编号：

2020 - 2021学年第 二 学期



**实 验 报 告**

实验课程名称 软件架构与应用开发

专 业 班 级 计算机1803

学 生 姓 号 31801150

学 生 姓 名 张帅

实验指导教师 彭彬

一. 实验目的和要求

1.学习本节示例L08RESTful

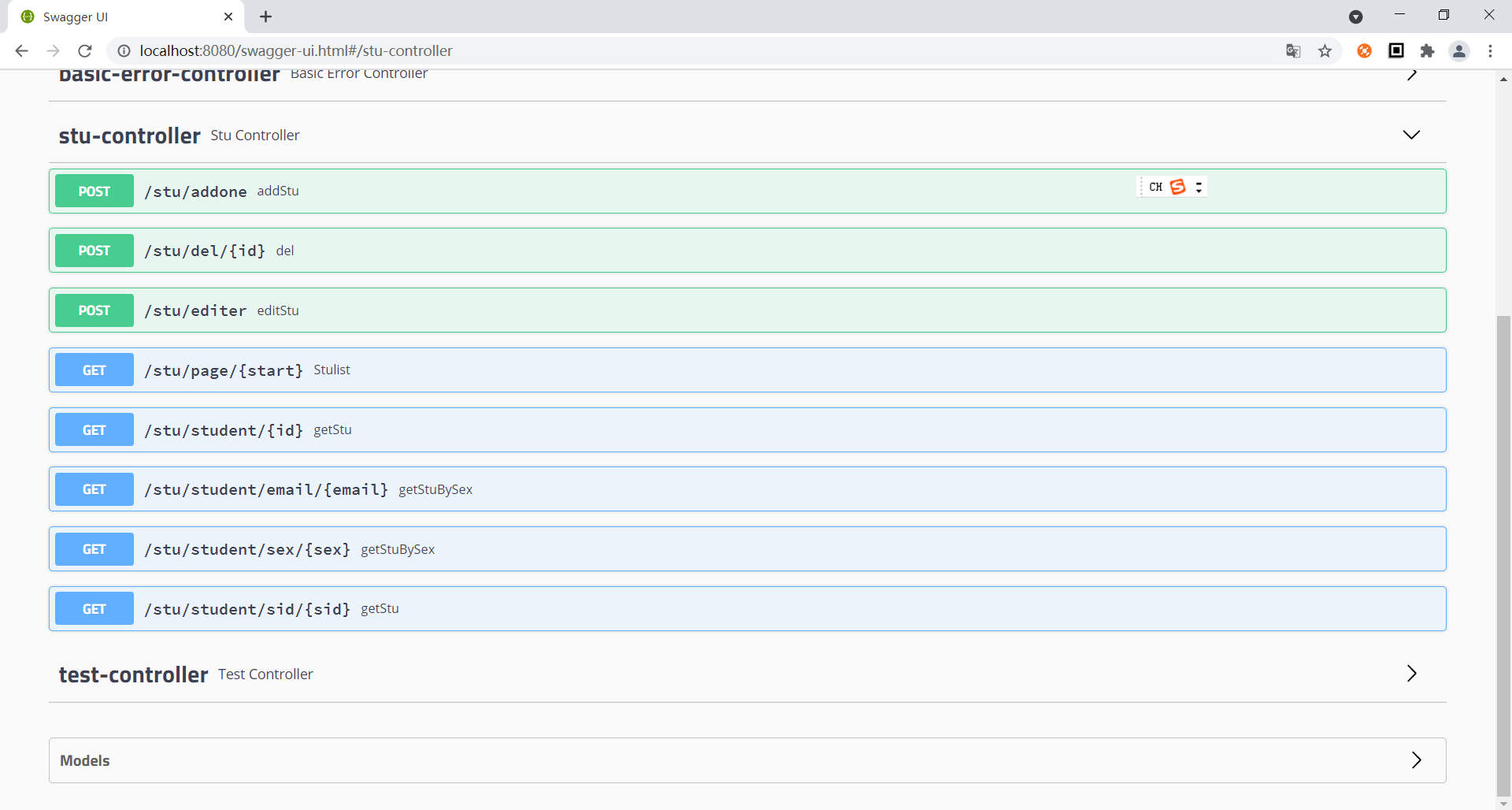
2.研读钉钉或者微信等的公开平台的API文档，熟悉API风格

3.将学生信息系统中的接口进行RESTful化的设计及改造（最少包括增删改、不少于4种查询），并且使用Swagger生成API文档

二. 实验内容

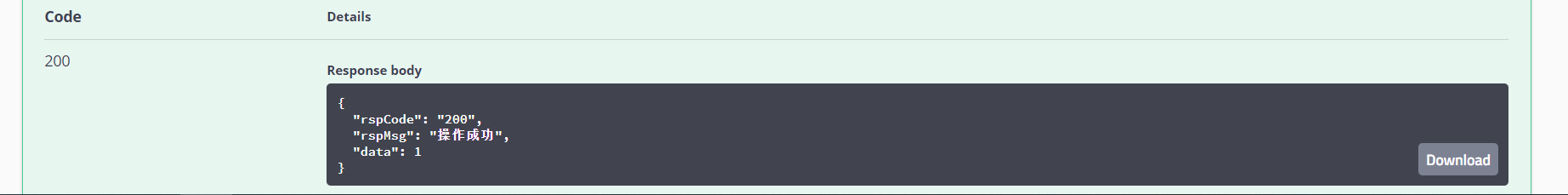
3.将学生信息系统中的接口进行RESTful化的设计及改造（最少包括增删改、不少于4种查询），并且使用Swagger生成API文档

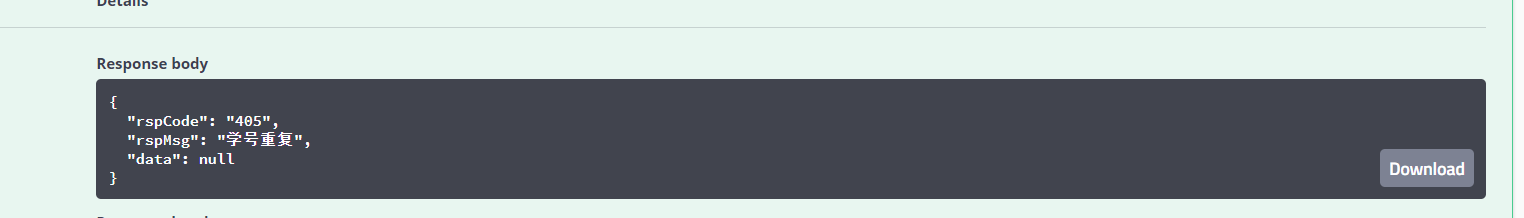
Swaggest文档：

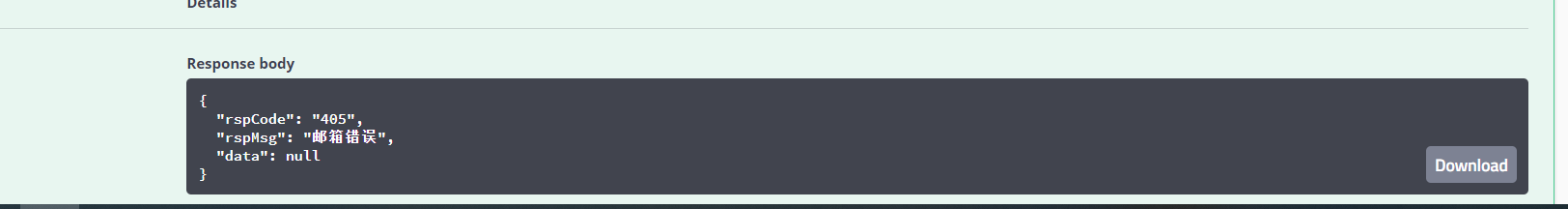
  
RESTful化：

@RestController  
@RequestMapping("stu")  
public class StuController {  
  
 protected Response result(ExceptionMsg msg){  
 return new Response(msg);  
 }  
 protected Response result(){  
 return new Response();  
 }  
  
  
 @Autowired  
 private StuMapper stuMapper;  
  
 @GetMapping("/page/{start}")  
 public ResponseData Stulist(@PathVariable("start") Integer start) {  
  
 PageHelper.*startPage*(start,5,"id desc");  
 List<Stu> cs = stuMapper.queryAll();  
 PageInfo<Stu> page = new PageInfo<>(cs);  
  
 if(page.getPages()<start){  
 return new ResponseData(new Response("405","该页码数据不存在"),null);  
 }  
  
 return new ResponseData(ExceptionMsg.*SUCCESS*,page);  
 }  
  
 @GetMapping("/student/{id}")  
 public ResponseData getStu(@PathVariable("id") Integer id) throws Exception {  
 Stu stu = stuMapper.queryById(id);  
 if(stu == null){  
 return new ResponseData(new Response("405","该学生不存在"),stu);  
 }  
 return new ResponseData(ExceptionMsg.*SUCCESS*,stu);  
  
 }  
  
 @GetMapping("/student/sid/{sid}")  
 public ResponseData getStu(@PathVariable("sid") String sid) throws Exception {  
 Stu stu = stuMapper.queryBySid(sid);  
 if(stu == null){  
 return new ResponseData(new Response("405","该学生不存在"),stu);  
 }  
 return new ResponseData(ExceptionMsg.*SUCCESS*,stu);  
  
 }  
  
 @GetMapping("/student/sex/{sex}")  
 public ResponseData getStuBySex(@PathVariable("sex") Integer sex) throws Exception {  
 List<Stu> stu = stuMapper.queryBySex(sex);  
 if(sex!=0&&sex!=1)  
 {  
 return new ResponseData(new Response("405","该性别不存在"),stu);  
 }  
 else if(stu == null){  
 return new ResponseData(new Response("405","该类学生不存在"),stu);  
 }  
 return new ResponseData(ExceptionMsg.*SUCCESS*,stu);  
  
 }  
  
 @GetMapping("/student/email/{email}")  
 public ResponseData getStuBySex(@PathVariable("email") String email) throws Exception {  
 List<Stu> stu = stuMapper.queryByEmail(email);  
  
 if(stu.size()==0){  
 return new ResponseData(new Response("405","该学生不存在"),stu);  
 }  
 return new ResponseData(ExceptionMsg.*SUCCESS*,stu);  
  
 }  
  
 @PostMapping("/del/{id}")  
 public ResponseData del(@PathVariable("id") Integer id) throws Exception{  
 Stu stu = stuMapper.queryById(id);  
 if(stu == null){  
 return new ResponseData(new Response("405","该学生不存在"),null);  
 }  
  
 int returnId = stuMapper.delById(id);  
 return new ResponseData(ExceptionMsg.*SUCCESS*,returnId);  
 }  
  
  
 @PostMapping("/editer")  
 public ResponseData editStu(@RequestBody Stu stu) throws Exception {  
  
 Stu temp = stuMapper.queryById((int)stu.getId());  
 if(temp==null ){  
 return new ResponseData(new Response("405","该学生不存在"),null);  
 }else if( stuMapper.queryBySid(stu.getSid()) != null && stuMapper.queryBySid(stu.getSid()).getId() != stu.getId()){  
 return new ResponseData(new Response("405","学号重复"),null);  
 }else if( stu.getSid() == null ){  
 return new ResponseData(new Response("405","学号不能为空"),null);  
 }else if( stu.getName() == null ){  
 return new ResponseData(new Response("405","姓名不能为空"),null);  
 }else if( stu.getName().length() <1 || stu.getName().length() >6){  
 return new ResponseData(new Response("405","姓名长度必须为1-6位"),null);  
 }else if( stu.getSex()==null){  
 return new ResponseData(new Response("405","性别不能为空"),null);  
 }else if( stu.getSex()!=1&&stu.getSex()!=0){  
 return new ResponseData(new Response("405","性别错误"),null);  
 }else if( stu.getAge()==null){  
 return new ResponseData(new Response("405","年龄不能为空"),null);  
 }else if( stu.getAge()<0 || stu.getAge()>100){  
 return new ResponseData(new Response("405","年龄错误"),null);  
 }else if( stu.getEmail()==null){  
 return new ResponseData(new Response("405","邮箱不能为空"),null);  
 }else if( !stu.getEmail().matches("^\\w+([-+.]\\w+)\*@\\w+([-.]\\w+)\*\\.\\w+([-.]\\w+)\*$")){  
 return new ResponseData(new Response("405","邮箱错误"),null);  
 }else if( stu.getCard\_id()==null){  
 return new ResponseData(new Response("405","身份证不能为空"),null);  
 }else if( !stu.getCard\_id().matches("(^[1-9]\\d{5}(19|([23]\\d))\\d{2}((0[1-9])|(10|11|12))(([0-2][1-9])|10|20|30|31)\\d{3}[0-9Xx]$)|(^[1-9]\\d{5}\\d{2}((0[1-9])|(10|11|12))(([0-2][1-9])|10|20|30|31)\\d{2}[0-9Xx]$)")){  
 return new ResponseData(new Response("405","身份证错误"),null);  
 }else if( stu.getText()==null){  
 return new ResponseData(new Response("405","简历不能为空"),null);  
 }  
  
 int returnId = stuMapper.updateById(stu);  
 return new ResponseData(ExceptionMsg.*SUCCESS*,returnId);  
  
 }  
  
  
 @PostMapping("/addone")  
 public ResponseData addStu(@RequestBody Stu stu) throws Exception {  
  
  
 if( stuMapper.queryBySid(stu.getSid()) != null){  
 return new ResponseData(new Response("405","学号重复"),null);  
 }else if( stu.getSid() == null ){  
 return new ResponseData(new Response("405","学号不能为空"),null);  
 }else if( stu.getName() == null ){  
 return new ResponseData(new Response("405","姓名不能为空"),null);  
 }else if( stu.getName().length() <1 || stu.getName().length() >6){  
 return new ResponseData(new Response("405","姓名长度必须为1-6位"),null);  
 }else if( stu.getSex()==null){  
 return new ResponseData(new Response("405","性别不能为空"),null);  
 }else if( stu.getSex()!=1&&stu.getSex()!=0){  
 return new ResponseData(new Response("405","性别错误"),null);  
 }else if( stu.getAge()==null){  
 return new ResponseData(new Response("405","年龄不能为空"),null);  
 }else if( stu.getAge()<0 || stu.getAge()>100){  
 return new ResponseData(new Response("405","年龄错误"),null);  
 }else if( stu.getEmail()==null){  
 return new ResponseData(new Response("405","邮箱不能为空"),null);  
 }else if( !stu.getEmail().matches("^\\w+([-+.]\\w+)\*@\\w+([-.]\\w+)\*\\.\\w+([-.]\\w+)\*$")){  
 return new ResponseData(new Response("405","邮箱错误"),null);  
 }else if( stu.getCard\_id()==null){  
 return new ResponseData(new Response("405","身份证不能为空"),null);  
 }else if( !stu.getCard\_id().matches("(^[1-9]\\d{5}(19|([23]\\d))\\d{2}((0[1-9])|(10|11|12))(([0-2][1-9])|10|20|30|31)\\d{3}[0-9Xx]$)|(^[1-9]\\d{5}\\d{2}((0[1-9])|(10|11|12))(([0-2][1-9])|10|20|30|31)\\d{2}[0-9Xx]$)")){  
 return new ResponseData(new Response("405","身份证错误"),null);  
 }else if( stu.getText()==null){  
 return new ResponseData(new Response("405","简历不能为空"),null);  
 }  
  
 int returnId = stuMapper.add(stu);  
 return new ResponseData(ExceptionMsg.*SUCCESS*,returnId);  
  
 }  
  
  
  
}

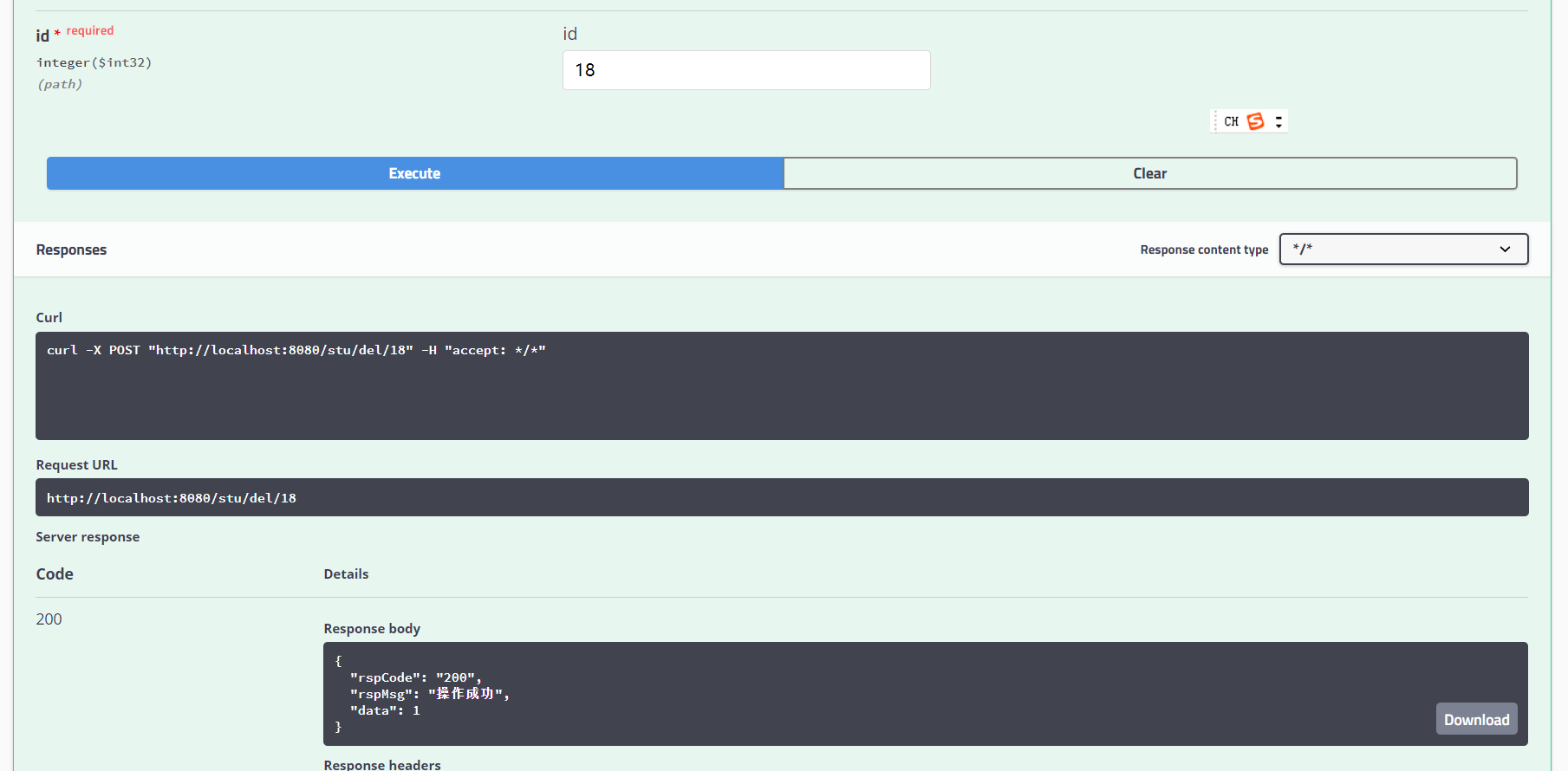
增加（错误检测）：

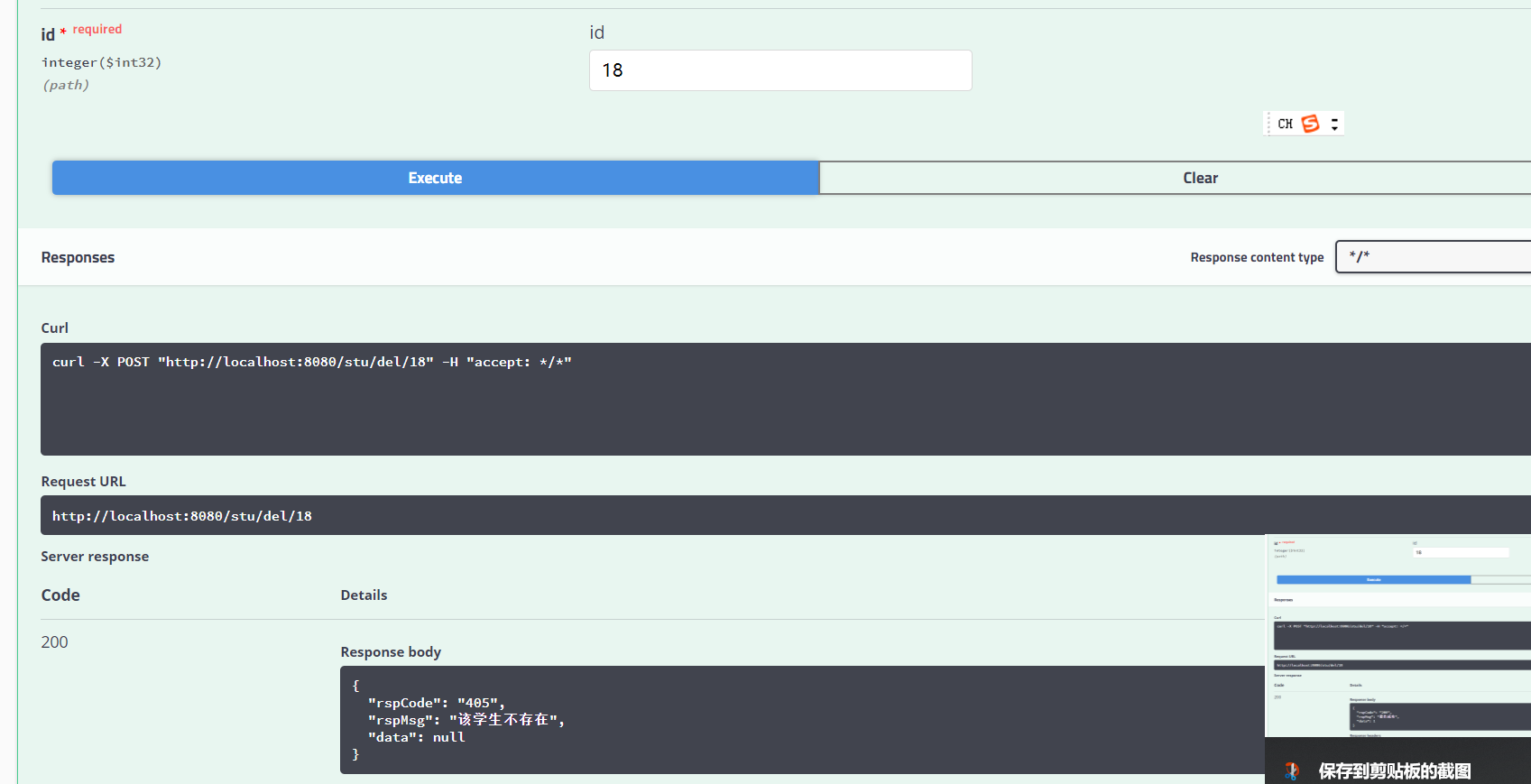




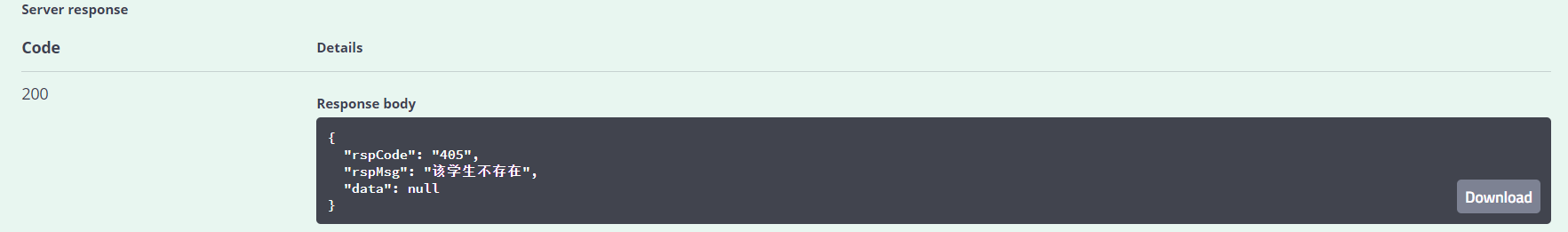


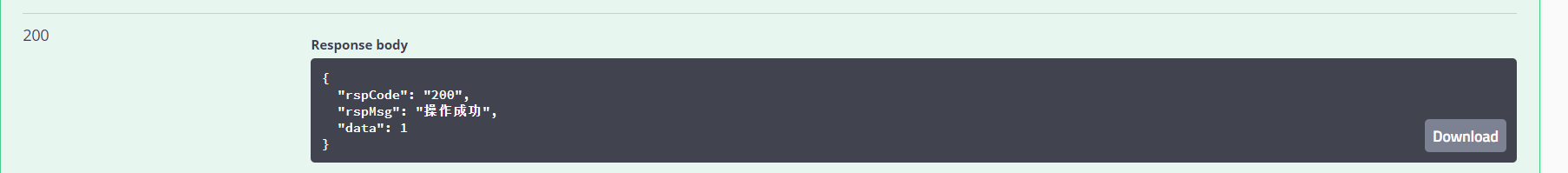
删：





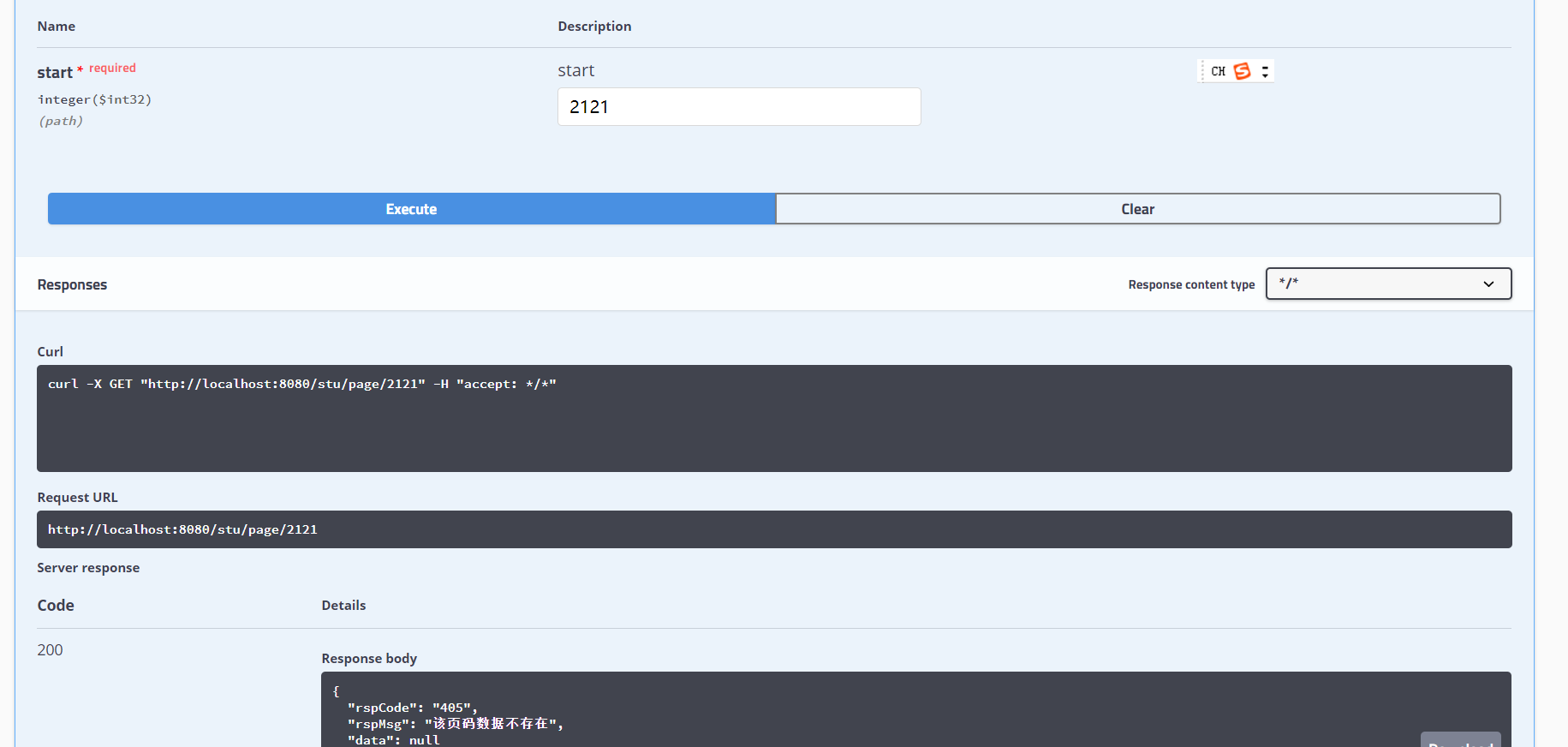
改：



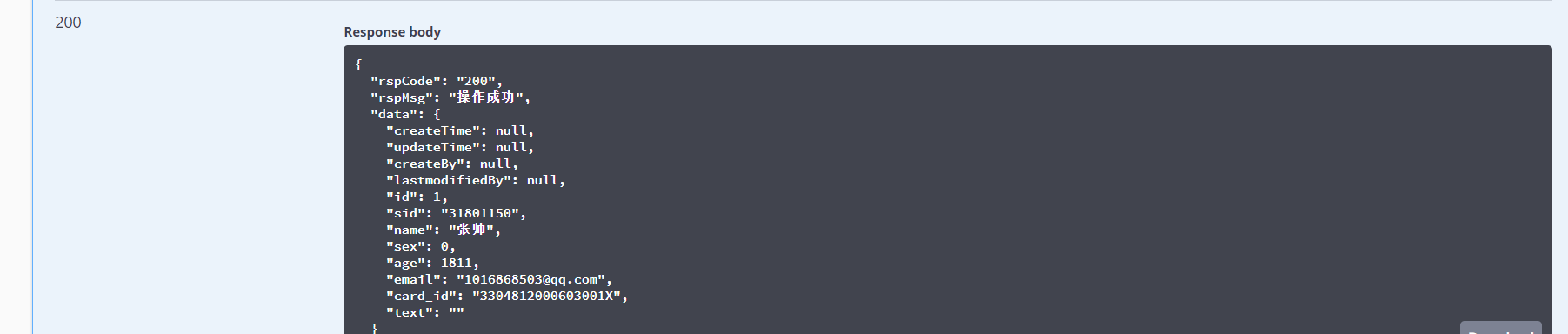


查：

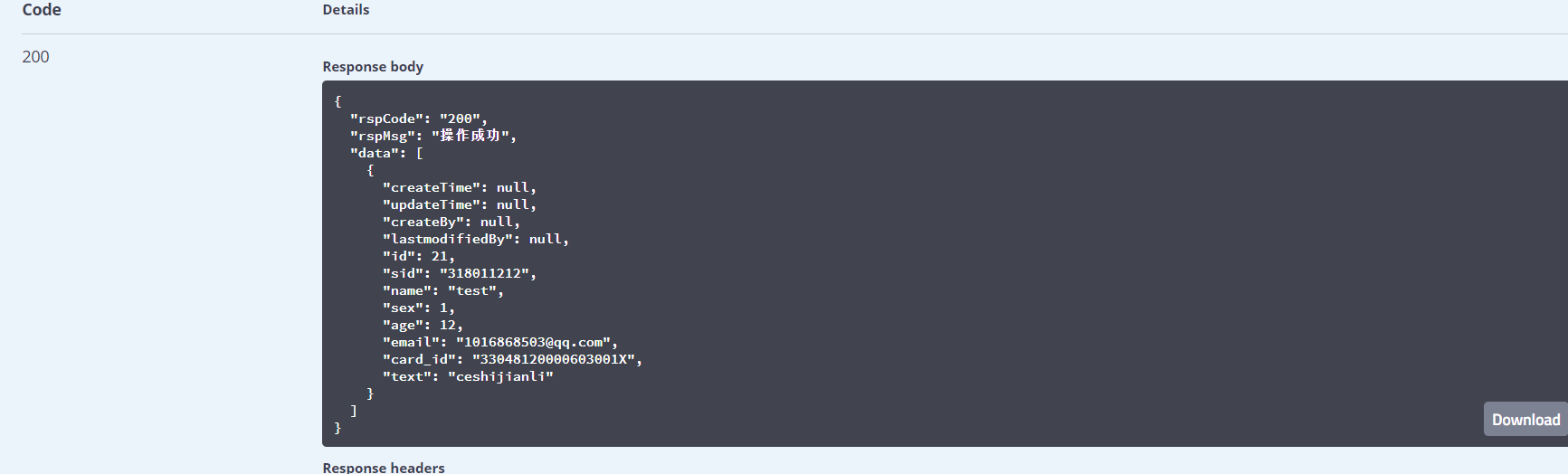
分页：

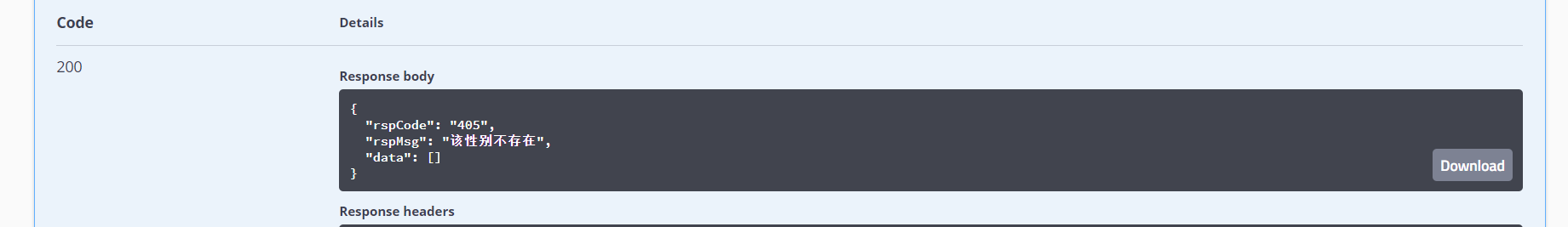


按照id：



按照性别：





按照学号：

