目录

[运行截图 1](#_Toc52120075)

[添加教师 2](#_Toc52120076)

[修改教师 4](#_Toc52120077)

[删除教师 5](#_Toc52120078)

[代码 6](#_Toc52120079)

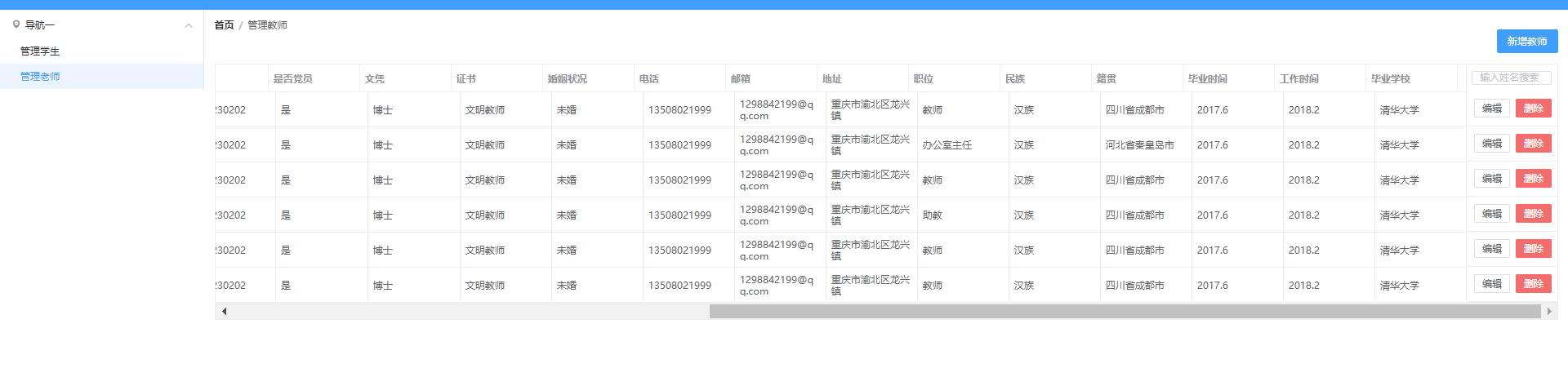
[Teacher 6](#_Toc52120080)

[StroeTeacher 12](#_Toc52120081)

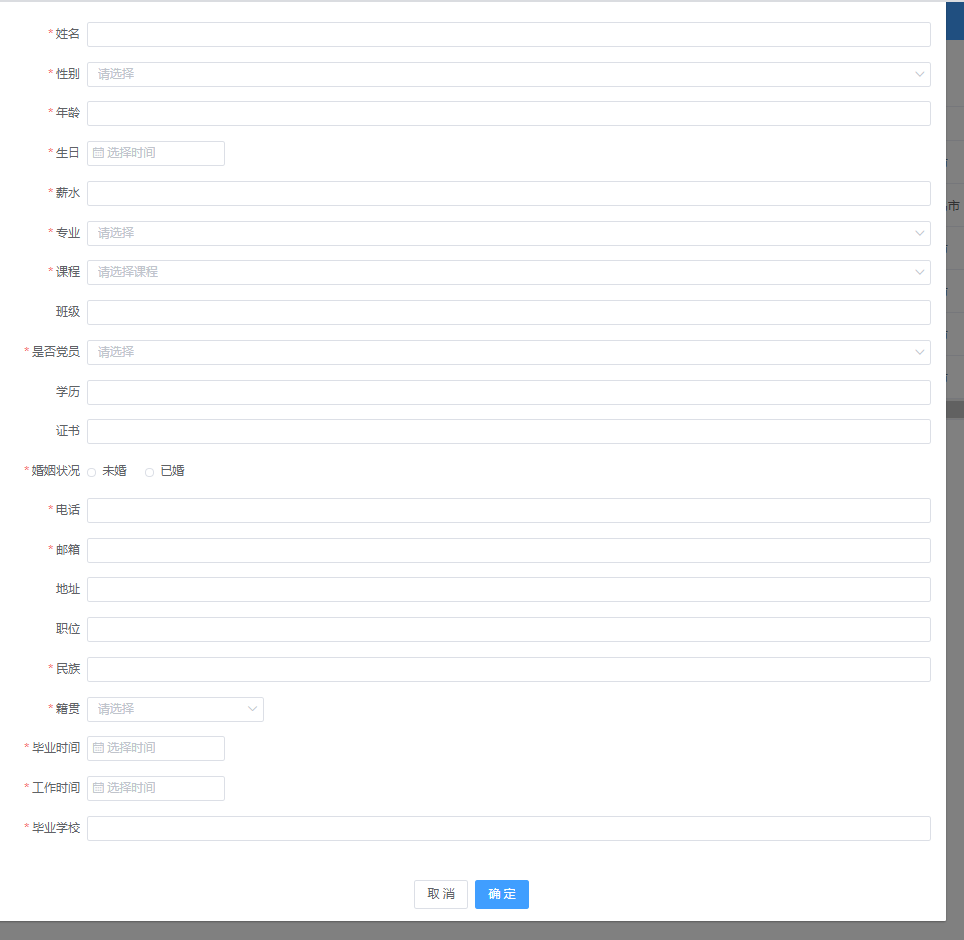
[Teacher.vue 14](#_Toc52120082)

# 运行截图

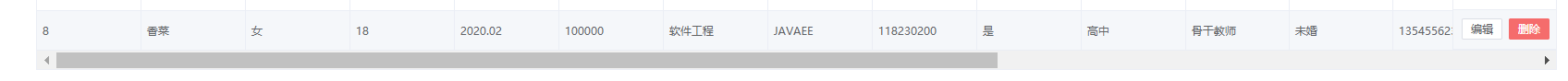




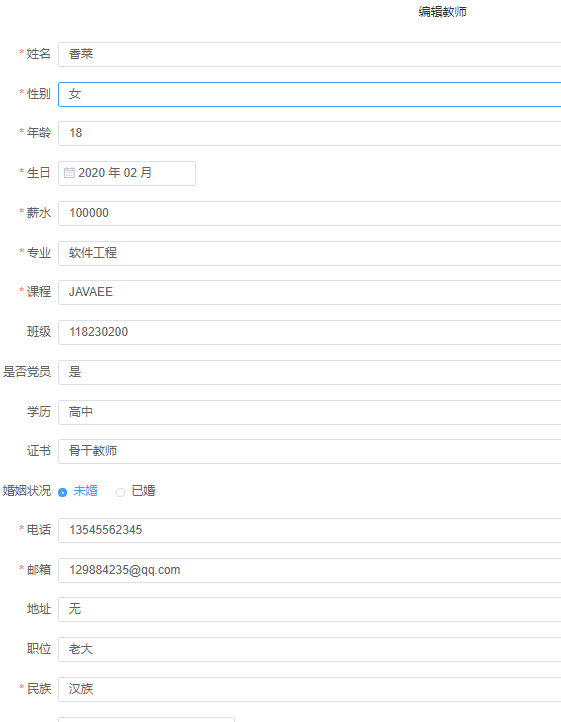
## 添加教师







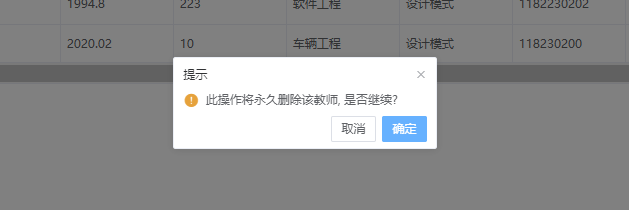
## 修改教师

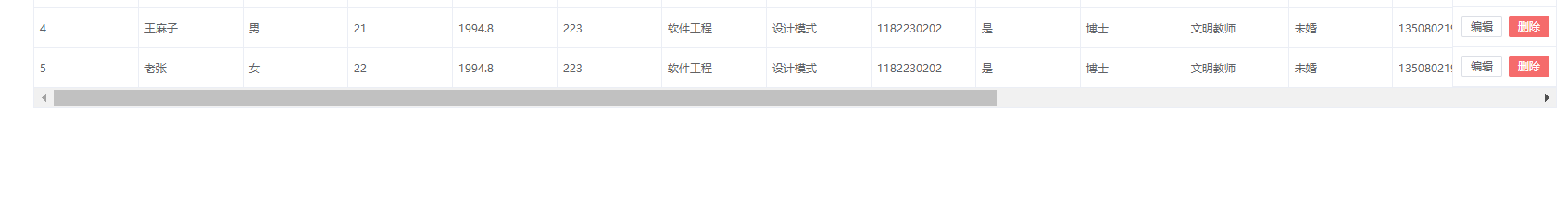






## 删除教师





# 代码

## Teacher

package com.example.second.entity;  
  
import java.util.Date;  
import java.util.List;  
  
public class Teacher {  
 private Long uid;  
 private String name;  
 private String sex;  
 private Integer age;  
 private String birth;  
 private Integer salary;  
 private String major;  
 private String course;  
 private String clazz;  
 private String partyMember;  
 private String diploma;  
 private String certificate;  
 private String maritalStatus;  
 private String phoneNumber;  
 private String email;  
 private String address;  
 private String position;  
 private String ethnic;  
 private String[] nativePlace;  
 private String yearOfGraduation;  
 private String yearOfWork;  
 private String schoolOfGraduation;  
*// private String[] hobby;* public Teacher() {  
  
 }  
  
 public Teacher(Long uid, String name, String sex, Integer age, String birth, Integer salary, String major, String course, String clazz, String partyMember, String diploma, String certificate, String maritalStatus, String phoneNumber, String email, String address, String position, String ethnic, String[] nativePlace, String yearOfGraduation, String yearOfWork, String schoolOfGraduation) {  
 this.uid = uid;  
 this.name = name;  
 this.sex = sex;  
 this.age = age;  
 this.birth = birth;  
 this.salary = salary;  
 this.major = major;  
 this.course = course;  
 this.clazz = clazz;  
 this.partyMember = partyMember;  
 this.diploma = diploma;  
 this.certificate = certificate;  
 this.maritalStatus = maritalStatus;  
 this.phoneNumber = phoneNumber;  
 this.email = email;  
 this.address = address;  
 this.position = position;  
 this.ethnic = ethnic;  
 this.nativePlace = nativePlace;  
 this.yearOfGraduation = yearOfGraduation;  
 this.yearOfWork = yearOfWork;  
 this.schoolOfGraduation = schoolOfGraduation;  
*// this.hobby = hobby;* }  
  
 public Long getUid() {  
 return uid;  
 }  
  
 public void setUid(Long uid) {  
 this.uid = uid;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getSex() {  
 return sex;  
 }  
  
 public void setSex(String sex) {  
 this.sex = sex;  
 }  
  
 public String getBirth() {  
 return birth;  
 }  
  
 public void setBirth(String birth) {  
 this.birth = birth;  
 }  
  
 public Integer getSalary() {  
 return salary;  
 }  
  
 public void setSalary(Integer salary) {  
 this.salary = salary;  
 }  
  
 public String getMajor() {  
 return major;  
 }  
  
 public void setMajor(String major) {  
 this.major = major;  
 }  
  
 public String getCourse() {  
 return course;  
 }  
  
 public void setCourse(String course) {  
 this.course = course;  
 }  
  
 public String getClazz() {  
 return clazz;  
 }  
  
 public void setClazz(String clazz) {  
 this.clazz = clazz;  
 }  
  
 public String getPartyMember() {  
 return partyMember;  
 }  
  
 public void setPartyMember(String partyMember) {  
 this.partyMember = partyMember;  
 }  
  
 public String getDiploma() {  
 return diploma;  
 }  
  
 public void setDiploma(String diploma) {  
 this.diploma = diploma;  
 }  
  
 public String getCertificate() {  
 return certificate;  
 }  
  
 public void setCertificate(String certificate) {  
 this.certificate = certificate;  
 }  
  
 public String getMaritalStatus() {  
 return maritalStatus;  
 }  
  
 public void setMaritalStatus(String maritalStatus) {  
 this.maritalStatus = maritalStatus;  
 }  
  
 public String getPhoneNumber() {  
 return phoneNumber;  
 }  
  
 public void setPhoneNumber(String phoneNumber) {  
 this.phoneNumber = phoneNumber;  
 }  
  
 public String getEmail() {  
 return email;  
 }  
  
 public void setEmail(String email) {  
 this.email = email;  
 }  
  
 public String getAddress() {  
 return address;  
 }  
  
 public void setAddress(String address) {  
 this.address = address;  
 }  
  
 public String getPosition() {  
 return position;  
 }  
  
 public void setPosition(String position) {  
 this.position = position;  
 }  
  
 public String getEthnic() {  
 return ethnic;  
 }  
  
 public void setEthnic(String ethnic) {  
 this.ethnic = ethnic;  
 }  
  
 public String[] getNativePlace() {  
 return nativePlace;  
 }  
  
 public void setNativePlace(String[] nativePlace) {  
 this.nativePlace = nativePlace;  
 }  
  
 public String getYearOfGraduation() {  
 return yearOfGraduation;  
 }  
  
 public void setYearOfGraduation(String yearOfGraduation) {  
 this.yearOfGraduation = yearOfGraduation;  
 }  
  
 public String getYearOfWork() {  
 return yearOfWork;  
 }  
  
 public void setYearOfWork(String yearOfWork) {  
 this.yearOfWork = yearOfWork;  
 }  
  
 public String getSchoolOfGraduation() {  
 return schoolOfGraduation;  
 }  
  
 public void setSchoolOfGraduation(String schoolOfGraduation) {  
 this.schoolOfGraduation = schoolOfGraduation;  
 }  
  
 public Integer getAge() {  
 return age;  
 }  
  
 public void setAge(Integer age) {  
 this.age = age;  
 }  
  
*// public String[] getHobby() {  
// return hobby;  
// }  
//  
// public void setHobby(String[] hobby) {  
// this.hobby = hobby;  
// }*}

## StroeTeacher

package com.example.second.store;  
import com.example.second.entity.Teacher;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class StoreTeacher {  
 private static List<Teacher> *teachers* = new ArrayList<Teacher>();  
 private static Long *uid* = 6L;  
  
 static {  
 Teacher stu1 = new Teacher(0L, "小王", "男", 21,"1994.8",223,"软件工程","设计模式","1182230202","是","博士","文明教师","未婚","13508021999","1298842199@qq.com","重庆市渝北区龙兴镇","教师","汉族",new String[]{"四川省","成都市"},"2017.6","2018.2","清华大学");  
 Teacher stu2 = new Teacher(1L, "小李", "女", 22,"1994.8",223,"软件工程","设计模式","1182230202","是","博士","文明教师","未婚","13508021999","1298842199@qq.com","重庆市渝北区龙兴镇","办公室主任","汉族",new String[]{"四川省","成都市"},"2017.6","2018.2","清华大学");  
 Teacher stu3 = new Teacher(2L, "张三", "男", 21,"1994.8",223,"软件工程","设计模式","1182230202","是","博士","文明教师","未婚","13508021999","1298842199@qq.com","重庆市渝北区龙兴镇","教师","汉族",new String[]{"四川省","成都市"},"2017.6","2018.2","清华大学");  
 Teacher stu4 = new Teacher(3L, "李四", "女", 22,"1994.8",223,"软件工程","设计模式","1182230202","是","博士","文明教师","未婚","13508021999","1298842199@qq.com","重庆市渝北区龙兴镇","助教","汉族",new String[]{"四川省","成都市"},"2017.6","2018.2","清华大学");  
 Teacher stu5 = new Teacher(4L, "王麻子", "男", 21,"1994.8",223,"软件工程","设计模式","1182230202","是","博士","文明教师","未婚","13508021999","1298842199@qq.com","重庆市渝北区龙兴镇","教师","汉族",new String[]{"四川省","成都市"},"2017.6","2018.2","清华大学");  
 Teacher stu6 = new Teacher(5L, "老张", "女", 22,"1994.8",223,"软件工程","设计模式","1182230202","是","博士","文明教师","未婚","13508021999","1298842199@qq.com","重庆市渝北区龙兴镇","教师","汉族",new String[]{"四川省","成都市"},"2017.6","2018.2","清华大学");  
 *teachers*.add(stu1);  
 *teachers*.add(stu2);  
 *teachers*.add(stu3);  
 *teachers*.add(stu4);  
 *teachers*.add(stu5);  
 *teachers*.add(stu6);  
 }  
  
 public static List<Teacher> findAllTeachers() {  
 return *teachers*;  
 }  
  
 public static void addTeacher(Teacher teacher) {  
 teacher.setUid(*uid*++);  
 *teachers*.add(teacher);  
 }  
  
 public static void deleteTeacherById(Long uid) {  
 for (Teacher teacher : *teachers*) {  
 if (teacher.getUid().equals(uid)) {  
 *teachers*.remove(teacher);  
 return;  
 }  
 }  
 }  
  
 public static void editTeacherById(Teacher teacher) {  
 for (int i = 0; i < *teachers*.size(); i++) {  
 if (*teachers*.get(i).getUid() == teacher.getUid()) {  
 *teachers*.set(i, teacher);  
 return;  
 }  
 }  
 }  
}

## Teacher.vue

<template>  
 <div>  
 <div>  
 <el-button type="primary" style="float:right;margin-bottom: 20px;" @click="showDialog">新增教师</el-button>  
 </div>  
 <div>  
 <el-table  
 class="table"  
 :data="teachers.filter(data => !search || data.name.toLowerCase().includes(search.toLowerCase()))"  
 border  
 style="width: 100%;">  
 <el-table-column  
  
 prop="uid"  
 label="学号"  
 width="180">  
 </el-table-column>  
 <el-table-column  
 prop="name"  
 label="姓名"  
 width="180">  
 </el-table-column>  
 <el-table-column  
 prop="sex"  
 label="性别"  
 width="180">  
 </el-table-column>  
 <el-table-column  
 prop="age"  
 label="年龄"  
 width="180">  
 </el-table-column>  
 <el-table-column  
 prop="birth"  
 label="生日"  
 width="180">  
 </el-table-column>  
 <el-table-column  
 prop="salary"  
 label="薪水"  
 width="180">  
 </el-table-column>  
 <el-table-column  
 prop="major"  
 label="专业"  
 width="180">  
 </el-table-column>  
 <el-table-column  
 prop="course"  
 label="课程"  
 width="180">  
 </el-table-column>  
 <el-table-column  
 prop="clazz"  
 label="班级"  
 width="180">  
 </el-table-column>  
 <el-table-column  
 prop="partyMember"  
 label="是否党员"  
 width="180">  
 </el-table-column>  
 <el-table-column  
 prop="diploma"  
 label="文凭"  
 width="180">  
 </el-table-column>  
 <el-table-column  
 prop="certificate"  
 label="证书"  
 width="180">  
 </el-table-column>  
 <el-table-column  
 prop="maritalStatus"  
 label="婚姻状况"  
 width="180">  
 </el-table-column>  
 <el-table-column  
 prop="phoneNumber"  
 label="电话"  
 width="180">  
 </el-table-column>  
 <el-table-column  
 prop="email"  
 label="邮箱"  
 width="180">  
 </el-table-column>  
 <el-table-column  
 prop="address"  
 label="地址"  
 width="180">  
 </el-table-column>  
 <el-table-column  
 prop="position"  
 label="职位"  
 width="180">  
 </el-table-column>  
 <el-table-column  
 prop="ethnic"  
 label="民族"  
 width="180">  
 </el-table-column>  
 <el-table-column  
 prop="nativePlace"  
 label="籍贯"  
 width="180">  
 </el-table-column>  
 <el-table-column  
 prop="yearOfGraduation"  
 label="毕业时间"  
 width="180">  
 </el-table-column>  
 <el-table-column  
 prop="yearOfWork"  
 label="工作时间"  
 width="180">  
 </el-table-column>  
 <el-table-column  
 prop="schoolOfGraduation"  
 label="毕业学校"  
 width="180">  
 </el-table-column>  
*<!-- <el-table-column-->  
<!-- prop="hobby"-->  
<!-- label="爱好"-->  
<!-- width="180">-->  
<!-- </el-table-column>-->* <el-table-column  
 align="right"  
 width="180"  
 fixed="right"  
 >  
 <template slot="header" slot-scope="scope">  
 <el-input  
 v-model="search"  
 size="mini"  
 placeholder="输入姓名搜索"/>  
 </template>  
 <template slot-scope="scope">  
 <el-button  
 size="mini"  
 @click="handleEdit(scope.row)"  
 >编辑  
 </el-button>  
 <el-button  
 size="mini"  
 type="danger"  
 @click="handleDelete(scope.row)">删除  
 </el-button>  
 </template>  
 </el-table-column>  
 </el-table>  
 </div>  
 <el-dialog :title="dialogTitle[dialogStatus]" :visible.sync="dialogFormVisible" center>  
 <el-form :model="form" :rules="rules" ref="student">  
 <el-form-item label="姓名" :label-width="formLabelWidth" prop="name">  
 <el-input v-model="form.name" autocomplete="off"></el-input>  
 </el-form-item>  
 <el-form-item label="性别" :label-width="formLabelWidth" prop="sex">  
 <el-select v-model="form.sex" placeholder="请选择" style="width: 100%">  
 <el-option  
 v-for="item in options[0]"  
 :key="item.value"  
 :value="item.label">  
 </el-option>  
 </el-select>  
 </el-form-item>  
 <el-form-item label="年龄" :label-width="formLabelWidth" prop="age">  
 <el-input v-model.number="form.age" autocomplete="off"></el-input>  
 </el-form-item>  
 <el-form-item label="生日" :label-width="formLabelWidth" prop="birth">  
 <el-date-picker  
 v-model="form.birth"  
 format="yyyy 年 MM 月"  
 value-format="yyyy.MM"  
 type="month"  
 placeholder="选择时间">  
 </el-date-picker>  
 </el-form-item>  
 <el-form-item label="薪水" :label-width="formLabelWidth" prop="salary">  
 <el-input v-model.number="form.salary" autocomplete="off"></el-input>  
 </el-form-item>  
 <el-form-item label="专业" :label-width="formLabelWidth" prop="major">  
 <el-select v-model="form.major" filterable placeholder="请选择" style="width: 100%">  
 <el-option  
 v-for="item in options[1]"  
 :key="item.value"  
 :value="item.label">  
 </el-option>  
 </el-select>  
 </el-form-item>  
 <el-form-item label="课程" :label-width="formLabelWidth" prop="course">  
 <el-select v-model="form.course" filterable placeholder="请选择课程" style="width: 100%">  
 <el-option  
 v-for="item in options[2]"  
 :key="item.value"  
 :value="item.label">  
 </el-option>  
 </el-select>  
 </el-form-item>  
 <el-form-item label="班级" :label-width="formLabelWidth" prop="clazz">  
 <el-input v-model="form.clazz" autocomplete="off"></el-input>  
 </el-form-item>  
 <el-form-item label="是否党员" :label-width="formLabelWidth" prop="partyMember">  
 <el-select v-model="form.partyMember" placeholder="请选择" style="width: 100%">  
 <el-option  
 v-for="item in options[3]"  
 :key="item.value"  
 :value="item.label">  
 </el-option>  
 </el-select>  
 </el-form-item>  
 <el-form-item label="学历" :label-width="formLabelWidth" prop="diploma">  
 <el-input v-model="form.diploma" autocomplete="off"></el-input>  
 </el-form-item>  
 <el-form-item label="证书" :label-width="formLabelWidth" prop="certificate">  
 <el-input v-model="form.certificate" autocomplete="off"></el-input>  
 </el-form-item>  
 <el-form-item label="婚姻状况" :label-width="formLabelWidth" prop="maritalStatus">  
 <el-radio v-model="form.maritalStatus" label="未婚">未婚</el-radio>  
 <el-radio v-model="form.maritalStatus" label="已婚">已婚</el-radio>  
 </el-form-item>  
 <el-form-item label="电话" :label-width="formLabelWidth" prop="phoneNumber">  
 <el-input v-model.number="form.phoneNumber"></el-input>  
 </el-form-item>  
 <el-form-item label="邮箱" :label-width="formLabelWidth" prop="email">  
 <el-input v-model="form.email"></el-input>  
 </el-form-item>  
 <el-form-item label="地址" :label-width="formLabelWidth" prop="address">  
 <el-input v-model="form.address" autocomplete="off"></el-input>  
 </el-form-item>  
 <el-form-item label="职位" :label-width="formLabelWidth" prop="position">  
 <el-input v-model="form.position" autocomplete="off"></el-input>  
 </el-form-item>  
 <el-form-item label="民族" :label-width="formLabelWidth" prop="ethnic">  
 <el-input v-model="form.ethnic" autocomplete="off"></el-input>  
 </el-form-item>  
 <el-form-item label="籍贯" :label-width="formLabelWidth" prop="nativePlace">  
*<!-- <el-cascader-->  
<!-- v-model="form.nativePlace"-->  
<!-- :options="regions"-->  
<!-- :props="{ expandTrigger: 'hover' }"-->  
<!-- @change="addressChange"-->  
<!-- ></el-cascader>-->* <el-cascader size="large" :options="regions" v-model="selectregon"  
 @change="addressChange"  
 :props="{ expandTrigger: 'hover' }"  
 >  
 </el-cascader>  
 </el-form-item>  
 <el-form-item label="毕业时间" :label-width="formLabelWidth" prop="yearOfGraduation">  
 <el-date-picker  
 v-model="form.yearOfGraduation"  
 format="yyyy 年 MM 月"  
 value-format="yyyy.MM"  
 type="month"  
 placeholder="选择时间">  
 </el-date-picker>  
 </el-form-item>  
 <el-form-item label="工作时间" :label-width="formLabelWidth" prop="yearOfWork">  
 <el-date-picker  
 v-model="form.yearOfWork"  
 type="month"  
 format="yyyy 年 MM 月"  
 value-format="yyyy.MM"  
 placeholder="选择时间">  
 </el-date-picker>  
 </el-form-item>  
 <el-form-item label="毕业学校" :label-width="formLabelWidth" prop="schoolOfGraduation">  
 <el-input v-model="form.schoolOfGraduation" autocomplete="off"></el-input>  
 </el-form-item>  
*<!-- <el-form-item required label="爱好" :label-width="formLabelWidth" prop="hobby">-->  
<!-- <el-checkbox-group v-model="form.hobby">-->  
<!-- <el-checkbox label="唱歌" name="hobby"></el-checkbox>-->  
<!-- <el-checkbox label="跳舞" name="hobby"></el-checkbox>-->  
<!-- <el-checkbox label="乐器" name="hobby"></el-checkbox>-->  
<!-- <el-checkbox label="书法" name="hobby"></el-checkbox>-->  
<!-- </el-checkbox-group>-->  
<!-- </el-form-item>-->* </el-form>  
 <div slot="footer" class="dialog-footer">  
 <el-button @click="dialogFormVisible = false">取 消</el-button>  
 <el-button type="primary" @click="submitStudent">确 定</el-button>  
 </div>  
 </el-dialog>  
 </div>  
</template>  
  
<script>  
 *// @ is an alias to /src* import { provinceAndCityData,CodeToText,TextToCode } from 'element-china-area-data'  
  
  
 export default {  
 name: 'Home',  
 components: {},  
 data() {  
 *// 手机号验证* let checkPhone = (rule, value, callback) => {  
 const phoneReg = /^1[3|4|5|6|7|8][0-9]{9}$/  
 if (!value) {  
 return callback(new Error('电话号码不能为空'))  
 }  
 *setTimeout*(() => {  
  
 if (!Number.isInteger(+value)) {  
 callback(new Error('请输入数字值'))  
 } else {  
 if (phoneReg.test(value)) {  
 callback()  
 } else {  
 callback(new Error('电话号码格式不正确'))  
 }  
 }  
 }, 100)  
 };  
  
 return {  
 selectregon:[],  
 dialogTitle: {  
 addTeacher: "新增教师",  
 editTeacher: "编辑教师"  
 },  
 dialogStatus: "",  
 regions: provinceAndCityData,  
 rules: {  
 name: [  
 {required: true, message: '请输入姓名', trigger: 'blur'}  
 ],  
 sex: [  
 {required: true, message: '请选择性别', trigger: 'blur'}  
 ],  
 age: [  
 {required: true, message: '年龄不能为空'},  
 {type: 'number', message: '年龄必须为数字值'}  
 ],  
 salary: [  
 {required: true, type: 'number', message: '薪水必须为数字值'}  
 ],  
 major: [  
 {required: true, message: '请选择专业', trigger: 'change'}  
 ],  
 course: [  
 {required: true, message: '请选择课程', trigger: 'change'}  
 ],  
 email: [  
 {required: true, message: '请输入邮箱地址', trigger: 'blur'},  
 {type: 'email', message: '请输入正确的邮箱地址', trigger: ['blur', 'change']}  
 ],  
 phoneNumber: [  
 {required: true,validator: checkPhone, trigger: 'blur' }  
 ],  
 birth: [  
 {required: true, message: '选择生日'}  
 ],  
 partyMember: [  
 {required: true, message: '选择是否党员'}  
 ],  
 maritalStatus: [  
 {required: true, message: '选择婚姻状况'}  
 ],  
 ethnic: [  
 {required: true, message: '输入民族'}  
 ],  
 nativePlace: [  
 {required: true, message: '选择籍贯'}  
 ],  
 yearOfGraduation: [  
 {required: true, message: '选择毕业时间'}  
 ],  
 yearOfWork: [  
 {required: true, message: '选择工作时间'}  
 ],  
 schoolOfGraduation: [  
 {required: true, message: '选择毕业学校'}  
 ],  
 hobby: [  
 {required: true, message: '选择爱好'}  
 ],  
 },  
 options: [  
 [{  
 value: '1',  
 label: '男'  
 }, {  
 value: '2',  
 label: '女'  
 }],  
 [  
 {  
 value: '1',  
 label: '软件工程'  
 },  
 {  
 value: '2',  
 label: '国际经济与贸易'  
 },  
 {  
 value: '3',  
 label: '会计'  
 },  
 {  
 value: '4',  
 label: '车辆工程'  
 }  
 ],  
 [  
 {  
 value: '1',  
 label: 'JAVAEE'  
 },  
 {  
 value: '2',  
 label: '设计模式'  
 },  
 {  
 value: '3',  
 label: '政治经济学'  
 },  
 {  
 value: '4',  
 label: '思修'  
 },  
 {  
 value: '5',  
 label: '高等数学'  
 }  
 ],  
 [{  
 value: '1',  
 label: "是"  
 }, {  
 value: '2',  
 label: "否"  
 }],  
 ],  
 msg: '',  
 teachers: [],  
 search: '',  
 dialogFormVisible: false,  
 form: {  
 name: '',  
 age: '',  
 sex: '',  
 birth: '',  
 salary: '',  
 major: '',  
 course: '',  
 clazz: '',  
 partyMember: '',  
 diploma: '',  
 certificate: '',  
 maritalStatus: '',  
 phoneNumber: '',  
 email: '',  
 address: '',  
 position: '',  
 ethnic: '',  
 nativePlace: [],  
 yearOfGraduation: '',  
 yearOfWork: '',  
 schoolOfGraduation: '',  
 *// hobby:'',* },  
 formLabelWidth: '120px',  
 }  
 },  
 created() {  
 this.findAllTeachers();  
 },  
 methods: {  
 submitStudent() {  
 if (this.dialogStatus === 'addTeacher') {  
 this.addTeacher();  
 } else if (this.dialogStatus === "editTeacher") {  
 this.editTeacher();  
 }  
 },  
 findAllTeachers() {  
 this.$axios.get("/first/findAllTeachers")  
 .then((res) => {  
 this.teachers = res.data  
 console.log(this.teachers)  
 })  
 .catch(err => {  
 *alert*("查询失败");  
 console.log(err);  
 });  
 },  
 handleEdit(row) {  
 this.dialogFormVisible = true;  
 this.dialogStatus = "editTeacher";  
 this.form.uid = row.uid;  
 this.form.name = row.name;  
 this.form.age = row.age;  
 this.form.sex = row.sex;  
 this.form.birth = row.birth;  
 this.form.salary = row.salary;  
 this.form.major = row.major;  
 this.form.course = row.course;  
 this.form.clazz = row.clazz;  
 this.form.partyMember = row.partyMember;  
 this.form.diploma = row.diploma;  
 this.form.certificate = row.certificate;  
 this.form.maritalStatus = row.maritalStatus;  
 this.form.phoneNumber = row.phoneNumber;  
 this.form.email = row.email;  
 this.form.address = row.address;  
 this.form.position = row.position;  
 this.form.ethnic = row.ethnic;  
 this.form.nativePlace = row.nativePlace;  
 this.selectregon = [TextToCode[row.nativePlace[0]].code,TextToCode[row.nativePlace[0]][row.nativePlace[1]].code];  
 this.form.yearOfGraduation = row.yearOfGraduation;  
 this.form.yearOfWork = row.yearOfWork;  
 this.form.schoolOfGraduation = row.schoolOfGraduation;  
 *// this.form.hobby = row.hobby;* console.log(this.form.nativePlace)  
 console.log(this.selectregon)  
  
 },  
 handleDelete(row) {  
 this.$confirm('此操作将永久删除该教师, 是否继续?', '提示', {  
 confirmButtonText: '确定',  
 cancelButtonText: '取消',  
 type: 'warning'  
 }).then(() => {  
 this.$axios.post("/first/deleteTeacher?uid=" + row.uid).then((resp) => {  
 this.$message.success("删除成功!")  
 this.findAllTeachers();  
 })  
 }).catch(() => {  
 this.$message({  
 type: 'info',  
 message: '已取消删除'  
 });  
 });  
 },  
 showDialog() {  
 this.form.uid = '';  
 this.form.name = '';  
 this.form.age = '';  
 this.form.sex = '';  
 this.form.birth = '';  
 this.form.salary = '';  
 this.form.major = '';  
 this.form.course = '';  
 this.form.clazz = '';  
 this.form.partyMember = '';  
 this.form.diploma = '';  
 this.form.certificate = '';  
 this.form.maritalStatus = '';  
 this.form.phoneNumber = '';  
 this.form.email = '';  
 this.form.address = '';  
 this.form.position = '';  
 this.form.ethnic = '';  
 this.form.nativePlace = '';  
 this.form.yearOfGraduation = '';  
 this.form.yearOfWork = '';  
 this.form.schoolOfGraduation = '';  
 *// this.form.hobby = '';* this.dialogFormVisible = true;  
 this.dialogStatus = "addTeacher";  
 },  
 addTeacher() {  
 this.$refs.student.validate((valid) => {  
 if (valid) {  
 this.$axios.post("/first/addTeacher", this.form).then((resp) => {  
 if (resp) {  
 this.dialogFormVisible = false;  
 this.$message.success("添加成功！");  
 this.findAllTeachers();  
 }  
 }).catch((error) => {  
 this.$message({  
 message: '添加数据失败，原因是' + error.data.message,  
 type: 'error'  
 })  
 })  
 } else {  
 this.$message({  
 message: '请输入所有字段',  
 type: 'error'  
 })  
 return false;  
 }  
 });  
 },  
 editTeacher() {  
 this.$axios.post("/first/editTeacher", this.form).then((resp) => {  
 this.dialogFormVisible = false;  
 this.$message.success("修改成功!");  
 this.findAllTeachers();  
 })  
 },  
 addressChange(arr){  
 console.log(CodeToText [arr[0]], CodeToText [arr[1]])  
 this.form.nativePlace =[CodeToText [arr[0]], CodeToText [arr[1]]]  
 },  
 }  
 }  
</script>  
<style scoped>  
 .table {  
 margin: 0 auto;  
 clear: both;  
 }  
</style>