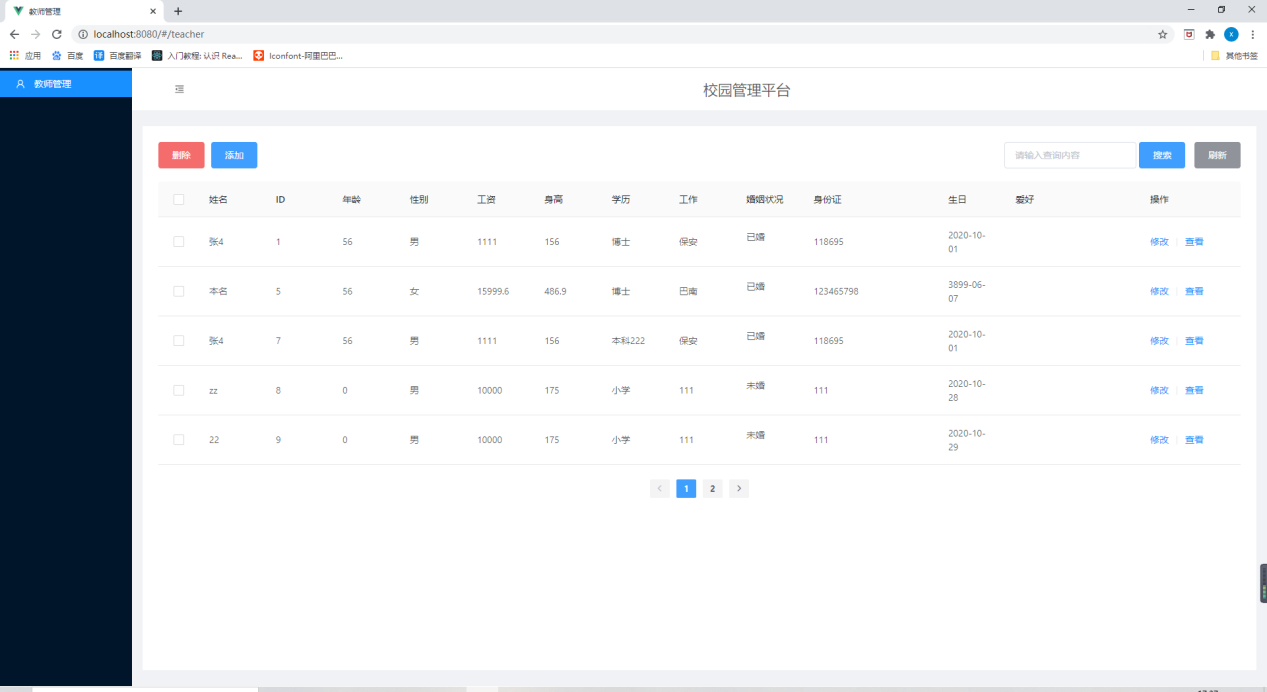
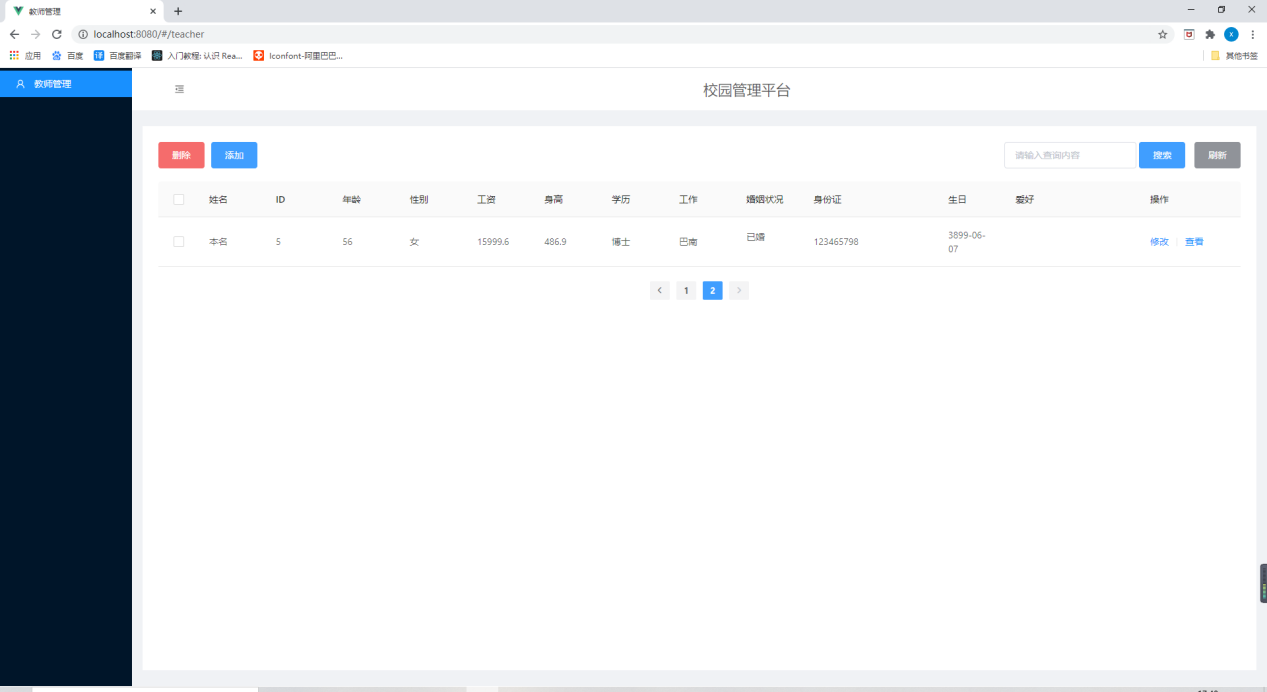
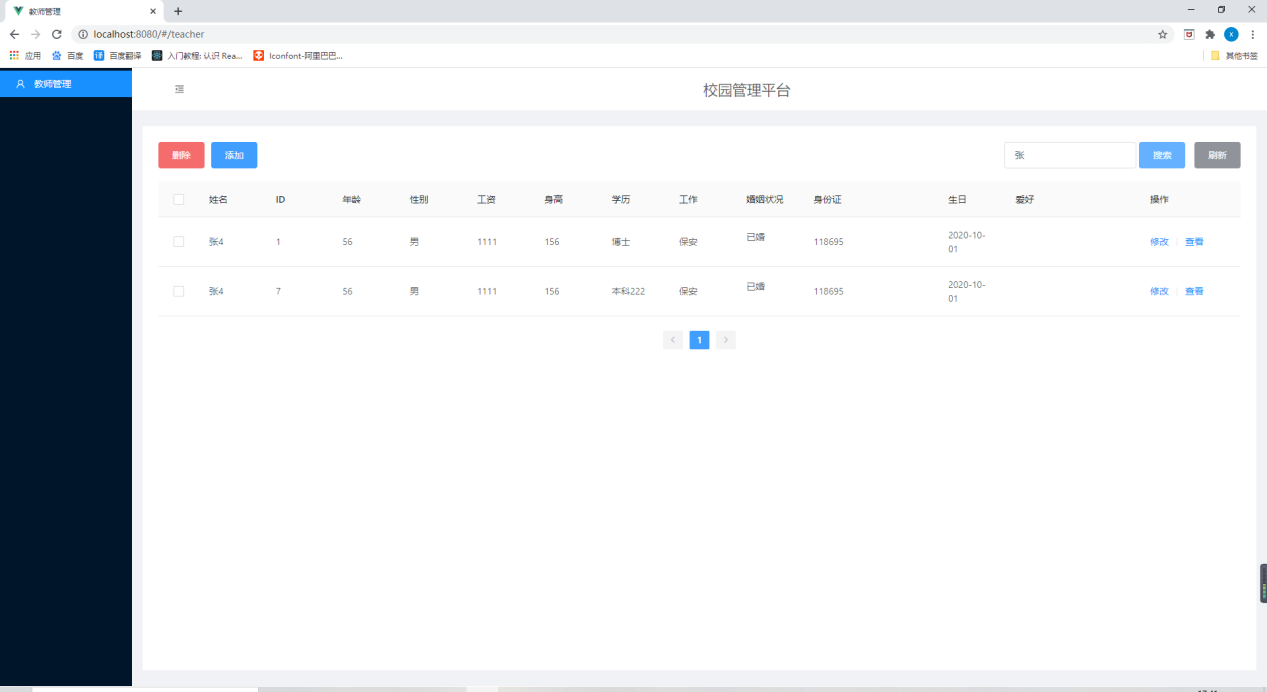
分页





查询





<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.tttt.student.dao.TeacherMapper">

<resultMap id="BaseResultMap" type="com.tttt.student.pojo.Teacher" autoMapping="true">

<result column="sex" property="sex" typeHandler="com.tttt.student.util.SexTypeHandler"/>

<result column="id" property="id"/>

<collection property="hobbies" ofType="com.tttt.student.pojo.Hobby" column="id" select="com.tttt.student.dao.HobbyMapper.selectByUserId"/>

</resultMap>

<sql id="Base\_Column\_List">

id, age, birthday, major , department , college, code, degree, height, is\_marry as isMarry, job, name,

place\_of\_residence as placeOfResidence, salary, sex, weight

</sql>

<delete id="deleteTeacherById" parameterType="java.lang.Long">

delete from teacher

where id = #{id,jdbcType=BIGINT}

</delete>

<insert id="addTeacher" parameterType="com.tttt.student.pojo.Teacher" keyProperty="id" useGeneratedKeys="true">

insert into teacher (age, birthday,

code, degree,

height, is\_marry, job, name,

place\_of\_residence,

salary, sex, weight, major , department , college)

values (#{age,jdbcType=INTEGER}, #{birthday,jdbcType=DATE},

#{code,jdbcType=VARCHAR}, #{degree,jdbcType=VARCHAR},

#{height,jdbcType=DOUBLE}, #{isMarry,jdbcType=BIT}, #{job,jdbcType=VARCHAR},

#{name,jdbcType=VARCHAR},#{placeOfResidence,jdbcType=VARCHAR},

#{salary,jdbcType=DOUBLE}, #{sex,typeHandler=com.unixstar.student.tool.SexTypeHandler,jdbcType=CHAR},

#{weight,jdbcType=DOUBLE},#{major},#{department},#{college})

</insert>

<select id="findAllTeacher" resultMap="BaseResultMap">

select <include refid="Base\_Column\_List"/> from teacher order by code

</select>

<select id="findTeachersByName" resultMap="BaseResultMap">

<bind name="namePattern" value="'%'+name+'%'"/>

select <include refid="Base\_Column\_List"/> from teacher where name like #{namePattern}

</select>

<update id="updateTeacher" parameterType="com.tttt.student.pojo.Teacher">

update teacher

set age = #{age,jdbcType=INTEGER},

birthday = #{birthday,jdbcType=DATE},

code = #{code,jdbcType=VARCHAR},

degree = #{degree,jdbcType=VARCHAR},

height = #{height,jdbcType=DOUBLE},

is\_marry = #{isMarry,jdbcType=BIT},

job = #{job,jdbcType=VARCHAR},

name = #{name,jdbcType=VARCHAR},

place\_of\_residence = #{placeOfResidence,jdbcType=VARCHAR},

salary = #{salary,jdbcType=DOUBLE},

sex = #{sex,jdbcType=CHAR,typeHandler=com.unixstar.student.tool.SexTypeHandler},

weight = #{weight,jdbcType=DOUBLE}

major=#{record.major},

department=#{record.department},

college=#{record.college}

where id = #{id,jdbcType=BIGINT}

</update>

</mapper>

<template>

<div>

<div style="display: flex;justify-content: space-between;margin-bottom: 20px">

<div style="display: flex">

<el-button type="danger" size="large" @click="()=>{delModal=!delModal}">删除</el-button>

<el-button type="primary" size="large" @click="()=>{addModal=!addModal}">添加</el-button>

</div>

<div>

<el-input v-model="searchInput" maxlength='10' placeholder="请输入查询内容" style="width: 200px" />

<el-button type="primary" size="large" @click="search()">搜索</el-button>

<el-button type="info" size="large" @click="refresh()">刷新</el-button>

</div>

</div>

<div style="width: 100%;">

<a-table :columns="columns" :data-source="list" :row-selection="rowSelection" rowKey='id' :pagination='false'>

<span slot="isMarry" slot-scope="isMarry">

<p v-if="isMarry">已婚</p>

<p v-else>未婚</p>

</span>

<span slot="hobby" slot-scope="hobby">

<a-tag

v-for="tag,index in hobby"

:key="index">

{{ tag.toUpperCase() }}

</a-tag>

</span>

<span slot="action" slot-scope="text, record">

<a @click="update(record.id)">修改</a>

<a-divider type="vertical"/>

<a @click="check(record.id)">查看</a>

</span>

</a-table>

</div>

<div style="display:flex;justify-content: center;margin-top: 20px">

<el-pagination

background

@current-change="handleCurrentChange"

:page-size=pageSize

layout="prev, pager, next"

:total=total>

</el-pagination>

</div>

<!-- 确认删除-->

<el-dialog

title="确认删除"

:visible.sync="delModal"

>

<p>是否确定删除?</p>

<p style="font-size: 18px;color:red">数据删除后无法恢复</p>

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

<el-button @click="delModal = false">取 消</el-button>

<el-button type="primary" @click="delTeacher">确 定</el-button>

</span>

</el-dialog>

<!-- 添加-->

<el-dialog

title="添加人员"

:visible.sync="addModal"

>

<el-form ref="form" :model="addForm" label-width="80px">

<el-form-item label="姓名">

<el-input v-model="addForm.name"></el-input>

</el-form-item>

<el-form-item label="性别">

<el-select v-model="addForm.sex" placeholder="请选择性别">

<el-option label="男" value="男"></el-option>

<el-option label="女" value="女"></el-option>

</el-select>

</el-form-item>

<el-form-item label="年龄">

<el-input v-model="addForm.age"></el-input>

</el-form-item>

<el-form-item label="工资">

<el-input v-model="addForm.salary"></el-input>

</el-form-item>

<el-form-item label="学历">

<div class="block">

<el-cascader

:options="options"

@change="cascader"></el-cascader>

</div>

</el-form-item>

<el-form-item label="婚姻状况">

<el-switch

v-model="addForm.isMarry"

active-color="#13ce66"

inactive-color="#ff4949">

</el-switch>

</el-form-item>

<el-form-item label="体重">

<el-input v-model="addForm.weight"></el-input>

</el-form-item>

<el-form-item label="身高">

<el-input v-model="addForm.height"></el-input>

</el-form-item>

<el-form-item label="身份证">

<el-input v-model="addForm.code"></el-input>

</el-form-item>

<el-form-item label="工作">

<el-input v-model="addForm.job"></el-input>

</el-form-item>

<el-form-item label="生日">

<el-date-picker

v-model="addForm.birthday"

type="date"

placeholder="选择日期">

</el-date-picker>

</el-form-item>

<el-form-item label="爱好">

<el-tag

:key="index"

v-for="tag,index in addForm.hobby"

closable

:disable-transitions="false"

@close="handleClose(tag)">

{{tag}}

</el-tag>

<el-input

class="input-new-tag"

v-if="inputVisible"

v-model="inputValue"

ref="saveTagInput"

size="small"

@keyup.enter.native="handleInputConfirm"

@blur="handleInputConfirm"

>

</el-input>

<el-button v-else class="button-new-tag" size="small" @click="showInput">+ New Tag</el-button>

</el-form-item>

<el-form-item>

<el-button type="primary" @click="addTeacher">立即创建</el-button>

<el-button @click="()=>{addModal=false}">取消</el-button>

</el-form-item>

</el-form>

</el-dialog>

<!-- 查看-->

<el-dialog

title="查看信息"

:visible.sync="checkModal">

<el-form ref="form" :model="checkForm" label-width="80px">

<el-form-item label="姓名">

<p>{{checkForm.name}}</p>

</el-form-item>

<el-form-item label="性别">

<p>{{checkForm.sex}}</p>

</el-form-item>

<el-form-item label="年龄">

<p>{{checkForm.age}}</p>

</el-form-item>

<el-form-item label="工资">

<p>{{checkForm.salary}}</p>

</el-form-item>

<el-form-item label="学历">

<p>{{checkForm.degree}}</p>

</el-form-item>

<el-form-item label="婚姻状况">

<p v-if="checkForm.isMarry">已婚</p>

<p v-else>未婚</p>

</el-form-item>

<el-form-item label="体重">

<p>{{checkForm.weight}}</p>

</el-form-item>

<el-form-item label="身高">

<p>{{checkForm.height}}</p>

</el-form-item>

<el-form-item label="身份证">

<p>{{checkForm.code}}</p>

</el-form-item>

<el-form-item label="工作">

<p>{{checkForm.job}}</p>

</el-form-item>

<el-form-item label="生日">

<p>{{checkForm.birthday}}</p>

</el-form-item>

<el-form-item label="爱好">

<el-tag

:key="index"

v-for="tag,index in checkForm.hobby"

:disable-transitions="false"

@close="handleClose(tag)">

{{tag}}

</el-tag>

</el-form-item>

<el-form-item>

<el-button @click="()=>{checkModal=false}">取消</el-button>

</el-form-item>

</el-form>

</el-dialog>

<!-- 修改-->

<el-dialog

title="修改信息"

:visible.sync="updateModal"

>

<el-form ref="form" :model="updateForm" label-width="80px">

<el-form-item label="姓名">

<el-input v-model="updateForm.name"></el-input>

</el-form-item>

<el-form-item label="性别">

<el-select v-model="updateForm.sex" placeholder="请选择性别">

<el-option label="男" value="男"></el-option>

<el-option label="女" value="女"></el-option>

</el-select>

</el-form-item>

<el-form-item label="年龄">

<el-input v-model="updateForm.age"></el-input>

</el-form-item>

<el-form-item label="工资">

<el-input v-model="updateForm.salary"></el-input>

</el-form-item>

<!-- <el-form-item label="学历">-->

<!-- <div class="block">-->

<!-- <el-cascader-->

<!-- :options="options"-->

<!-- @change="cascader"></el-cascader>-->

<!-- </div>-->

<!-- </el-form-item>-->

<el-form-item label="婚姻状况">

<el-switch

v-model="updateForm.isMarry"

active-color="#13ce66"

inactive-color="#ff4949">

</el-switch>

</el-form-item>

<el-form-item label="体重">

<el-input v-model="updateForm.weight"></el-input>

</el-form-item>

<el-form-item label="身高">

<el-input v-model="updateForm.height"></el-input>

</el-form-item>

<el-form-item label="身份证">

<el-input v-model="updateForm.code"></el-input>

</el-form-item>

<el-form-item label="工作">

<el-input v-model="updateForm.job"></el-input>

</el-form-item>

<el-form-item label="生日">

<el-date-picker

v-model="updateForm.birthday"

type="date"

placeholder="选择日期">

</el-date-picker>

</el-form-item>

<el-form-item label="爱好">

<el-tag

:key="index"

v-for="tag,index in updateForm.hobby"

closable

:disable-transitions="false"

@close="handleClose(tag)">

{{tag}}

</el-tag>

<el-input

class="input-new-tag"

v-if="inputVisible"

v-model="inputValue"

ref="saveTagInput"

size="small"

@keyup.enter.native="handleInputConfirmUpdate"

@blur="handleInputConfirmUpdate"

>

</el-input>

<el-button v-else class="button-new-tag" size="small" @click="showInput">+ New Tag</el-button>

</el-form-item>

<el-form-item>

<el-button type="primary" @click="updateTeacher">修改</el-button>

<el-button @click="()=>{updateModal=false}">取消</el-button>

</el-form-item>

</el-form>

</el-dialog>

</div>

</template>

<script>

import {axios\_get, axios\_post} from "../network/request";

import HelloWorld from "../components/HelloWorld";

export default {

name: "home",

components: {HelloWorld},

data() {

return {

query:[],

total:0,

pageSize:5,

searchInput:'',

updateModal: false,

updateForm: {},

checkForm: {},

addForm: {

name: '',

id: '',

age: 0,

sex: '',

birthday: '',

code: '',

height: 175,

weight: 70,

hobby: [],

isMarry: false,

job: '',

degree: '',

salary: 10000,

},

addModal: false,

delModal: false,

checkModal: false,

list: [

{

name: 'zx',

id: '1',

age: 35,

sex: '男',

birthday: '9.26',

code: '500225',

height: 175,

weight: 70,

hobby: ['编程', '运动'],

isMarry: true,

job: '老师',

degree: '教授',

salary: 10000,

}

],

columns: [

{

title: '姓名',

dataIndex: 'name',

key: 'name',

width: 100

},

{

title: 'ID',

dataIndex: 'id',

key: 'id',

width: 100

},

{

title: '年龄',

dataIndex: 'age',

key: 'age',

width: 100

},

{

title: '性别',

dataIndex: 'sex',

key: 'sex',

width: 100

},

{

title: '工资',

dataIndex: 'salary',

key: 'salary',

width: 100

},

// {

// title: '体重',

// dataIndex: 'weight',

// key: 'weight',

// width: 100

// },

{

title: '身高',

dataIndex: 'height',

key: 'height',

width: 100

},

{

title: '学历',

dataIndex: 'degree',

key: 'degree',

width: 100

},

{

title: '工作',

dataIndex: 'job',

key: 'job',

width: 100

},

{

title: '婚姻状况',

dataIndex: 'isMarry',

key: 'isMarry',

width: 100,

scopedSlots: {customRender: 'isMarry'},

},

{

title: '身份证',

dataIndex: 'code',

key: 'code',

width: 200

},

{

title: '生日',

dataIndex: 'birthday',

key: 'birthday',

width: 100

},

{

title: '爱好',

dataIndex: 'hobby',

key: 'hobby',

width: 200,

scopedSlots: {customRender: 'hobby'},

},

{

title: '操作',

key: 'action',

scopedSlots: {customRender: 'action'},

width: 150

},

],

selected: [],

rowSelection: {

onChange: (selectedRowKeys, selectedRows) => {

console.log(`selectedRowKeys: ${selectedRowKeys}`, 'selectedRows: ', selectedRows);

},

onSelect: (record, selected, selectedRows) => {

this.selected = selectedRows;

console.log(record, selected, selectedRows);

},

onSelectAll: (selected, selectedRows, changeRows) => {

this.selected = selectedRows;

console.log(selected, selectedRows, changeRows);

},

},

options: [

{

value: '初等教育',

label: '初等教育',

children: [

{

value: '小学',

label: '小学',

},

]

},

{

value: '中等教育',

label: '中等教育',

children: [

{

value: '初中',

label: '初中',

},

{

value: '高中',

label: '高中',

},

{

value: '技校',

label: '技校',

},

]

},

{

value: '高等教育',

label: '高等教育',

children: [

{

value: '专科',

label: '专科',

},

{

value: '本科',

label: '本科',

},

{

value: '研究生',

label: '研究生',

},

]

},

],

inputVisible: false,

inputValue: '',

}

},

created() {

this.refresh()

},

methods: {

refresh(){

this.query=[];

axios\_get('teacher/getAll').then((data) => {

this.total=data.length;

if (data.length>this.pageSize){

this.list=data.slice(0,this.pageSize)

}else{

this.list=data

}

}).catch(() => {

this.$message.error('Request data failed');

})

},

// 分页

handleCurrentChange(val){

if (this.query.length!==0){

this.list=this.query.slice(this.pageSize\*(val-1),this.pageSize\*val)

}else{

axios\_get('teacher/getCodeSortAndPage',{pageNum:val,pageSize:this.pageSize})

.then((data)=>{

this.list=data

})

}

},

//查询

search(){

let reg=/[A-z]|[0-9]|[\u4e00-\u9fa5]/g;

if (reg.test(this.searchInput)){

axios\_get('teacher/query',{name:this.searchInput})

.then((value)=>{

this.list=[];

if (this.pageSize>value.length){

this.list=value.slice(0,this.pageSize)

this.query=value

}else{

this.list=value

}

this.total=value.length

})

}else{

console.log('无效查询')

}

},

update(id) {

let t = {};

for (t of this.list) {

if (t.id === id) {

this.updateForm = t;

this.updateModal = true

}

}

},

updateTeacher() {

this.updateModal = false

axios\_post('teacher/update', this.updateForm)

.then(() => {

for (let t of this.list) {

if (t.id === this.updateForm.id) {

this.refresh()

}

}

})

.catch(() => {

})

},

check(id) {

let t = {};

for (t of this.list) {

if (t.id === id) {

this.checkForm = t;

this.checkModal = true

}

}

},

handleClose(tag) {

this.dynamicTags.splice(this.dynamicTags.indexOf(tag), 1);

},

showInput() {

this.inputVisible = true;

this.$nextTick(\_ => {

this.$refs.saveTagInput.$refs.input.focus();

});

},

handleInputConfirm() {

let inputValue = this.inputValue;

if (inputValue) {

this.addForm.hobby.push(inputValue);

}

this.inputVisible = false;

this.inputValue = '';

},

handleInputConfirmUpdate() {

let inputValue = this.inputValue;

if (inputValue) {

this.updateForm.hobby.push(inputValue);

}

this.inputVisible = false;

this.inputValue = '';

},

cascader(value) {

this.addForm.degree = value[value.length - 1]

},

delTeacher() {

this.delModal = false

for (let s of this.selected) {

console.log('删除', s)

axios\_get('teacher/remove', {id: s.id})

.then(() => {

this.$message.info('Delete Success');

for (let t in this.list) {

if (this.list[t].id === s.id) {

this.refresh()

}

}

})

.catch(() => {

this.$message.error('Delete failed');

})

}

},

dateformat(fmt, date) {

let ret;

const opt = {

"y+": date.getFullYear().toString(), // 年

"m+": (date.getMonth() + 1).toString(), // 月

"d+": date.getDate().toString(), // 日

"H+": date.getHours().toString(), // 时

"M+": date.getMinutes().toString(), // 分

"S+": date.getSeconds().toString() // 秒

// 有其他格式化字符需求可以继续添加，必须转化成字符串

};

for (let k in opt) {

ret = new RegExp("(" + k + ")").exec(fmt);

if (ret) {

fmt = fmt.replace(ret[1], (ret[1].length === 1) ? (opt[k]) : (opt[k].padStart(ret[1].length, "0")))

}

}

return fmt;

},

addTeacher() {

this.addModal = false;

this.addForm.birthday = this.dateformat('yyyy-mm-dd', this.addForm.birthday);

console.log(this.addForm.birthday)

axios\_post('teacher/add', this.addForm)

.then((data) => {

console.log(data)

this.refresh()

})

.catch(() => {

this.$message.error('Add failed');

})

},

},

}

</script>

<style scoped>

</style>