package com.ying.springboot;  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.transaction.annotation.EnableTransactionManagement;  
@SpringBootApplication  
@EnableTransactionManagement  
public class SpringbootApplication {  
 public static void main(String[] args) {  
 SpringApplication.run(SpringbootApplication.class, args);  
 }  
 }  
package com.ying.springboot.common;  
public interface Constants {  
 String CODE\_200 = "200"; //成功  
 String CODE\_401 = "401"; // 权限不足  
 String CODE\_400 = "400"; // 参数错误  
 String CODE\_500 = "500"; // 系统错误  
 String CODE\_600 = "600"; // 其他业务异常  
 String DICT\_TYPE\_ICON = "icon";  
 String FILES\_KEY = "FILES\_FRONT\_ALL";  
}  
package com.ying.springboot.common;  
import lombok.AllArgsConstructor;  
import lombok.Data;  
import lombok.NoArgsConstructor;  
@Data  
@NoArgsConstructor  
@AllArgsConstructor  
public class Result {  
 private String code;  
 private String msg;  
 private Object data;  
 public static Result success() {  
 return new Result(Constants.CODE\_200, "", null);  
 }  
 public static Result success(Object data) {  
 return new Result(Constants.CODE\_200, "", data);  
 }  
 public static Result error(String code, String msg) {  
 return new Result(code, msg, null);  
 }  
 public static Result error() {  
 return new Result(Constants.CODE\_500, "系统错误", null);  
 }  
}package com.ying.springboot.config;  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.web.cors.CorsConfiguration;  
import org.springframework.web.cors.UrlBasedCorsConfigurationSource;  
import org.springframework.web.filter.CorsFilter;  
@Configuration  
public class CorsConfig {  
 private static final long MAX\_AGE = 24 \* 60 \* 60;  
 @Bean  
 public CorsFilter corsFilter() {  
 UrlBasedCorsConfigurationSource source = new UrlBasedCorsConfigurationSource();  
 CorsConfiguration corsConfiguration = new CorsConfiguration();  
 corsConfiguration.addAllowedOrigin("\*"); // 1 设置访问源地址  
 corsConfiguration.addAllowedHeader("\*"); // 2 设置访问源请求头  
 corsConfiguration.addAllowedMethod("\*"); // 3 设置访问源请求方法  
 corsConfiguration.setMaxAge(MAX\_AGE);  
 source.registerCorsConfiguration("/\*\*", corsConfiguration); // 4 对接口配置跨域设置  
 return new CorsFilter(source);  
 }  
}  
package com.ying.springboot.config;  
import com.baomidou.mybatisplus.annotation.DbType;  
import com.baomidou.mybatisplus.extension.plugins.MybatisPlusInterceptor;  
import com.baomidou.mybatisplus.extension.plugins.inner.PaginationInnerInterceptor;  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
@Configuration  
public class MPConfig {  
 @Bean  
 public MybatisPlusInterceptor mybatisPlusInterceptor() {  
 MybatisPlusInterceptor interceptor = new MybatisPlusInterceptor();  
 interceptor.addInnerInterceptor(new PaginationInnerInterceptor(DbType.MYSQL));  
 return interceptor;  
 }  
}  
package com.ying.springboot.config;  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
import springfox.documentation.builders.ApiInfoBuilder;  
import springfox.documentation.builders.PathSelectors;  
import springfox.documentation.builders.RequestHandlerSelectors;  
import springfox.documentation.service.ApiInfo;  
import springfox.documentation.service.Contact;  
import springfox.documentation.spi.DocumentationType;  
import springfox.documentation.spring.web.plugins.Docket;  
@Configuration  
public class SwaggerConfig {  
 Boolean swaggerEnabled = true;  
 @Bean  
 public Docket createRestApi() {  
 return new Docket(DocumentationType.OAS\_30)  
 .apiInfo(apiInfo())  
 .enable(swaggerEnabled)//true  
 .select()  
 .apis(RequestHandlerSelectors.basePackage("com.ying.springboot.controller"))  
 .paths(PathSelectors.any())  
 .build();  
 }  
 private ApiInfo apiInfo() {  
 return new ApiInfoBuilder()  
 .title("后台管理系统swagger业务")  
 .contact(new Contact("yjt", "http://www.baidu.com", "yjt@qq.com"))  
 .version("1.0")  
 .description("不需要描述")  
 .build();  
 }  
}  
package com.ying.springboot.controller;  
import com.baomidou.mybatisplus.extension.plugins.pagination.Page;  
import com.ying.springboot.common.Result;  
import org.springframework.web.bind.annotation.\*;  
import javax.annotation.Resource;  
import java.util.List;  
@RestController  
@RequestMapping("/course")  
public class CourseController {  
 @Resource  
 private ICourseService courseService;  
 @GetMapping  
 public Result getAll(){return Result.success(courseService.list());}  
 @GetMapping("/{id}")  
 public Result getById(@PathVariable Integer id){return Result.success(courseService.getById(id));}  
 @PostMapping  
 public Result save(@RequestBody Course course){return Result.success(courseService.saveOrUpdate(course));}  
 @DeleteMapping("/{id}")  
 public Result delete(@PathVariable Integer id) {  
 return Result.success(courseService.removeById(id));  
 }  
 @PostMapping("/del/batch")  
 public Result deleteBatch(@RequestBody List<Integer> ids) {  
 return Result.success(courseService.removeByIds(ids));  
 }  
 @GetMapping("/page")  
 public Result findPage(@RequestParam String name,  
 @RequestParam Integer pageNum,  
 @RequestParam Integer pageSize){  
 Page<Course> page=courseService.findPage(new Page<>(pageNum,pageSize),name);  
 return Result.success(page);  
 }  
}  
package com.ying.springboot.controller;  
import cn.hutool.core.util.StrUtil;  
import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;  
import com.baomidou.mybatisplus.core.metadata.IPage;  
import com.baomidou.mybatisplus.extension.plugins.pagination.Page;  
import com.ying.springboot.common.Constants;  
import com.ying.springboot.common.Result;  
import com.ying.springboot.controller.dto.UserDTO;  
import com.ying.springboot.entity.User;  
import com.ying.springboot.service.UserService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
import java.util.List;  
import com.ying.springboot.service.IDeadlineService;  
import com.ying.springboot.entity.Deadline;  
import org.springframework.web.bind.annotation.RestController;  
import javax.annotation.Resource;  
@RestController  
@RequestMapping("/deadline")  
public class DeadlineController {  
 @Resource  
 private IDeadlineService deadlineService;  
 @GetMapping  
 public Result getAll(){return Result.success(deadlineService.list());}  
 @GetMapping("/{id}")  
 public Result getById(@PathVariable Integer id){return Result.success(deadlineService.getById(id));}  
 @PostMapping  
 public Result save(@RequestBody Deadline deadline){return Result.success(deadlineService.saveOrUpdate(deadline));}  
 @DeleteMapping("/{id}")  
 public Result delete(@PathVariable Integer id) {  
 return Result.success(deadlineService.removeById(id));  
 }  
 @PostMapping("/del/batch")  
 public Result deleteBatch(@RequestBody List<Integer> ids) {  
 return Result.success(deadlineService.removeByIds(ids));  
 }  
 @GetMapping("/page")  
 public Result findPage(@RequestParam Integer pageNum,  
 @RequestParam Integer pageSize){  
 IPage<Deadline> page=new Page<>(pageNum,pageSize);  
 QueryWrapper<Deadline> queryWrapper=new QueryWrapper<>();  
 return Result.success(deadlineService.page(page,queryWrapper));  
 }  
}  
package com.ying.springboot.controller;  
import com.baomidou.mybatisplus.core.conditions.Wrapper;  
import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;  
import com.baomidou.mybatisplus.core.conditions.update.UpdateWrapper;  
import com.baomidou.mybatisplus.core.metadata.IPage;  
import com.baomidou.mybatisplus.extension.plugins.pagination.Page;  
import com.ying.springboot.common.Result;  
import com.ying.springboot.entity.User;  
import com.ying.springboot.mapper.IntentionMapper;  
import com.ying.springboot.service.UserService;  
import io.swagger.models.auth.In;  
import org.springframework.transaction.annotation.Isolation;  
import org.springframework.transaction.annotation.Transactional;  
import org.springframework.web.bind.annotation.\*;  
import java.io.Console;  
import java.util.List;  
import java.util.concurrent.TimeUnit;  
import com.ying.springboot.service.IIntentionService;  
import com.ying.springboot.entity.Intention;  
import org.springframework.web.bind.annotation.RestController;  
import javax.annotation.Resource;  
@Transactional(isolation = Isolation.SERIALIZABLE )  
@RestController  
@RequestMapping("/intention")  
public class IntentionController {  
 @Resource  
 private IIntentionService intentionService;  
 @Resource  
 private UserService userService;  
 @GetMapping  
 public Result getAll(){return Result.success(intentionService.list());}  
 @GetMapping("/{id}")  
 public Result getById(@PathVariable Integer id){return Result.success(intentionService.getById(id));}  
 @PostMapping  
 public Result save(@RequestBody Intention intention){return Result.success(intentionService.saveOrUpdate(intention));}  
 @DeleteMapping("/{id}")  
 public Result delete(@PathVariable Integer id) {  
 return Result.success(intentionService.removeById(id));  
 }  
 @PostMapping("/del/batch")  
 public Result deleteBatch(@RequestBody List<Integer> ids) {  
 return Result.success(intentionService.removeByIds(ids));  
 }  
 @GetMapping("/page")  
 public Result findPage(@RequestParam Integer pageNum,  
 @RequestParam Integer pageSize,  
 @RequestParam (defaultValue="") Integer id) {  
 QueryWrapper<Intention> queryWrapper=new QueryWrapper<>();  
 queryWrapper.orderByAsc("sequence");  
 queryWrapper.eq("student\_id",id);  
 Page<Intention> page=intentionService.page(new Page<>(pageNum,pageSize),queryWrapper);  
 List<Intention> records=page.getRecords();  
 for (Intention record :records){  
 User user = userService.getById(record.getTeacherId());  
 if (user!=null){  
 record.setTeacherName(user.getNickname());  
 }  
 }  
 return Result.success(records);  
 }  
 @GetMapping("/addIntention")  
 public Result addIntention(@RequestParam Integer student\_id,  
 @RequestParam Integer teacher\_id){  
 QueryWrapper<Intention> queryWrapper=new QueryWrapper<>();  
 queryWrapper.eq("student\_id",student\_id);  
 List<Intention> intentionList=intentionService.list(queryWrapper);  
 Intention intention=new Intention();  
 intention.setStudentId(student\_id);  
 intention.setTeacherId(teacher\_id);  
 intention.setSequence(intentionList.size()+1);  
 return Result.success(intentionService.save(intention));  
 }  
 @GetMapping("/plus")  
 public Result sequencePlus(@RequestParam Integer studentId,  
 @RequestParam Integer sequence){  
 UpdateWrapper<Intention> updateWrapper=new UpdateWrapper<>();  
 updateWrapper.eq("student\_id",studentId)  
 .eq("sequence",sequence-1);  
 updateWrapper.set("sequence",sequence);  
 return Result.success();  
 }  
 @GetMapping("/intentionUp")  
 public Result sequenceReduce(@RequestParam Integer id,  
 @RequestParam Integer studentId,  
 @RequestParam Integer sequence) {  
 UpdateWrapper<Intention> plusWrapper=new UpdateWrapper<>();  
 plusWrapper.eq("student\_id",studentId)  
 .eq("sequence",sequence-1);  
 plusWrapper.set("sequence",sequence);  
 intentionService.update(null, plusWrapper);  
 UpdateWrapper<Intention> reduceWrapper=new UpdateWrapper<>();  
 reduceWrapper.eq("id",id);  
 reduceWrapper.set("sequence",sequence-1);  
 return Result.success(intentionService.update(null, reduceWrapper));  
 }  
 @GetMapping("/push/{id}")  
 public Result sequencePush(@PathVariable Integer id){  
 QueryWrapper<Intention> queryWrapper=new QueryWrapper<>();  
 queryWrapper.eq("student\_id",id)  
 .eq("sequence",1);  
 Intention intention=intentionService.getOne(queryWrapper);  
 Integer sid=intention.getStudentId();  
 Integer seq=intention.getSequence();  
 UpdateWrapper<Intention> updateWrapper=new UpdateWrapper<>();  
 updateWrapper.eq("student\_id",sid)  
 .gt("sequence",seq)  
 .setSql("sequence=sequence-1");  
 intentionService.update(null, updateWrapper);  
 return Result.success(intentionService.removeById(intention.getId()));  
 }  
}  
package com.ying.springboot.controller;  
import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;  
import com.baomidou.mybatisplus.core.metadata.IPage;  
import com.baomidou.mybatisplus.extension.plugins.pagination.Page;  
import com.ying.springboot.common.Result;  
import com.ying.springboot.entity.Student;  
import com.ying.springboot.entity.User;  
import com.ying.springboot.service.IStudentService;  
import com.ying.springboot.service.UserService;  
import org.springframework.web.bind.annotation.\*;  
import javax.annotation.Resource;  
import java.util.List;  
@RestController  
@RequestMapping("/student")  
public class StudentController {  
 @Resource  
 private IStudentService studentService;  
 @Resource  
 private UserService userService;  
 @GetMapping  
 public Result getAll(){return Result.success(studentService.list());}  
 @GetMapping("/{id}")  
 public Result getById(@PathVariable Integer id){return Result.success(studentService.getById(id));}  
 @PostMapping  
 public Result save(@RequestBody Student student){return Result.success(studentService.saveOrUpdate(student));}  
 @DeleteMapping("/{id}")  
 public Result delete(@PathVariable Integer id) {  
 return Result.success(studentService.removeById(id));  
 }  
 @PostMapping("/del/batch")  
 public Result deleteBatch(@RequestBody List<Integer> ids) {  
 return Result.success(studentService.removeByIds(ids));  
 }  
 @GetMapping("/saveTeacherId/{id}")  
 public Result saveTeacherId(@RequestParam Integer id,  
 @RequestParam Integer tid) {  
 Student student=new Student();  
 student.setId(id);  
 student.setTeacherId(tid);  
 return Result.success(studentService.updateById(student));  
 }  
 @GetMapping("/page")  
 public Result findPage(@RequestParam Integer pageNum,  
 @RequestParam Integer pageSize){  
 IPage<Student> page=new Page<>(pageNum,pageSize);  
 QueryWrapper<Student> queryWrapper=new QueryWrapper<>();  
 return Result.success(studentService.page(page,queryWrapper));  
 }  
}  
package com.ying.springboot.controller;  
import com.ying.springboot.common.Result;  
import com.ying.springboot.entity.Teacher;  
import com.ying.springboot.service.ITeacherService;  
import org.springframework.web.bind.annotation.\*;  
import javax.annotation.Resource;  
import java.util.List;  
@RestController  
@RequestMapping("/teacher")  
public class TeacherController {  
 @Resource  
 private ITeacherService teacherService;  
 @GetMapping  
 public Result getAll(){return Result.success(teacherService.list());}  
 @GetMapping("/{id}")  
 public Result getById(@PathVariable Integer id){return Result.success(teacherService.getById(id));}  
 @PostMapping  
 public Result save(@RequestBody Teacher teacher){return Result.success(teacherService.saveOrUpdate(teacher));}  
 @DeleteMapping("/{id}")  
 public Result delete(@PathVariable Integer id) {  
 return Result.success(teacherService.removeById(id));  
 }  
 @PostMapping("/del/batch")  
 public Result deleteBatch(@RequestBody List<Integer> ids) {  
 return Result.success(teacherService.removeByIds(ids));  
 }  
}  
package com.ying.springboot.controller;  
import cn.hutool.core.util.StrUtil;  
import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;  
import com.baomidou.mybatisplus.core.metadata.IPage;  
import com.baomidou.mybatisplus.extension.plugins.pagination.Page;  
import com.ying.springboot.common.Constants;  
import com.ying.springboot.common.Result;  
import com.ying.springboot.controller.dto.UserDTO;  
import com.ying.springboot.entity.User;  
import com.ying.springboot.service.UserService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
import java.util.List;  
@RestController  
@RequestMapping("/user")  
public class UserController {  
 @Autowired  
 private UserService userService;  
 @PostMapping("/login")  
 public Result login(@RequestBody UserDTO userDTO){  
 String username=userDTO.getUsername();  
 String password=userDTO.getPassword();  
 if (StrUtil.isBlank(username)||StrUtil.isBlank(password)){  
 return Result.error(Constants.CODE\_400,"参数错误");  
 }  
 UserDTO dto=userService.login(userDTO);  
 return Result.success(dto);  
 }  
 @PostMapping("/register")  
 public Result register(@RequestBody UserDTO userDTO){  
 String username=userDTO.getUsername();  
 String password=userDTO.getPassword();  
 if (StrUtil.isBlank(username)||StrUtil.isBlank(password)){  
 return Result.error(Constants.CODE\_400,"参数错误");  
 }  
 return Result.success(userService.register(userDTO));  
 }  
 @GetMapping  
 public Result getAll(){return Result.success(userService.list());}  
 @GetMapping("/{id}")  
 public Result getById(@PathVariable Integer id){return Result.success(userService.getById(id));}  
 @PostMapping  
 public Result save(@RequestBody User user){return Result.success(userService.saveOrUpdate(user));}  
 @DeleteMapping("/{id}")  
 public Result delete(@PathVariable Integer id) {  
 return Result.success(userService.removeById(id));  
 }  
 @GetMapping("/username/{username}")  
 public Result findOne(@PathVariable String username){  
 QueryWrapper<User> queryWrapper=new QueryWrapper<>();  
 queryWrapper.eq("username",username);  
 return Result.success(userService.getOne(queryWrapper));  
 }  
 @PostMapping("/del/batch")  
 public Result deleteBatch(@RequestBody List<Integer> ids) {  
 return Result.success(userService.removeByIds(ids));  
 }  
 @GetMapping("/page")  
 public Result findPage(@RequestParam Integer pageNum,  
 @RequestParam Integer pageSize,  
 @RequestParam (defaultValue="") String username ,  
 @RequestParam (defaultValue="") String email,  
 @RequestParam (defaultValue="") String address  
 ){  
 IPage<User> page=new Page<>(pageNum,pageSize);  
 QueryWrapper<User> queryWrapper=new QueryWrapper<>();  
 if (!"".equals(username)){  
 queryWrapper.like("username",username);  
 }  
 if (!"".equals(email)){  
 queryWrapper.like("email",email);  
 }  
 if (!"".equals(address)){  
 queryWrapper.like("address",address);  
 }  
 return Result.success(userService.page(page,queryWrapper));  
 }  
 @GetMapping("/page/teacher")  
 public Result findTeacher(@RequestParam Integer pageNum,  
 @RequestParam Integer pageSize,  
 @RequestParam (defaultValue="") String username  
 ){  
 IPage<User> page=new Page<>(pageNum,pageSize);  
 QueryWrapper<User> queryWrapper=new QueryWrapper<>();  
 queryWrapper.eq("role","老师");  
 if (!"".equals(username)){  
 queryWrapper.like("username",username);  
 }  
 return Result.success(userService.page(page,queryWrapper));  
 }  
 @GetMapping("/page/student")  
 public Result findStudent(@RequestParam Integer pageNum,  
 @RequestParam Integer pageSize,  
 @RequestParam (defaultValue="") Integer id  
 ){  
 IPage<User> page=new Page<>(pageNum,pageSize);  
 QueryWrapper<User> queryWrapper=new QueryWrapper<>();  
 queryWrapper.inSql("id","SELECT student\_id FROM intention WHERE teacher\_id="+id+" AND sequence=1");  
 return Result.success(userService.page(page,queryWrapper));  
 }  
 @GetMapping("/page/ChosenStudent")  
 public Result findChosenStudent(@RequestParam Integer pageNum,  
 @RequestParam Integer pageSize,  
 @RequestParam (defaultValue="") Integer id  
 ){  
 IPage<User> page=new Page<>(pageNum,pageSize);  
 QueryWrapper<User> queryWrapper=new QueryWrapper<>();  
 queryWrapper.inSql("id","SELECT id FROM student WHERE teacher\_id="+id);  
 return Result.success(userService.page(page,queryWrapper));  
 }  
 @GetMapping("/page/findSelectedTeacher")  
 public Result findSelectedTeacher(@RequestParam Integer pageNum,  
 @RequestParam Integer pageSize,  
 @RequestParam (defaultValue="") Integer id  
 ){  
 IPage<User> page=new Page<>(pageNum,pageSize);  
 QueryWrapper<User> queryWrapper=new QueryWrapper<>();  
 queryWrapper.inSql("id","SELECT teacher\_id FROM intention WHERE student\_id="+id);  
 return Result.success(userService.page(page,queryWrapper));  
 }  
 @GetMapping("/page/result")  
 public Result getResult(@RequestParam Integer pageNum,  
 @RequestParam Integer pageSize  
 ){  
 IPage<User> page=userService.getResult(new Page<>(pageNum,pageSize));  
 return Result.success(page);  
 }  
}  
package com.ying.springboot.controller.dto;  
import lombok.Data;  
@Data  
public class ResDTO {  
 private Integer id;  
 private String 学号;  
 private String 姓名;  
 private String 导师姓名;  
 private String 导师邮箱;  
}  
package com.ying.springboot.controller.dto;  
import lombok.Data;  
@Data  
public class UserDTO {  
 private Integer id;  
 private String username;  
 private String password;  
 private String nickname;  
 private String role;  
}  
package com.ying.springboot.ecxeption;  
import com.ying.springboot.common.Result;  
import org.springframework.web.bind.annotation.ControllerAdvice;  
import org.springframework.web.bind.annotation.ResponseBody;  
@ControllerAdvice  
public class GlobalExceptionHandler {  
 @org.springframework.web.bind.annotation.ExceptionHandler(ServiceException.class)  
 @ResponseBody  
 public Result handle(ServiceException se){  
 return Result.error(se.getCode(),se.getMessage());  
 }  
}  
package com.ying.springboot.ecxeption;  
import lombok.Getter;  
@Getter  
public class ServiceException extends RuntimeException{  
 private String code;  
 public ServiceException(String code,String msg){  
 super(msg);  
 this.code=code;  
 }  
}  
package com.ying.springboot.entity;  
import com.baomidou.mybatisplus.annotation.IdType;  
import com.baomidou.mybatisplus.annotation.TableId;  
import java.io.Serializable;  
import java.time.LocalDateTime;  
import io.swagger.annotations.ApiModel;  
import io.swagger.annotations.ApiModelProperty;  
import lombok.Getter;  
import lombok.Setter;  
@Getter  
@Setter  
@ApiModel(value = "Deadline对象", description = "")  
public class Deadline implements Serializable {  
 private static final long serialVersionUID = 1L;  
 @TableId(value = "id", type = IdType.AUTO)  
 private Integer id;  
 private String first;  
 private String second;  
 private String third;  
}  
package com.ying.springboot.entity;  
import com.baomidou.mybatisplus.annotation.IdType;  
import com.baomidou.mybatisplus.annotation.TableField;  
import com.baomidou.mybatisplus.annotation.TableId;  
import java.io.Serializable;  
import io.swagger.annotations.ApiModel;  
import io.swagger.annotations.ApiModelProperty;  
import lombok.Getter;  
import lombok.Setter;  
@Getter  
@Setter  
@ApiModel(value = "Intention对象", description = "")  
public class Intention implements Serializable {  
 private static final long serialVersionUID = 1L;  
 @TableId(value = "id", type = IdType.AUTO)  
 private Integer id;  
 private Integer studentId;  
 private Integer teacherId;  
 @ApiModelProperty("志愿顺序")  
 private Integer sequence;  
 @TableField(exist = false)  
 private String teacherName;  
}  
package com.ying.springboot.entity;  
import java.io.Serializable;  
import java.math.BigDecimal;  
import io.swagger.annotations.ApiModel;  
import io.swagger.annotations.ApiModelProperty;  
import lombok.Getter;  
import lombok.Setter;  
@Getter  
@Setter  
@ApiModel(value = "Student对象", description = "")  
public class Student implements Serializable {  
 private static final long serialVersionUID = 1L;  
 private Integer id;  
 @ApiModelProperty("绩点")  
 private BigDecimal gpa;  
 @ApiModelProperty("自我介绍")  
 private String introduction;  
 @ApiModelProperty("导师id")  
 private Integer teacherId;  
}  
package com.ying.springboot.entity;  
import io.swagger.annotations.ApiModel;  
import io.swagger.annotations.ApiModelProperty;  
import lombok.Getter;  
import lombok.Setter;  
import java.io.Serializable;  
@Getter  
@Setter  
@ApiModel(value = "Teacher对象", description = "")  
public class Teacher implements Serializable {  
 private static final long serialVersionUID = 1L;  
 private Integer id;  
 @ApiModelProperty("职称")  
 private String title;  
 @ApiModelProperty("招生人数")  
 private Integer enrollment;  
 @ApiModelProperty("要求")  
 private String requirement;  
 @ApiModelProperty("自我介绍")  
 private String introduction;  
}  
package com.ying.springboot.entity;  
import com.baomidou.mybatisplus.annotation.TableName;  
import lombok.Data;  
@Data  
@TableName(value = "sys\_user")  
public class User {  
 private Integer id;  
 private String username;  
 private String password;  
 private String nickname;  
 private String email;  
 private String phone;  
 private String address;  
 private String avatarUrl;  
 private String role;  
}  
package com.ying.springboot.mapper;  
import com.ying.springboot.entity.Deadline;  
import com.baomidou.mybatisplus.core.mapper.BaseMapper;  
import org.apache.ibatis.annotations.Mapper;  
@Mapper  
public interface DeadlineMapper extends BaseMapper<Deadline> {  
}  
package com.ying.springboot.mapper;  
import com.ying.springboot.entity.Intention;  
import com.baomidou.mybatisplus.core.mapper.BaseMapper;  
import org.apache.ibatis.annotations.Mapper;  
@Mapper  
public interface IntentionMapper extends BaseMapper<Intention> {  
}  
package com.ying.springboot.mapper;  
import com.ying.springboot.entity.Student;  
import com.baomidou.mybatisplus.core.mapper.BaseMapper;  
import org.apache.ibatis.annotations.Mapper;  
@Mapper  
public interface StudentMapper extends BaseMapper<Student> {  
}  
package com.ying.springboot.mapper;  
import com.ying.springboot.entity.Teacher;  
import com.baomidou.mybatisplus.core.mapper.BaseMapper;  
import org.apache.ibatis.annotations.Mapper;  
@Mapper  
public interface TeacherMapper extends BaseMapper<Teacher> {  
}  
package com.ying.springboot.mapper;  
import com.baomidou.mybatisplus.core.mapper.BaseMapper;  
import com.baomidou.mybatisplus.core.metadata.IPage;  
import com.baomidou.mybatisplus.extension.plugins.pagination.Page;  
import com.ying.springboot.entity.User;  
import org.apache.ibatis.annotations.Mapper;  
import org.apache.ibatis.annotations.Param;  
import java.util.List;  
@Mapper  
public interface UserMapper extends BaseMapper<User> {  
 IPage<User> getResult(Page<User> page);  
}  
package com.ying.springboot.service;  
import com.ying.springboot.entity.Deadline;  
import com.baomidou.mybatisplus.extension.service.IService;  
public interface IDeadlineService extends IService<Deadline> {  
}  
package com.ying.springboot.service;  
import com.ying.springboot.entity.Intention;  
import com.baomidou.mybatisplus.extension.service.IService;  
public interface IIntentionService extends IService<Intention> {  
}  
package com.ying.springboot.service;  
import com.ying.springboot.entity.Student;  
import com.baomidou.mybatisplus.extension.service.IService;  
public interface IStudentService extends IService<Student> {  
}  
package com.ying.springboot.service;  
import com.ying.springboot.entity.Teacher;  
import com.baomidou.mybatisplus.extension.service.IService;  
public interface ITeacherService extends IService<Teacher> {  
}  
package com.ying.springboot.service;  
import com.baomidou.mybatisplus.core.metadata.IPage;  
import com.baomidou.mybatisplus.extension.plugins.pagination.Page;  
import com.baomidou.mybatisplus.extension.service.IService;  
import com.ying.springboot.controller.dto.UserDTO;  
import com.ying.springboot.entity.User;  
public interface UserService extends IService<User> {  
 UserDTO login(UserDTO userDTO);  
 User register(UserDTO userDTO);  
 IPage<User> getResult(Page<User> page);  
}  
package com.ying.springboot.service.impl;  
import com.ying.springboot.entity.Deadline;  
import com.ying.springboot.mapper.DeadlineMapper;  
import com.ying.springboot.service.IDeadlineService;  
import com.baomidou.mybatisplus.extension.service.impl.ServiceImpl;  
import org.springframework.stereotype.Service;  
@Service  
public class DeadlineServiceImpl extends ServiceImpl<DeadlineMapper, Deadline> implements IDeadlineService {  
}  
package com.ying.springboot.service.impl;  
import com.ying.springboot.entity.Intention;  
import com.ying.springboot.mapper.IntentionMapper;  
import com.ying.springboot.service.IIntentionService;  
import com.baomidou.mybatisplus.extension.service.impl.ServiceImpl;  
import org.springframework.stereotype.Service;  
@Service  
public class IntentionServiceImpl extends ServiceImpl<IntentionMapper, Intention> implements IIntentionService {  
}  
package com.ying.springboot.service.impl;  
import com.ying.springboot.entity.Student;  
import com.ying.springboot.mapper.StudentMapper;  
import com.ying.springboot.service.IStudentService;  
import com.baomidou.mybatisplus.extension.service.impl.ServiceImpl;  
import org.springframework.stereotype.Service;  
@Service  
public class StudentServiceImpl extends ServiceImpl<StudentMapper, Student> implements IStudentService {  
}  
package com.ying.springboot.service.impl;  
import com.ying.springboot.entity.Teacher;  
import com.ying.springboot.mapper.TeacherMapper;  
import com.ying.springboot.service.ITeacherService;  
import com.baomidou.mybatisplus.extension.service.impl.ServiceImpl;  
import org.springframework.stereotype.Service;  
@Service  
public class TeacherServiceImpl extends ServiceImpl<TeacherMapper, Teacher> implements ITeacherService {  
}  
package com.ying.springboot.service.impl;  
import cn.hutool.core.bean.BeanUtil;  
import cn.hutool.log.Log;  
import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;  
import com.baomidou.mybatisplus.core.metadata.IPage;  
import com.baomidou.mybatisplus.extension.plugins.pagination.Page;  
import com.baomidou.mybatisplus.extension.service.impl.ServiceImpl;  
import com.ying.springboot.common.Constants;  
import com.ying.springboot.controller.dto.UserDTO;  
import com.ying.springboot.ecxeption.ServiceException;  
import com.ying.springboot.entity.User;  
import com.ying.springboot.mapper.UserMapper;  
import com.ying.springboot.service.UserService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
import javax.annotation.Resource;  
@Service  
public class UserServiceImpl extends ServiceImpl<UserMapper, User> implements UserService {  
 @Resource  
 private UserMapper userMapper;  
 private static final Log LOG= Log.get();  
 @Override  
 public UserDTO login(UserDTO userDTO) {  
 User one=getUserInfo(userDTO);  
 if(one!=null){  
 BeanUtil.copyProperties(one,userDTO,true);  
 return userDTO;  
 }  
 else{  
 throw new ServiceException(Constants.CODE\_600,"用户名或密码错误");  
 }  
 }  
 @Override  
 public User register(UserDTO userDTO) {  
 User one=getUserInfo(userDTO);  
 if (one==null){  
 one=new User();  
 BeanUtil.copyProperties(userDTO,one,true);  
 save(one);//把userDTO数据copy完后存储到数据库中  
 }else{  
 throw new ServiceException(Constants.CODE\_600,"用户名已存在");  
 }  
 return one;  
 }  
 private User getUserInfo(UserDTO userDTO){  
 QueryWrapper<User> queryWrapper = new QueryWrapper<>();  
 queryWrapper.eq("username", userDTO.getUsername());  
 queryWrapper.eq("password", userDTO.getPassword());  
 User one;  
 try {  
 one = getOne(queryWrapper);//从数据库查询用户信息  
 } catch (Exception e) {  
 LOG.error(e);  
 throw new ServiceException(Constants.CODE\_500,"系统错误");  
 }  
 return one;  
 }  
 @Override  
 public IPage<User> getResult(Page<User> page) {  
 return userMapper.getResult(page);  
 }  
}  
package com.ying.springboot.utils;  
import com.baomidou.mybatisplus.generator.FastAutoGenerator;  
import com.baomidou.mybatisplus.generator.config.OutputFile;  
import java.util.Collections;  
public class CodeGenerator {  
 public static void main(String[] args) {  
 generate();  
 }  
 private static void generate(){  
 FastAutoGenerator.create("jdbc:mysql://localhost:3306/wyz?severTimezone=UTC", "root", "root")  
 .globalConfig(builder -> {  
 builder.author("应健霆") // 设置作者  
 .enableSwagger() // 开启 swagger 模式  
 .fileOverride() // 覆盖已生成文件  
 .outputDir("D:\\AHub\\WYZ001\\springboot\\src\\main\\java\\"); // 指定输出目录  
 })  
 .packageConfig(builder -> {  
 builder.parent("com.ying.springboot") // 设置父包名  
 .moduleName("") // 设置父包模块名  
 .pathInfo(Collections.singletonMap(OutputFile.mapperXml, "D:\\AHub\\WYZ001\\springboot\\src\\main\\resources\\mapper\\")); // 设置mapperXml生成路径  
 })  
 .strategyConfig(builder -> {  
 builder.entityBuilder().enableLombok();  
 builder.controllerBuilder().enableHyphenStyle()//开启驼峰转连字符  
 .enableRestStyle();//开启生成@RestController控制器  
 builder.addInclude("") // 设置需要生成的表名  
 .addTablePrefix("sys\_"); // 设置过滤表前缀  
 })  
 .execute();  
 }  
}  
package com.ying.springboot;  
import org.junit.jupiter.api.Test;  
import org.springframework.boot.test.context.SpringBootTest;  
@SpringBootTest  
class SpringbootApplicationTests {  
 @Test  
 void contextLoads() {  
 }  
}  
module.exports = {  
 presets: [  
 '@vue/cli-plugin-babel/preset'  
 ]  
}  
const { defineConfig } = require('@vue/cli-service')  
module.exports = defineConfig({  
 transpileDependencies: true  
})  
<!DOCTYPE html>  
<html lang="">  
 <head>  
 <meta charset="utf-8">  
 <meta http-equiv="X-UA-Compatible" content="IE=edge">  
 <meta name="viewport" content="width=device-width,initial-scale=1.0">  
 <link rel="icon" href="<%= BASE\_URL %>favicon.ico">  
 <title><%= htmlWebpackPlugin.options.title %></title>  
 </head>  
 <body>  
 <noscript>  
 <strong>We're sorry but <%= htmlWebpackPlugin.options.title %> doesn't work properly without JavaScript enabled. Please enable it to continue.</strong>  
 </noscript>  
 <div id="app"></div>  
 </body>  
</html>  
<template>  
 <div id="app">  
 <router-view/>  
 </div>  
</template>  
<style>  
#app{  
 min-height: 100vh;  
}  
</style>  
import Vue from 'vue'  
import ElementUI from 'element-ui';  
import 'element-ui/lib/theme-chalk/index.css';  
import App from './App.vue'  
import router from './router'  
import './assets/global.css'  
import request from "@/utils/request";  
import store from './store'  
Vue.config.productionTip = false  
Vue.use(ElementUI,{size:"small"});  
Vue.prototype.request=request;  
new Vue({  
 router,  
 store,  
 render: h => h(App)  
}).$mount('#app')  
\*{  
 margin: 0;  
 padding: 0;  
 box-sizing: border-box;  
}  
html,body{  
 height: 100%;  
}<template>  
 <el-menu :default-openeds="['1', '3']" style="min-height: 100%; overflow-x: hidden"  
 background-color="rgb(48, 65, 86)"  
 text-color="#fff"  
 active-text-color="#ffd04b"  
 :collapse-transition="false"  
 :collapse="isCollapse"  
 router  
 >  
 <div style="height: 60px; line-height: 60px; text-align: center">  
 <img src="../assets/zjut.png" alt="" style="width: 20px; position: relative; top: 5px; right: 5px">  
 <b style="color: white" v-show="logoTextShow">导师制双向选择系统</b>  
 </div>  
 <div>  
 <el-menu-item index="/home">  
 <i class="el-icon-house"></i>  
 <span slot="title">主页</span>  
 </el-menu-item>  
 <el-menu-item index="/teacherInfo">  
 <i class="el-icon-user"></i>  
 <span slot="title">导师信息</span>  
 </el-menu-item>  
 <div v-if="this.user.role==='学生'">  
 <el-menu-item index="/intention" >  
 <i class="el-icon-reading"></i>  
 <span slot="title">志愿信息</span>  
 </el-menu-item>  
 </div>  
 <div v-if="this.user.role==='老师'">  
 <el-submenu index="teacherPower" >  
 <template slot="title">  
 <i class="el-icon-menu"></i>  
 <span slot="title">教师功能</span>  
 </template>  
 <el-menu-item index="/studentInfo">  
 <i class="el-icon-search"></i>  
 <span slot="title">查看报名</span>  
 </el-menu-item>  
 <el-menu-item index="/ChosenStudentInfo">  
 <i class="el-icon-search"></i>  
 <span slot="title">查看已选</span>  
 </el-menu-item>  
 </el-submenu >  
 </div>  
 <el-submenu index="1">  
 <template slot="title">  
 <i class="el-icon-menu"></i>  
 <span slot="title">系统管理</span>  
 </template>  
 <el-menu-item index="/user">  
 <i class="el-icon-s-custom"></i>  
 <span slot="title">用户管理</span>  
 </el-menu-item>  
 </el-submenu>  
 <el-menu-item index="/progress">  
 <i class="el-icon-time"></i>  
 <span slot="title">查看进度</span>  
 </el-menu-item>  
 <el-menu-item index="/result">  
 <i class="el-icon-data-board"></i>  
 <span slot="title">结果公示</span>  
 </el-menu-item>  
 </div>  
 </el-menu>  
</template>  
<script>  
export default {  
 name: "Aside",  
 props: {  
 isCollapse: Boolean,  
 logoTextShow: Boolean  
 },  
 data(){  
 return{  
 user:localStorage.getItem("user") ? JSON.parse(localStorage.getItem("user")) : {}  
 }  
 }  
}  
</script>  
<style scoped>  
</style>  
<style>  
.el-menu-item.is-active {  
 background-color: rgb(38, 52, 69) !important;  
}  
.el-menu-item:hover {  
 background-color: rgb(38, 52, 69) !important;  
}  
.el-submenu\_\_title:hover {  
 background-color: rgb(38, 52, 69) !important;  
}  
.el-menu--collapse span {  
 visibility: hidden;  
}  
</style>  
<template>  
 <div style="line-height: 60px; display: flex">  
 <div style="flex: 1;">  
 <span :class="collapseBtnClass" style="cursor: pointer; font-size: 18px" @click="collapse"></span>  
 <el-breadcrumb separator="/" style="display: inline-block; margin-left: 10px">  
 <el-breadcrumb-item :to="'/'">首页</el-breadcrumb-item>  
 <el-breadcrumb-item>{{ currentPathName }}</el-breadcrumb-item>  
 </el-breadcrumb>  
 </div>  
 <el-dropdown style="width: 70px; cursor: pointer">  
 <span>{{user.nickname}}</span><i class="el-icon-arrow-down" style="margin-left: 5px"></i>  
 <el-dropdown-menu slot="dropdown" style="width: 100px; text-align: center">  
 <el-dropdown-item style="font-size: 14px; padding: 5px 0">  
 <router-link to="/person">个人信息</router-link>  
 </el-dropdown-item>  
 <el-dropdown-item style="font-size: 14px; padding: 5px 0">  
 <span style="text-decoration: none" @click="logout">退出</span>  
 </el-dropdown-item>  
 </el-dropdown-menu>  
 </el-dropdown>  
 </div>  
</template>  
<script>  
export default {  
 name: "Header",  
 props: {  
 collapseBtnClass: String,  
 },  
 computed: {  
 currentPathName () {  
 return this.$store.state.currentPathName;　　//需要监听的数据  
 }  
 },  
 data(){  
 return{  
 user:localStorage.getItem("user")?JSON.parse(localStorage.getItem("user")):{}  
 }  
 },  
 methods:{  
 collapse(){  
 this.$emit("asideCollapse")  
 },  
 logout(){  
 this.$router.push("/login")  
 localStorage.removeItem("user")  
 this.$message.success("退出成功")  
 }  
 },  
 watch: {  
 currentPathName (newVal, oldVal) {  
 console.log(newVal)  
 }  
 },  
}  
</script>  
<style scoped>  
</style>  
<template>  
 <div class="hello">  
 <h1>{{ msg }}</h1>  
 <p>  
 For a guide and recipes on how to configure / customize this project,<br>  
 check out the  
 <a href="https://cli.vuejs.org" target="\_blank" rel="noopener">vue-cli documentation</a>.  
 </p>  
 <h3>Installed CLI Plugins</h3>  
 <ul>  
 <li><a href="https://github.com/vuejs/vue-cli/tree/dev/packages/%40vue/cli-plugin-babel" target="\_blank" rel="noopener">babel</a></li>  
 <li><a href="https://github.com/vuejs/vue-cli/tree/dev/packages/%40vue/cli-plugin-router" target="\_blank" rel="noopener">router</a></li>  
 </ul>  
 <h3>Essential Links</h3>  
 <ul>  
 <li><a href="https://vuejs.org" target="\_blank" rel="noopener">Core Docs</a></li>  
 <li><a href="https://forum.vuejs.org" target="\_blank" rel="noopener">Forum</a></li>  
 <li><a href="https://chat.vuejs.org" target="\_blank" rel="noopener">Community Chat</a></li>  
 <li><a href="https://twitter.com/vuejs" target="\_blank" rel="noopener">Twitter</a></li>  
 <li><a href="https://news.vuejs.org" target="\_blank" rel="noopener">News</a></li>  
 </ul>  
 <h3>Ecosystem</h3>  
 <ul>  
 <li><a href="https://router.vuejs.org" target="\_blank" rel="noopener">vue-router</a></li>  
 <li><a href="https://vuex.vuejs.org" target="\_blank" rel="noopener">vuex</a></li>  
 <li><a href="https://github.com/vuejs/vue-devtools#vue-devtools" target="\_blank" rel="noopener">vue-devtools</a></li>  
 <li><a href="https://vue-loader.vuejs.org" target="\_blank" rel="noopener">vue-loader</a></li>  
 <li><a href="https://github.com/vuejs/awesome-vue" target="\_blank" rel="noopener">awesome-vue</a></li>  
 </ul>  
 </div>  
</template>  
<script>  
export default {  
 name: 'HelloWorld',  
 props: {  
 msg: String  
 }  
}  
</script>  
<style scoped>  
h3 {  
 margin: 40px 0 0;  
}  
ul {  
 list-style-type: none;  
 padding: 0;  
}  
li {  
 display: inline-block;  
 margin: 0 10px;  
}  
a {  
 color: #42b983;  
}  
</style>  
<template>  
 <el-menu :default-openeds="opens" style="min-height: 100%; overflow-x: hidden"  
 background-color="rgb(48, 65, 86)"  
 text-color="#fff"  
 active-text-color="#ffd04b"  
 :collapse-transition="false"  
 :collapse="isCollapse"  
 router  
 >  
 <div style="height: 60px; line-height: 60px; text-align: center">  
 <img src="../assets/logo.png" alt="" style="width: 20px; position: relative; top: 5px;">  
 <b style="color: white; margin-left: 5px" v-show="logoTextShow">后台管理系统</b>  
 </div>  
 <div v-for="item in menus" :key="item.id">  
 <div v-if="item.path">  
 <el-menu-item :index="item.path">  
 <i :class="item.icon"></i>  
 <span slot="title">{{ item.name }}</span>  
 </el-menu-item>  
 </div>  
 <div v-else>  
 <el-submenu :index="item.id + ''">  
 <template slot="title">  
 <i :class="item.icon"></i>  
 <span slot="title">{{ item.name }}</span>  
 </template>  
 <div v-for="subItem in item.children" :key="subItem.id">  
 <el-menu-item :index="subItem.path">  
 <i :class="subItem.icon"></i>  
 <span slot="title">{{ subItem.name }}</span>  
 </el-menu-item>  
 </div>  
 </el-submenu>  
 </div>  
 </div>  
 </el-menu>  
</template>  
import Vue from 'vue'  
import VueRouter from 'vue-router'  
import store from "@/store";  
Vue.use(VueRouter)  
const routes = [  
 {  
 path: '/',  
 component: () => import('../views/Manage.vue'),  
 redirect: "/home",  
 children: [  
 { path: 'home', name: '首页', component: () => import('../views/Home.vue')},  
 { path: 'teacherInfo', name: '导师信息', component: () => import('../views/TeacherInfo.vue')},  
 { path: 'intention', name: '志愿信息', component: () => import('../views/Intention.vue')},  
 { path: 'studentInfo', name: '查看报名', component: () => import('../views/StudentInfo.vue')},  
 { path: 'ChosenStudentInfo', name: '查看已选', component: () => import('../views/ChosenStudentInfo')},  
 { path: 'progress', name: '查看进度', component: () => import('../views/Progress')},  
 { path: 'result', name: '结果公示', component: () => import('../views/Result')},  
 { path: 'user', name: '用户管理', component: () => import('../views/User.vue')},  
 { path: 'person', name: '个人信息', component: () => import('../views/Person.vue')},  
 ]  
 },  
 {  
 path: '/about',  
 name: 'About',  
 component: () => import('../views/About.vue')  
 },  
 {  
 path: '/login',  
 name: 'Login',  
 component: () => import('../views/Login.vue')  
 },  
 {  
 path: '/register',  
 name: 'Register',  
 component: () => import("@/views/Register")  
 },  
]  
const router = new VueRouter({  
 mode: 'history',  
 base: process.env.BASE\_URL,  
 routes  
})  
router.beforeEach((to, from, next) => {  
 localStorage.setItem("currentPathName", to.name) // 设置当前的路由名称，为了在Header组件中去使用  
 store.commit("setPath") // 触发store的数据更新  
 next() // 放行路由  
})  
export default router  
import Vue from 'vue'  
import Vuex from 'vuex'  
Vue.use(Vuex)  
const store = new Vuex.Store({  
 state: {  
 currentPathName: ''  
 },  
 mutations: {  
 setPath (state) {  
 state.currentPathName = localStorage.getItem("currentPathName")  
 }  
 }  
})  
export default storeimport axios from 'axios'  
const request = axios.create({  
 baseURL: 'http://localhost:9090', // 注意！！ 这里是全局统一加上了 后端接口前缀 前缀，后端必须进行跨域配置！  
 timeout: 50000  
})  
request.interceptors.request.use(config => {  
 config.headers['Content-Type'] = 'application/json;charset=utf-8';  
 return config  
}, error => {  
 return Promise.reject(error)  
});  
request.interceptors.response.use(  
 response => {  
 let res = response.data;  
 if (response.config.responseType === 'blob') {  
 return res  
 }  
 if (typeof res === 'string') {  
 res = res ? JSON.parse(res) : res  
 }  
 return res;  
 },  
 error => {  
 console.log('err' + error) // for debug  
 return Promise.reject(error)  
 }  
)  
export default request  
<template>  
 <div class="about">  
 <h1>This is an about page</h1>  
 </div>  
</template>  
<template>  
 <div>  
 <div style="padding: 10px 0">  
 <el-input style="width: 200px;margin-left: 5px" placeholder="请输入名称" suffix-icon="el-icon-search" v-model="username"></el-input>  
 <el-button style="margin-left: 5px" type="primary" icon="el-icon-search" @click="load">搜索</el-button>  
 <el-button style="margin-left: 5px" type="warning" @click="reset">重置</el-button>  
 </div>  
 <div style="margin: 10px 0">  
 </div>  
 <el-table :data="tableData" border stripe @selection-change="handleSelectionChange">  
 <el-table-column  
 type="selection"  
 width="55"  
 >  
 </el-table-column>  
 <el-table-column prop="id" label="ID" width="80">  
 </el-table-column>  
 <el-table-column prop="username" label="学号" width="140">  
 </el-table-column>  
 <el-table-column prop="nickname" label="姓名" width="120">  
 </el-table-column>  
 <el-table-column prop="email" label="邮箱">  
 </el-table-column>  
 <el-table-column prop="phone" label="电话">  
 </el-table-column>  
 <el-table-column prop="address" label="地址">  
 </el-table-column>  
 <el-table-column label="操作" width="300" align="center">  
 <template slot-scope="scope">  
 <el-button type="info" @click="handleDetail(scope.row.id)">查看详情</el-button>  
 </template>  
 </el-table-column>  
 </el-table>  
 <div style="padding: 10px 0">  
 <el-pagination  
 @size-change="handleSizeChange"  
 @current-change="handleCurrentChange"  
 :current-page="pageNum"  
 :page-sizes="[5, 10, 15, 20]"  
 :page-size="pageSize"  
 layout="total, sizes, prev, pager, next, jumper"  
 :total="total">  
 </el-pagination>  
 </div>  
 <el-dialog  
 title="学生详情"  
 :visible.sync="dialogFormVisible"  
 width="30%"  
 >  
 <el-form label-width="80px" size="small">  
 <el-form-item label="绩点">  
 <el-input v-model="form.gpa" auto-complete="off"></el-input>  
 </el-form-item>  
 <el-form-item label="自我介绍">  
 <el-input type="textarea" v-model="form.introduction" auto-complete="off" autosize></el-input>  
 </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="save">确定</el-button>  
 </div>  
 </el-dialog>  
 </div>  
</template>  
<script>  
export default {  
 name: "ChosenStudentInfo.vue",  
 data(){  
 return{  
 tableData: [],  
 pageNum: 1,  
 pageSize: 5,  
 username:"",  
 email:"",  
 address:"",  
 total:0,  
 dialogFormVisible:false,  
 form:{},  
 multipleSelection:[],  
 gpa:0,  
 introduction:"",  
 user:localStorage.getItem("user") ? JSON.parse(localStorage.getItem("user")) : {}  
 }  
 },  
 created() {  
 this.load()  
 },  
 methods:{  
 load() {  
 this.request.get("/user/page/ChosenStudent",{  
 params:{  
 id:this.user.id,  
 pageNum:this.pageNum,  
 pageSize:this.pageSize,  
 username:this.username,  
 email:this.email,  
 address:this.address,  
 }  
 }).then(res =>{  
 console.log(res)  
 this.tableData=res.data.records  
 this.total=res.data.total  
 })  
 },  
 save(){  
 this.request.post("/user",this.form).then(res=> {  
 if(res.data){  
 this.$message.success("保存成功")  
 this.dialogFormVisible=false  
 this.load()  
 }else {  
 this.$message.error("保存失败！")  
 }  
 })  
 },  
 handleAdd(){  
 this.dialogFormVisible=true  
 this.form={}  
 },  
 handleChoose(id){  
 this.request.get("/student/saveTeacherId/"+id,{  
 params:{  
 id:id,  
 tid:this.user.id  
 }  
 }).then(res=>{  
 if(res.data){  
 this.$message.success("选择成功！")  
 }else {  
 this.$message.error("选择失败！")  
 }  
 }  
 )  
 },  
 handleDetail(id){  
 this.request.get("/student/"+id).then(res=>{  
 if(res.data){  
 this.form=Object.assign({},res.data)  
 this.dialogFormVisible=true  
 }else {  
 this.$message.error("查询失败！学生没有填写任何信息！")  
 }  
 }  
 )  
 },  
 del(id){  
 this.request.get("/intention/push/"+id).then(res=>{  
 if(res.data){  
 this.$message.success("PUSH成功")  
 this.load()  
 }else {  
 this.$message.error("PUSH失败！")  
 }  
 })  
 },  
 handleSelectionChange(val){  
 console.log(val)  
 this.multipleSelection=val  
 },  
 delBatch(){  
 let ids=this.multipleSelection.map(v=>v.id)//把对象数组变成纯id数组  
 this.request.post("/user/del/batch",ids).then(res=>{  
 if(res.data){  
 this.$message.success("批量删除成功")  
 this.load()  
 }else {  
 this.$message.error("批量删除失败！")  
 }  
 })  
 },  
 reset(){  
 this.username=""  
 this.email=""  
 this.address=""  
 this.load()  
 },  
 handleSizeChange(pageSize){  
 console.log(pageSize)  
 this.pageSize=pageSize  
 this.load()  
 },  
 handleCurrentChange(pageNum){  
 console.log(pageNum)  
 this.pageNum=pageNum  
 this.load()  
 }  
 }  
}  
</script>  
<style scoped>  
</style>  
<template>  
 <div style="color: #666;font-size: 14px;">  
 <div style="padding-bottom: 20px">  
 <b>欢迎你！{{ user.nickname }}</b>  
 </div>  
 <div style="height: 1px; background: #ddd; margin: 20px 0"/>  
 </div>  
</template>  
<script>  
export default {  
 name: "Home",  
 data() {  
 return {  
 user: localStorage.getItem("user") ? JSON.parse(localStorage.getItem("user")) : {}  
 }  
 }  
}  
</script>  
<template>  
 <div>  
 <div style="padding: 10px 0">  
 <el-input style="width: 200px;margin-left: 5px" placeholder="请输入名称" suffix-icon="el-icon-search" v-model="username"></el-input>  
 <el-button style="margin-left: 5px" type="primary" icon="el-icon-search" @click="load">搜索</el-button>  
 <el-button style="margin-left: 5px" type="warning" @click="reset">重置</el-button>  
 </div>  
 <div style="margin: 10px 0">  
 <el-button type="primary" icon="el-icon-circle-plus" @click="handleAdd">新增</el-button>  
 <el-popconfirm  
 confirm-button-text='确定'  
 cancel-button-text='我再想想'  
 icon="el-icon-info"  
 icon-color="red"  
 title="您确定批量删除这条记录吗？"  
 @confirm="delBatch"  
 >  
 <el-button type="danger" icon="el-icon-remove-outline" slot="reference">批量删除</el-button>  
 </el-popconfirm>  
 <el-button type="primary" icon="el-icon-share" style="margin-left: 10px">导出</el-button>  
 <el-button type="primary">导入<i class="el-icon-upload el-icon--right"></i></el-button>  
 </div>  
 <el-table :data="tableData" border stripe @selection-change="handleSelectionChange">  
 <el-table-column  
 type="selection"  
 width="55"  
 >  
 </el-table-column>  
 <el-table-column prop="sequence" label="志愿" width="140">  
 </el-table-column>  
 <el-table-column prop="teacherName" label="教师" width="120">  
 </el-table-column>  
 <el-table-column label="操作" width="200" align="center">  
 <template slot-scope="scope">  
 <el-button type="info" @click="handleDetail(scope.row.teacherId)">查看详情</el-button>  
 <el-button type="success" icon="el-icon-upload2" slot="reference" style="margin-left: 5px" v-if="scope.row.sequence>1" @click="handleUp(scope.row.id,scope.row.sequence,scope.row.studentId)">{{scope.row.sequence}}上移</el-button>  
 </template>  
 </el-table-column>  
 </el-table>  
 <div style="padding: 10px 0">  
 <el-pagination  
 @size-change="handleSizeChange"  
 @current-change="handleCurrentChange"  
 :current-page="pageNum"  
 :page-sizes="[5, 10, 15, 20]"  
 :page-size="pageSize"  
 layout="total, sizes, prev, pager, next, jumper"  
 :total="total">  
 </el-pagination>  
 </div>  
 <el-dialog  
 title="导师详情"  
 :visible.sync="dialogFormVisible"  
 width="30%"  
 >  
 <el-form label-width="80px" size="small">  
 <el-form-item label="职称">  
 <el-input v-model="form.title" auto-complete="off"></el-input>  
 </el-form-item>  
 <el-form-item label="招生人数">  
 <el-input v-model="form.enrollment" auto-complete="off"></el-input>  
 </el-form-item>  
 <el-form-item label="要求">  
 <el-input type="textarea" v-model="form.requirement" auto-complete="off" autosize></el-input>  
 </el-form-item>  
 <el-form-item label="自我介绍">  
 <el-input type="textarea" v-model="form.introduction" auto-complete="off" autosize></el-input>  
 </el-form-item>  
 </el-form>  
 <div slot="footer" class="dialog-footer">  
 <el-button type="primary" @click="dialogFormVisible=false">确定</el-button>  
 </div>  
 </el-dialog>  
 </div>  
</template>  
<script>  
export default {  
 name: "Intention.vue",  
 data(){  
 return{  
 tableData: [],  
 pageNum: 1,  
 pageSize: 5,  
 username:"",  
 nickname:"",  
 email:"",  
 address:"",  
 total:0,  
 dialogFormVisible:false,  
 form:{},  
 multipleSelection:[],  
 title:"",  
 enrollment:0,  
 requirement:"",  
 introduction:"",  
 user:localStorage.getItem("user") ? JSON.parse(localStorage.getItem("user")) : {}  
 }  
 },  
 created() {  
 this.load()  
 },  
 methods:{  
 load() {  
 this.request.get("/intention/page",{  
 params:{  
 pageNum:this.pageNum,  
 pageSize:this.pageSize,  
 id:this.user.id,  
 }  
 }).then(res =>{  
 console.log(res)  
 this.tableData=res.data  
 this.total=res.data.total  
 })  
 },  
 save(){  
 this.request.post("/user",this.form).then(res=> {  
 if(res.data){  
 this.$message.success("保存成功")  
 this.dialogFormVisible=false  
 this.load()  
 }else {  
 this.$message.error("保存失败！")  
 }  
 })  
 },  
 handleAdd(){  
 this.dialogFormVisible=true  
 this.form={}  
 },  
 handleDetail(id){  
 this.request.get("/teacher/"+id).then(res=>{  
 if(res.data){  
 this.form=Object.assign({},res.data)  
 this.dialogFormVisible=true  
 }else {  
 this.$message.error("查询失败！老师没有留下任何信息！")  
 }  
 }  
 )  
 },  
 handleUp(id,sequence,studentId){  
 this.request.get("/intention/intentionUp",{  
 params:{  
 id:id,  
 studentId:studentId,  
 sequence:sequence,  
 }  
 }).then(res=>{  
 if(res.data){  
 this.$message("上移成功")  
 }else {  
 this.$message.error("上移失败")  
 }  
 })  
 this.load()  
 },  
 del(id){  
 this.request.delete("/user/"+id).then(res=>{  
 if(res.data){  
 this.$message.success("删除成功")  
 this.load()  
 }else {  
 this.$message.error("删除失败！")  
 }  
 })  
 },  
 handleSelectionChange(val){  
 console.log(val)  
 this.multipleSelection=val  
 },  
 delBatch(){  
 let ids=this.multipleSelection.map(v=>v.id)//把对象数组变成纯id数组  
 this.request.post("/user/del/batch",ids).then(res=>{  
 if(res.data){  
 this.$message.success("批量删除成功")  
 this.load()  
 }else {  
 this.$message.error("批量删除失败！")  
 }  
 })  
 },  
 reset(){  
 this.username=""  
 this.email=""  
 this.address=""  
 this.load()  
 },  
 handleSizeChange(pageSize){  
 console.log(pageSize)  
 this.pageSize=pageSize  
 this.load()  
 },  
 handleCurrentChange(pageNum){  
 console.log(pageNum)  
 this.pageNum=pageNum  
 this.load()  
 }  
 }  
}  
</script>  
<style scoped>  
</style>  
<template>  
 <div class="wrapper">  
 <div style="margin: 200px auto; background-color: #fff; width: 350px; height: 300px; padding: 20px; border-radius: 10px">  
 <div style="margin: 20px 0; text-align: center; font-size: 24px"><b>登 录</b></div>  
 <el-form :model="user" :rules="rules" ref="userForm">  
 <el-form-item prop="username">  
 <el-input size="medium" style="margin: 10px 0" prefix-icon="el-icon-user" v-model="user.username"></el-input>  
 </el-form-item>  
 <el-form-item prop="password">  
 <el-input size="medium" style="margin: 10px 0" prefix-icon="el-icon-lock" show-password v-model="user.password"></el-input>  
 </el-form-item>  
 <el-form-item style="margin: 10px 0; text-align: right">  
 <el-button type="primary" size="small" autocomplete="off" @click="login">登录</el-button>  
 <el-button type="warning" size="small" autocomplete="off" @click="$router.push('/register')">注册</el-button>  
 </el-form-item>  
 </el-form>  
 </div>  
 </div>  
</template>  
<script>  
export default {  
 name: "Login",  
 data() {  
 return {  
 user: {},  
 rules: {  
 username: [  
 { required: true, message: '请输入用户名', trigger: 'blur' },  
 { min: 1, max: 10, message: '长度在 1 到 10 个字符', trigger: 'blur' }  
 ],  
 password: [  
 { required: true, message: '请输入密码', trigger: 'blur' },  
 { min: 1, max: 20, message: '长度在 1 到 20 个字符', trigger: 'blur' }  
 ],  
 }  
 }  
 },  
 methods: {  
 login() {  
 this.$refs['userForm'].validate((valid) => {  
 if (valid) { // 表单校验合法  
 this.request.post("/user/login", this.user).then(res => {  
 if(res.code==='200') {  
 localStorage.setItem("user",JSON.stringify(res.data))//存储用户信息到浏览器  
 this.$router.push("/")  
 this.$message.success("登录成功")  
 } else {  
 this.$message.error(res.msg)  
 }  
 })  
 }  
 });  
 }  
 }  
}  
</script>  
<style>  
.wrapper {  
 height: 100vh;  
 background-image: linear-gradient(to bottom right, #FC466B , #3F5EFB);  
 overflow: hidden;  
}  
</style>  
<template>  
 <div style="height: 100%">  
 <el-container style="height: 100%">  
 <el-aside :width="sideWith" style="background-color: rgb(238, 241, 246);box-shadow:2px 0px 6px aqua" height="100%">  
 <Aside :isCollapse="isCollapse" :logoTextShow="logoTextShow" />  
 </el-aside>  
 <el-container>  
 <el-header style="border-bottom: 1px solid #ccc;">  
 <Header :collapseBtnClass="collapseBtnClass" @asideCollapse="collapse" />  
 </el-header>  
 <el-main>  
 <router-view />  
 </el-main>  
 </el-container>  
 </el-container>  
 </div>  
</template>  
<script>  
import Aside from "@/components/Aside";  
import Header from "@/components/Header";  
export default {  
 name: 'Home',  
 data() {  
 return {  
 collapseBtnClass:'el-icon-s-fold',  
 isCollapse:false,  
 sideWith:200,  
 logoTextShow:true,  
 user:localStorage.getItem("user") ? JSON.parse(localStorage.getItem("user")) : {}  
 }  
 },  
 components:{  
 Aside,  
 Header  
 }  
 ,  
 methods:{  
 collapse(){//点击收缩按钮收缩  
 this.isCollapse=!this.isCollapse  
 if (this.isCollapse) {  
 this.collapseBtnClass = 'el-icon-s-unfold';  
 this.sideWith = 64;  
 this.logoTextShow = false;  
 }  
 else{//展开  
 this.collapseBtnClass='el-icon-s-fold';  
 this.sideWith=200;  
 this.logoTextShow=true;  
 }  
 },  
 }  
}  
</script>  
<template>  
 <el-card style=" width: 500px;">  
 <el-form label-width="80px" size="small">  
 <el-form-item label="用户名">  
 <el-input v-model="form.username" disabled autocomplete="off"></el-input>  
 </el-form-item>  
 <el-form-item label="姓名">  
 <el-input v-model="form.nickname" disabled autocomplete="off"></el-input>  
 </el-form-item>  
 <el-form-item label="邮箱">  
 <el-input v-model="form.email" autocomplete="off"></el-input>  
 </el-form-item>  
 <el-form-item label="电话">  
 <el-input v-model="form.phone" autocomplete="off"></el-input>  
 </el-form-item>  
 <el-form-item label="地址">  
 <el-input type="textarea" v-model="form.address" autocomplete="off" autosize></el-input>  
 </el-form-item>  
 <el-form-item label="招生人数" v-if="user.role==='老师'" prop="enrollment">  
 <el-input-number v-model="extra.enrollment" autocomplete="off" :min="1" :max="6"></el-input-number>  
 </el-form-item>  
 <el-form-item label="要求" v-if="user.role==='老师'">  
 <el-input type="textarea" v-model="extra.requirement" autocomplete="off" autosize></el-input>  
 </el-form-item>  
 <el-form-item label="自我介绍">  
 <el-input type="textarea" v-model="extra.introduction" autocomplete="off" autosize></el-input>  
 </el-form-item>  
 <el-form-item label="导师id" v-if="user.role==='学生'">  
 <el-input v-model="extra.teacherId" disabled autocomplete="off"></el-input>  
 </el-form-item>  
 <el-form-item>  
 <el-button type="primary" @click="save">确 定</el-button>  
 </el-form-item>  
 </el-form>  
 </el-card>  
</template>  
<script>  
export default {  
 name: "Person",  
 data() {  
 return {  
 extra:{},  
 form: {},  
 user: localStorage.getItem("user") ? JSON.parse(localStorage.getItem("user")) : {}  
 }  
 },  
 created() {  
 this.getUser().then(res => {  
 console.log(res)  
 this.form = res  
 })  
 this.getUserInfo(this.user.id,this.user.role)  
 },  
 methods: {  
 async getUser() {  
 return (await this.request.get("/user/" + this.user.id)).data  
 },  
 getUserInfo(id, role) {  
 if (role === "老师") {  
 this.request.get("/teacher/" + id).then(res => {  
 if (res.data) {  
 this.extra = Object.assign({}, res.data)  
 } else {  
 this.$message.error("There is something wrong!")  
 }  
 }  
 )  
 } else if (role === "学生") {  
 this.request.get("/student/" + id).then(res => {  
 if (res.data) {  
 this.extra = Object.assign({}, res.data)  
 } else {  
 this.$message.error("There is something wrong!")  
 }  
 }  
 )  
 }  
 },  
 save() {  
 this.request.post("/user", this.form).then(res => {  
 if (res.code === '200') {  
 this.$message.success("保存成功")  
 this.$emit("refreshUser")  
 this.getUser().then(res => {  
 res.token = JSON.parse(localStorage.getItem("user")).token  
 localStorage.setItem("user", JSON.stringify(res))  
 })  
 } else {  
 this.$message.error("保存失败")  
 }  
 });  
 this.request.post("/teacher", this.extra);  
 this.request.post("/student", this.extra);  
 },  
 handleAvatarSuccess(res) {  
 this.form.avatarUrl = res  
 },  
 }  
}  
</script>  
<style scoped>  
</style>  
<template>  
 <div>  
 <el-timeline>  
 <el-timeline-item  
 v-for="(activity, index) in activities"  
 :key="index"  
 :icon="activity.icon"  
 :type="activity.type"  
 :color="activity.color"  
 :size="activity.size"  
 :timestamp="activity.timestamp">  
 {{activity.content}}  
 </el-timeline-item>  
 </el-timeline>  
 </div>  
</template>  
<script>  
export default {  
 name: "Progress",  
 data() {  
 return {  
 form: {  
 name: '',  
 region: '',  
 date1: '',  
 date2: '',  
 delivery: false,  
 type: [],  
 resource: '',  
 desc: ''  
 },  
 time:{},  
 activities: [{  
 content: '第三志愿截至,结果公布',  
 timestamp:'',  
 size: 'large',  
 type: 'primary',  
 icon: 'el-icon-top',  
 color: '#3112d9'  
 }, {  
 content: '第二志愿截至',  
 timestamp: '',  
 size: 'large',  
 icon: 'el-icon-top',  
 color: '#0bbd87'  
 }, {  
 content: '第一志愿截至',  
 timestamp: '',  
 icon: 'el-icon-top',  
 size: 'large',  
 color:'#d23232'  
 }, {  
 content: '学生志愿上报截至',  
 timestamp: '',  
 size: 'large',  
 icon: 'el-icon-top',  
 color:'#00e3e0'  
 }]  
 }  
 },  
 created() {  
 this.load()  
 },  
 methods:{  
 load(){  
 this.request.get("/deadline/"+1).then(res =>{  
 console.log(res)  
 if(res.code==='200') {  
 this.activities[2].timestamp=res.data.first  
 this.activities[1].timestamp=res.data.second  
 this.activities[0].timestamp=res.data.third  
 } else {  
 this.$message.error(res.msg)}  
 })  
 }  
 }  
}  
</script>  
<style scoped>  
</style>  
<template>  
 <div style="width: 100%; height: 100vh; overflow: hidden">  
 <div style="width: 400px; margin: 100px auto">  
 <div style="font-size: 30px; text-align: center; padding: 30px 0">欢迎注册</div>  
 <el-form ref="form" :model="form" size="normal" :rules="rules">  
 <el-form-item prop="username">  
 <el-input prefix-icon="el-icon-user-solid" v-model="form.username"></el-input>  
 </el-form-item>  
 <el-form-item prop="password">  
 <el-input prefix-icon="el-icon-lock" v-model="form.password" show-password></el-input>  
 </el-form-item>  
 <el-form-item prop="confirm">  
 <el-input prefix-icon="el-icon-lock" v-model="form.confirm" show-password></el-input>  
 </el-form-item>  
 <el-form-item>  
 <el-button style="width: 100%" type="primary" @click="register">注册</el-button>  
 </el-form-item>  
 <el-form-item><el-button type="text" @click="$router.push('/login')">&lt;&lt;返回登录 </el-button></el-form-item>  
 </el-form>  
 </div>  
 </div>  
</template>  
<script>  
import request from "@/utils/request";  
export default {  
 name: "Register",  
 data() {  
 return {  
 form: {},  
 rules: {  
 username: [  
 {required: true, message: '请输入用户名', trigger: 'blur'},  
 ],  
 password: [  
 {required: true, message: '请输入密码', trigger: 'blur'},  
 ],  
 confirm: [  
 {required: true, message: '请确认密码', trigger: 'blur'},  
 ],  
 }  
 }  
 },  
 methods: {  
 register() {  
 if (this.form.password !== this.form.confirm) {  
 this.$message({  
 type: "error",  
 message: '2次密码输入不一致！'  
 })  
 return  
 }  
 this.$refs['form'].validate((valid) => {  
 if (valid) {  
 request.post("/user/register", this.form).then(res => {  
 if (res.code === '0') {  
 this.$message({  
 type: "success",  
 message: "注册成功"  
 })  
 this.$router.push("/login") //登录成功之后进行页面的跳转，跳转到主页  
 } else {  
 this.$message({  
 type: "error",  
 message: res.msg  
 })  
 }  
 })  
 }  
 })  
 }  
 }  
}  
</script>  
<style scoped>  
</style>  
<template>  
 <div>  
 <el-table :data="tableData" border stripe @selection-change="handleSelectionChange">  
 <el-table-column prop="学号" label="学号" width="140">  
 </el-table-column>  
 <el-table-column prop="姓名" label="姓名" width="120">  
 </el-table-column>  
 <el-table-column prop="导师姓名" label="导师姓名">  
 </el-table-column>  
 <el-table-column prop="导师邮箱" label="导师邮箱">  
 </el-table-column>  
 </el-table>  
 <div style="padding: 10px 0">  
 <el-pagination  
 @size-change="handleSizeChange"  
 @current-change="handleCurrentChange"  
 :current-page="pageNum"  
 :page-sizes="[5, 10, 15, 20]"  
 :page-size="pageSize"  
 layout="total, sizes, prev, pager, next, jumper"  
 :total="total">  
 </el-pagination>  
 </div>  
 </div>  
</template>  
<script>  
export default {  
 name: "Result.vue",  
 data(){  
 return{  
 tableData: [],  
 pageNum: 1,  
 pageSize: 5,  
 username:"",  
 email:"",  
 total:0,  
 dialogFormVisible:false,  
 form:{},  
 multipleSelection:[]  
 }  
 },  
 created() {  
 this.load()  
 },  
 methods:{  
 load() {  
 this.request.get("/user/page/result",{  
 params:{  
 pageNum:this.pageNum,  
 pageSize:this.pageSize  
 }  
 }).then(res =>{  
 console.log(res)  
 this.tableData=res.data.records  
 this.total=res.data.total  
 })  
 },  
 handleSelectionChange(val){  
 console.log(val)  
 this.multipleSelection=val  
 },  
 handleSizeChange(pageSize){  
 console.log(pageSize)  
 this.pageSize=pageSize  
 this.load()  
 },  
 handleCurrentChange(pageNum){  
 console.log(pageNum)  
 this.pageNum=pageNum  
 this.load()  
 }  
 }  
}  
</script>  
<style scoped>  
</style>  
<template>  
 <div>  
 <div style="padding: 10px 0">  
 <el-input style="width: 200px;margin-left: 5px" placeholder="请输入名称" suffix-icon="el-icon-search" v-model="username"></el-input>  
 <el-button style="margin-left: 5px" type="primary" icon="el-icon-search" @click="load">搜索</el-button>  
 <el-button style="margin-left: 5px" type="warning" @click="reset">重置</el-button>  
 </div>  
 <div style="margin: 10px 0">  
 </div>  
 <el-table :data="tableData" border stripe @selection-change="handleSelectionChange">  
 <el-table-column  
 type="selection"  
 width="55"  
 >  
 </el-table-column>  
 <el-table-column prop="id" label="ID" width="80">  
 </el-table-column>  
 <el-table-column prop="username" label="学号" width="140">  
 </el-table-column>  
 <el-table-column prop="nickname" label="姓名" width="120">  
 </el-table-column>  
 <el-table-column prop="email" label="邮箱">  
 </el-table-column>  
 <el-table-column prop="phone" label="电话">  
 </el-table-column>  
 <el-table-column prop="address" label="地址">  
 </el-table-column>  
 <el-table-column label="操作" width="300" align="center">  
 <template slot-scope="scope">  
 <el-button type="info" @click="handleDetail(scope.row.id)">查看详情</el-button>  
 <el-popconfirm  
 confirm-button-text='确定'  
 cancel-button-text='我再想想'  
 icon="el-icon-info"  
 icon-color="#67C23A"  
 title="确定选择这位学生嘛？"  
 @confirm="handleChoose(scope.row.id)"  
 >  
 <el-button type="success" icon="el-icon-check" slot="reference" style="margin-left: 5px" >选择</el-button>  
 </el-popconfirm>  
 <el-popconfirm  
 confirm-button-text='确定'  
 cancel-button-text='我再想想'  
 icon="el-icon-info"  
 icon-color="#67C23A"  
 title="确定不要这位学生嘛？"  
 @confirm="del(scope.row.id)"  
 >  
 <el-button type="danger" icon="el-icon-delete-solid" slot="reference" style="margin-left: 5px" >不要</el-button>  
 </el-popconfirm>  
 </template>  
 </el-table-column>  
 </el-table>  
 <div style="padding: 10px 0">  
 <el-pagination  
 @size-change="handleSizeChange"  
 @current-change="handleCurrentChange"  
 :current-page="pageNum"  
 :page-sizes="[5, 10, 15, 20]"  
 :page-size="pageSize"  
 layout="total, sizes, prev, pager, next, jumper"  
 :total="total">  
 </el-pagination>  
 </div>  
 <el-dialog  
 title="学生详情"  
 :visible.sync="dialogFormVisible"  
 width="30%"  
 >  
 <el-form label-width="80px" size="small">  
 <el-form-item label="绩点">  
 <el-input v-model="form.gpa" auto-complete="off"></el-input>  
 </el-form-item>  
 <el-form-item label="自我介绍">  
 <el-input type="textarea" v-model="form.introduction" auto-complete="off" autosize></el-input>  
 </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="save">确定</el-button>  
 </div>  
 </el-dialog>  
 </div>  
</template>  
<script>  
export default {  
 name: "StudentInfo.vue",  
 data(){  
 return{  
 tableData: [],  
 pageNum: 1,  
 pageSize: 5,  
 username:"",  
 email:"",  
 address:"",  
 total:0,  
 dialogFormVisible:false,  
 form:{},  
 multipleSelection:[],  
 gpa:0,  
 introduction:"",  
 user:localStorage.getItem("user") ? JSON.parse(localStorage.getItem("user")) : {}  
 }  
 },  
 created() {  
 this.load()  
 },  
 methods:{  
 load() {  
 this.request.get("/user/page/student",{  
 params:{  
 id:this.user.id,  
 pageNum:this.pageNum,  
 pageSize:this.pageSize,  
 username:this.username,  
 email:this.email,  
 address:this.address,  
 }  
 }).then(res =>{  
 console.log(res)  
 this.tableData=res.data.records  
 this.total=res.data.total  
 })  
 },  
 save(){  
 this.request.post("/user",this.form).then(res=> {  
 if(res.data){  
 this.$message.success("保存成功")  
 this.dialogFormVisible=false  
 this.load()  
 }else {  
 this.$message.error("保存失败！")  
 }  
 })  
 },  
 handleAdd(){  
 this.dialogFormVisible=true  
 this.form={}  
 },  
 handleChoose(id){  
 this.request.get("/student/saveTeacherId/"+id,{  
 params:{  
 id:id,  
 tid:this.user.id  
 }  
 }).then(res=>{  
 if(res.data){  
 this.$message.success("选择成功！")  
 }else {  
 this.$message.error("选择失败！")  
 }  
 }  
 )  
 },  
 handleDetail(id){  
 this.request.get("/student/"+id).then(res=>{  
 if(res.data){  
 this.form=Object.assign({},res.data)  
 this.dialogFormVisible=true  
 }else {  
 this.$message.error("查询失败！学生没有填写任何信息！")  
 }  
 }  
 )  
 },  
 del(id){  
 this.request.get("/intention/push/"+id).then(res=>{  
 if(res.data){  
 this.$message.success("PUSH成功")  
 this.load()  
 }else {  
 this.$message.error("PUSH失败！")  
 }  
 })  
 },  
 handleSelectionChange(val){  
 console.log(val)  
 this.multipleSelection=val  
 },  
 delBatch(){  
 let ids=this.multipleSelection.map(v=>v.id)//把对象数组变成纯id数组  
 this.request.post("/user/del/batch",ids).then(res=>{  
 if(res.data){  
 this.$message.success("批量删除成功")  
 this.load()  
 }else {  
 this.$message.error("批量删除失败！")  
 }  
 })  
 },  
 reset(){  
 this.username=""  
 this.email=""  
 this.address=""  
 this.load()  
 },  
 handleSizeChange(pageSize){  
 console.log(pageSize)  
 this.pageSize=pageSize  
 this.load()  
 },  
 handleCurrentChange(pageNum){  
 console.log(pageNum)  
 this.pageNum=pageNum  
 this.load()  
 }  
 }  
}  
</script>  
<style scoped>  
</style>  
<template>  
</template>  
<script>  
export default {  
 name: "Teacher"  
}  
</script>  
<style scoped>  
</style>  
<template>  
 <div>  
 <div style="padding: 10px 0">  
 <el-input style="width: 200px;margin-left: 5px" placeholder="请输入名称" suffix-icon="el-icon-search" v-model="username"></el-input>  
 <el-button style="margin-left: 5px" type="primary" icon="el-icon-search" @click="load">搜索</el-button>  
 <el-button style="margin-left: 5px" type="warning" @click="reset">重置</el-button>  
 </div>  
 <div style="margin: 10px 0">  
 <el-button type="primary" icon="el-icon-circle-plus" @click="handleAdd">新增</el-button>  
 <el-popconfirm  
 confirm-button-text='确定'  
 cancel-button-text='我再想想'  
 icon="el-icon-info"  
 icon-color="red"  
 title="您确定批量删除这条记录吗？"  
 @confirm="delBatch"  
 >  
 <el-button type="danger" icon="el-icon-remove-outline" slot="reference">批量删除</el-button>  
 </el-popconfirm>  
 <el-button type="primary" icon="el-icon-share" style="margin-left: 10px">导出</el-button>  
 <el-button type="primary">导入<i class="el-icon-upload el-icon--right"></i></el-button>  
 </div>  
 <el-table :data="tableData" border stripe @selection-change="handleSelectionChange">  
 <el-table-column  
 type="selection"  
 width="55"  
 >  
 </el-table-column>  
 <el-table-column prop="id" label="ID" width="80">  
 </el-table-column>  
 <el-table-column prop="username" label="工号" width="140">  
 </el-table-column>  
 <el-table-column prop="nickname" label="姓名" width="120">  
 </el-table-column>  
 <el-table-column prop="email" label="邮箱">  
 </el-table-column>  
 <el-table-column prop="phone" label="电话">  
 </el-table-column>  
 <el-table-column prop="address" label="地址">  
 </el-table-column>  
 <el-table-column label="操作" width="200" align="center">  
 <template slot-scope="scope">  
 <el-button type="info" @click="handleDetail(scope.row.id)">查看详情</el-button>  
 <el-popconfirm  
 confirm-button-text='确定'  
 cancel-button-text='我再想想'  
 icon="el-icon-info"  
 icon-color="#67C23A"  
 title="确定选择这位导师嘛？"  
 @confirm="addIntention(scope.row.id)"  
 >  
 <el-button type="success" icon="el-icon-check" slot="reference" style="margin-left: 5px" v-if="user.role==='学生'">选择</el-button>  
 </el-popconfirm>  
 </template>  
 </el-table-column>  
 </el-table>  
 <div style="padding: 10px 0">  
 <el-pagination  
 @size-change="handleSizeChange"  
 @current-change="handleCurrentChange"  
 :current-page="pageNum"  
 :page-sizes="[5, 10, 15, 20]"  
 :page-size="pageSize"  
 layout="total, sizes, prev, pager, next, jumper"  
 :total="total">  
 </el-pagination>  
 </div>  
 <el-dialog  
 title="导师详情"  
 :visible.sync="dialogFormVisible"  
 width="30%"  
 >  
 <el-form label-width="80px" size="small">  
 <el-form-item label="职称">  
 <el-input v-model="form.title" auto-complete="off"></el-input>  
 </el-form-item>  
 <el-form-item label="招生人数">  
 <el-input v-model="form.enrollment" auto-complete="off"></el-input>  
 </el-form-item>  
 <el-form-item label="要求">  
 <el-input type="textarea" v-model="form.requirement" auto-complete="off" autosize></el-input>  
 </el-form-item>  
 <el-form-item label="自我介绍">  
 <el-input type="textarea" v-model="form.introduction" auto-complete="off" autosize></el-input>  
 </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="save">确定</el-button>  
 </div>  
 </el-dialog>  
 </div>  
</template>  
<script>  
export default {  
 name: "TeacherInfo.vue",  
 data(){  
 return{  
 tableData: [],  
 pageNum: 1,  
 pageSize: 5,  
 username:"",  
 email:"",  
 address:"",  
 total:0,  
 dialogFormVisible:false,  
 form:{},  
 multipleSelection:[],  
 title:"",  
 enrollment:0,  
 requirement:"",  
 introduction:"",  
 intention:{},  
 user:localStorage.getItem("user") ? JSON.parse(localStorage.getItem("user")) : {}  
 }  
 },  
 created() {  
 this.load()  
 },  
 methods:{  
 load() {  
 this.request.get("/user/page/teacher",{  
 params:{  
 pageNum:this.pageNum,  
 pageSize:this.pageSize,  
 username:this.username,  
 email:this.email,  
 address:this.address,  
 }  
 }).then(res =>{  
 console.log(res)  
 this.tableData=res.data.records  
 this.total=res.data.total  
 })  
 },  
 save(){  
 this.request.post("/user",this.form).then(res=> {  
 if(res.data){  
 this.$message.success("保存成功")  
 this.dialogFormVisible=false  
 this.load()  
 }else {  
 this.$message.error("保存失败！")  
 }  
 })  
 },  
 handleAdd(){  
 this.dialogFormVisible=true  
 this.form={}  
 },  
 handleDetail(id){  
 this.request.get("/teacher/"+id).then(res=>{  
 if(res.data){  
 this.form=Object.assign({},res.data)  
 this.dialogFormVisible=true  
 }else {  
 this.$message.error("查询失败！老师没有留下任何信息！")  
 }  
 }  
 )  
 },  
 addIntention(teacher\_id){  
 this.request.get("/intention/addIntention",{  
 params:{  
 student\_id:this.user.id,  
 teacher\_id:teacher\_id  
 }  
 }).then(res=> {  
 if(res.data){  
 this.$message.success("添加志愿成功")  
 this.load()  
 }else {  
 this.$message.error("选择失败！")  
 }  
 })  
 },  
 del(id){  
 this.request.delete("/user/"+id).then(res=>{  
 if(res.data){  
 this.$message.success("删除成功")  
 this.load()  
 }else {  
 this.$message.error("删除失败！")  
 }  
 })  
 },  
 handleSelectionChange(val){  
 console.log(val)  
 this.multipleSelection=val  
 },  
 delBatch(){  
 let ids=this.multipleSelection.map(v=>v.id)//把对象数组变成纯id数组  
 this.request.post("/user/del/batch",ids).then(res=>{  
 if(res.data){  
 this.$message.success("批量删除成功")  
 this.load()  
 }else {  
 this.$message.error("批量删除失败！")  
 }  
 })  
 },  
 reset(){  
 this.username=""  
 this.email=""  
 this.address=""  
 this.load()  
 },  
 handleSizeChange(pageSize){  
 console.log(pageSize)  
 this.pageSize=pageSize  
 this.load()  
 },  
 handleCurrentChange(pageNum){  
 console.log(pageNum)  
 this.pageNum=pageNum  
 this.load()  
 }  
 }  
}  
</script>  
<style scoped>  
</style>  
<template>  
 <div>  
 <div style="padding: 10px 0">  
 <el-input style="width: 200px;margin-left: 5px" placeholder="请输入名称" suffix-icon="el-icon-search" v-model="username"></el-input>  
 <el-input style="width: 200px;margin-left: 5px" placeholder="请输入邮箱" suffix-icon="el-icon-message" v-model="email"></el-input>  
 <el-input style="width: 200px;margin-left: 5px" placeholder="请输入地址" suffix-icon="el-icon-position" v-model="address"></el-input>  
 <el-button style="margin-left: 5px" type="primary" icon="el-icon-search" @click="load">搜索</el-button>  
 <el-button style="margin-left: 5px" type="warning" @click="reset">重置</el-button>  
 </div>  
 <div style="margin: 10px 0">  
 <el-button type="primary" icon="el-icon-circle-plus" @click="handleAdd">新增</el-button>  
 <el-popconfirm  
 confirm-button-text='确定'  
 cancel-button-text='我再想想'  
 icon="el-icon-info"  
 icon-color="red"  
 title="您确定批量删除这条记录吗？"  
 @confirm="delBatch"  
 >  
 <el-button type="danger" icon="el-icon-remove-outline" slot="reference">批量删除</el-button>  
 </el-popconfirm>  
 <el-button type="primary" icon="el-icon-share" style="margin-left: 10px">导出</el-button>  
 <el-button type="primary">导入<i class="el-icon-upload el-icon--right"></i></el-button>  
 </div>  
 <el-table :data="tableData" border stripe @selection-change="handleSelectionChange">  
 <el-table-column  
 type="selection"  
 width="55"  
 >  
 </el-table-column>  
 <el-table-column prop="id" label="ID" width="80">  
 </el-table-column>  
 <el-table-column prop="username" label="用户名" width="140">  
 </el-table-column>  
 <el-table-column prop="nickname" label="昵称" width="120">  
 </el-table-column>  
 <el-table-column prop="email" label="邮箱">  
 </el-table-column>  
 <el-table-column prop="phone" label="电话">  
 </el-table-column>  
 <el-table-column prop="address" label="地址">  
 </el-table-column>  
 <el-table-column label="操作" width="200" align="center">  
 <template slot-scope="scope">  
 <el-button type="success" icon="el-icon-edit" @click="handleEdit(scope.row)">编辑</el-button>  
 <el-popconfirm  
 confirm-button-text='确定'  
 cancel-button-text='我再想想'  
 icon="el-icon-info"  
 icon-color="red"  
 title="确定删除这条记录吗？"  
 @confirm="del(scope.row.id)"  
 >  
 <el-button type="danger" icon="el-icon-delete" slot="reference" style="margin-left: 5px">删除</el-button>  
 </el-popconfirm>  
 </template>  
 </el-table-column>  
 </el-table>  
 <div style="padding: 10px 0">  
 <el-pagination  
 @size-change="handleSizeChange"  
 @current-change="handleCurrentChange"  
 :current-page="pageNum"  
 :page-sizes="[5, 10, 15, 20]"  
 :page-size="pageSize"  
 layout="total, sizes, prev, pager, next, jumper"  
 :total="total">  
 </el-pagination>  
 </div>  
 <el-dialog  
 title="用户信息"  
 :visible.sync="dialogFormVisible"  
 width="30%"  
 >  
 <el-form label-width="80px" size="small">  
 <el-form-item label="用户名">  
 <el-input v-model="form.username" auto-complete="off"></el-input>  
 </el-form-item>  
 <el-form-item label="昵称">  
 <el-input v-model="form.nickname" auto-complete="off"></el-input>  
 </el-form-item>  
 <el-form-item label="邮箱">  
 <el-input v-model="form.email" auto-complete="off"></el-input>  
 </el-form-item>  
 <el-form-item label="电话">  
 <el-input v-model="form.phone" auto-complete="off"></el-input>  
 </el-form-item>  
 <el-form-item label="地址">  
 <el-input v-model="form.address" auto-complete="off"></el-input>  
 </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="save">确定</el-button>  
 </div>  
 </el-dialog>  
 </div>  
</template>  
<script>  
export default {  
 name: "User.vue",  
 data(){  
 return{  
 tableData: [],  
 pageNum: 1,  
 pageSize: 5,  
 username:"",  
 email:"",  
 address:"",  
 total:0,  
 dialogFormVisible:false,  
 form:{},  
 multipleSelection:[]  
 }  
 },  
 created() {  
 this.load()  
 },  
 methods:{  
 load() {  
 this.request.get("/user/page",{  
 params:{  
 pageNum:this.pageNum,  
 pageSize:this.pageSize,  
 username:this.username,  
 email:this.email,  
 address:this.address,  
 }  
 }).then(res =>{  
 console.log(res)  
 this.tableData=res.data.records  
 this.total=res.data.total  
 })  
 },  
 save(){  
 this.request.post("/user",this.form).then(res=> {  
 if(res.data){  
 this.$message.success("保存成功")  
 this.dialogFormVisible=false  
 this.load()  
 }else {  
 this.$message.error("保存失败！")  
 }  
 })  
 },  
 handleAdd(){  
 this.dialogFormVisible=true  
 this.form={}  
 },  
 handleEdit(row){  
 this.form=Object.assign({},row)  
 this.dialogFormVisible=true  
 },  
 del(id){  
 this.request.delete("/user/"+id).then(res=>{  
 if(res.data){  
 this.$message.success("删除成功")  
 this.load()  
 }else {  
 this.$message.error("删除失败！")  
 }  
 })  
 },  
 handleSelectionChange(val){  
 console.log(val)  
 this.multipleSelection=val  
 },  
 delBatch(){  
 let ids=this.multipleSelection.map(v=>v.id)//把对象数组变成纯id数组  
 this.request.post("/user/del/batch",ids).then(res=>{  
 if(res.data){  
 this.$message.success("批量删除成功")  
 this.load()  
 }else {  
 this.$message.error("批量删除失败！")  
 }  
 })  
 },  
 reset(){  
 this.username=""  
 this.email=""  
 this.address=""  
 this.load()  
 },  
 handleSizeChange(pageSize){  
 console.log(pageSize)  
 this.pageSize=pageSize  
 this.load()  
 },  
 handleCurrentChange(pageNum){  
 console.log(pageNum)  
 this.pageNum=pageNum  
 this.load()  
 }  
 }  
}  
</script>  
<style scoped>  
</style>