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

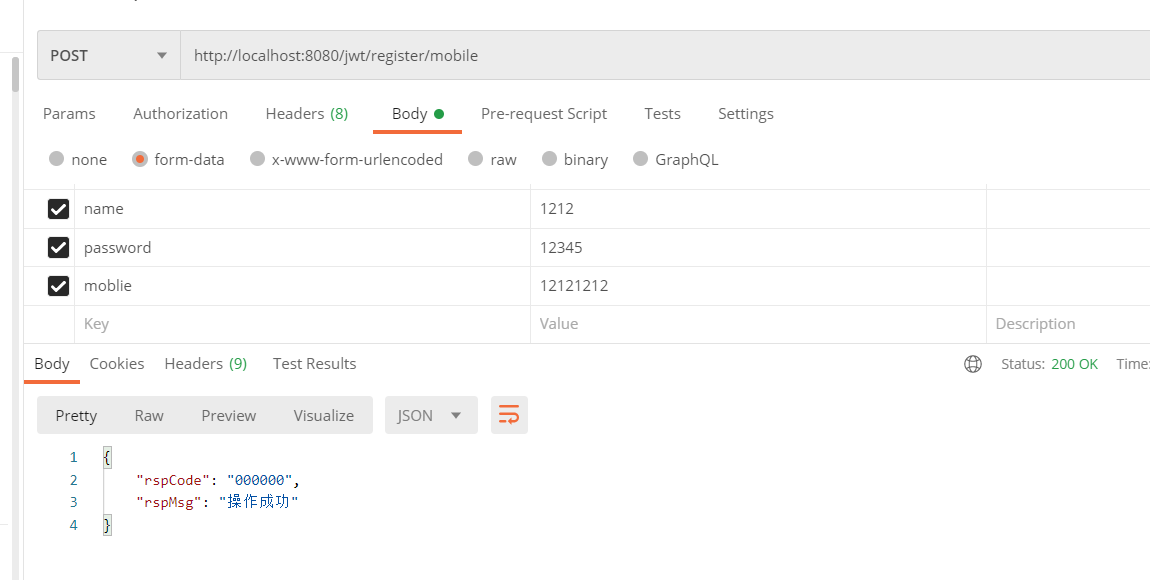
1.学习本节示例用JWT技术为Spring Boot的API增加认证保护，并代码重现

2.学习本节示例用Shiro实现管理后台的动态权限，并代码重现

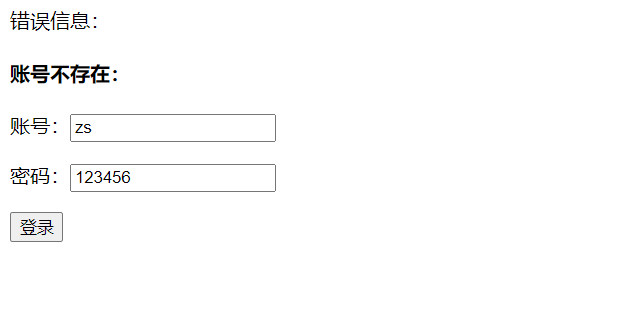
3.为学生信息系统增加权限功能，采用JWT技术或Shiro技术

二. 实验内容

1.学习本节示例用JWT技术为Spring Boot的API增加认证保护，并代码重现

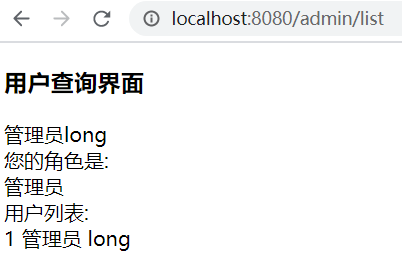


2.学习本节示例用Shiro实现管理后台的动态权限，并代码重现









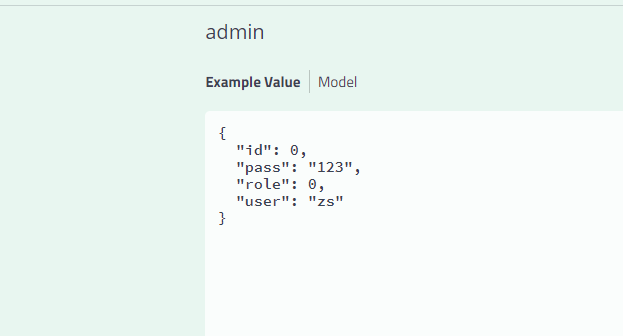
3.为学生信息系统增加权限功能，采用JWT技术或Shiro技术

public class JwtUtil {  
  
 //密钥  
 public static final String *SECRET* = "sdjhakdhajdklsl;o653632";  
 //过期时间:秒  
 public static final int *EXPIRE* = 1000;  
  
  
 public static String createToken(int relo,int id) throws Exception {  
 Calendar nowTime = Calendar.*getInstance*();  
 nowTime.add(Calendar.*SECOND*, *EXPIRE*);  
 Date expireDate = nowTime.getTime();  
  
 Map<String, Object> map = new HashMap<>();  
 map.put("alg", "HS256");  
 map.put("typ", "JWT");  
  
 String token = JWT.*create*()  
 .withHeader(map)//头  
 .withClaim("Id", id)  
 .withClaim("relo", relo)  
 .withSubject("测试")//  
 .withIssuedAt(new Date())//签名时间  
 .withExpiresAt(expireDate)//过期时间  
 .sign(Algorithm.*HMAC256*(*SECRET*));//签名  
 return token;  
 }  
  
 public static Map<String, Claim> verifyToken(String token)throws Exception{  
 JWTVerifier verifier = JWT.*require*(Algorithm.*HMAC256*(*SECRET*)).build();  
 if(token == null)  
 return null;  
 DecodedJWT jwt = null;  
 try {  
 jwt = verifier.verify(token);  
 }catch (Exception e){  
 e.printStackTrace();  
 return null;  
 }  
  
 return jwt.getClaims();  
 }  
  
  
 public static Map<String, Claim> parseToken(String token){  
 DecodedJWT decodedJWT = JWT.*decode*(token);  
 return decodedJWT.getClaims();  
 }  
  
}

public class PermitUtil {  
  
 public static int getPermit (String token){  
 int relo = 0;  
 try {  
  
 if(JwtUtil.*verifyToken*(token).get("relo") == null){  
 return 0;  
 }else{  
  
 relo = JwtUtil.*verifyToken*(token).get("relo").asInt();  
 }  
  
 } catch (Exception e) {  
 e.printStackTrace();  
 return 0;  
 }  
  
  
 if(relo == 1){  
 return 1;  
 }else{  
 return 2;  
 }  
  
 }  
}

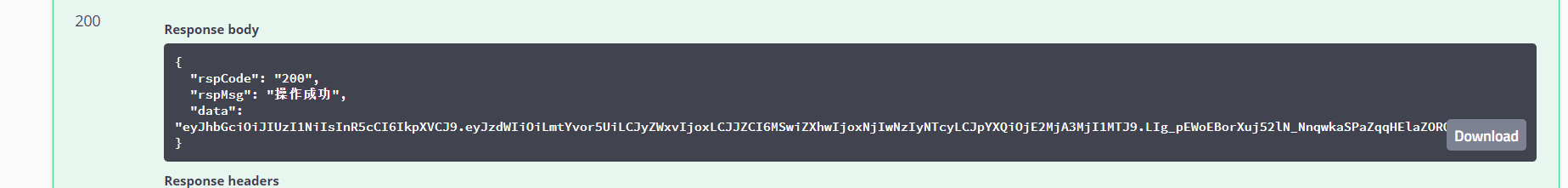
@RestController  
@RequestMapping("admin")  
public class AdminController {  
  
  
  
 @Autowired  
 private AdminMapper adminMapper;  
  
 @PostMapping("/enter")  
 public ResponseData Stulist(@RequestBody AdminEntity admin ) {  
  
  
  
 if(adminMapper.queryByUser(admin.getUser()) == null)  
 return new ResponseData(new Response("405","账号不存在"),null);  
 else if( !adminMapper.queryByUser(admin.getUser()).getPass().equals(admin.getPass()) )  
  
 return new ResponseData(new Response("405","密码错误"),null);  
 else {  
 String token = null;  
 try {  
 token = JwtUtil.*createToken*(adminMapper.queryByUser(admin.getUser()).getRole(),  
 adminMapper.queryByUser(admin.getUser()).getId());  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 return new ResponseData(ExceptionMsg.*SUCCESS*,token);  
 }  
  
 }  
  
  
  
  
  
}

登录系统：（失败）



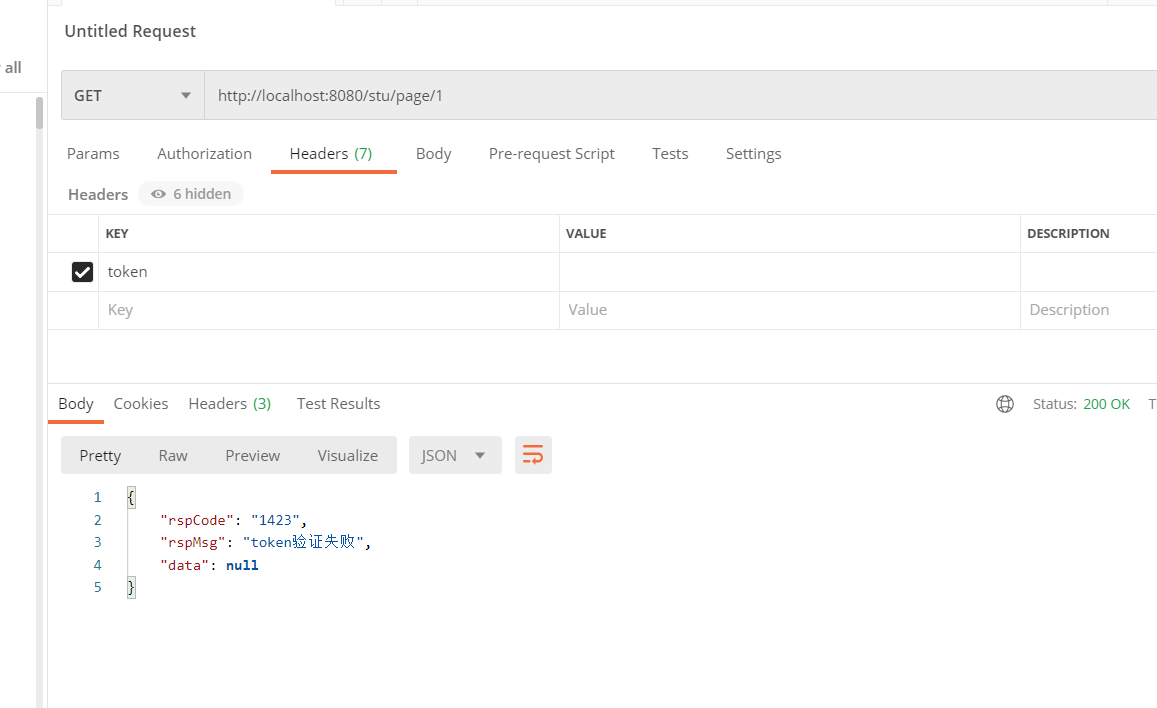


登录系统：（成功，获得token）

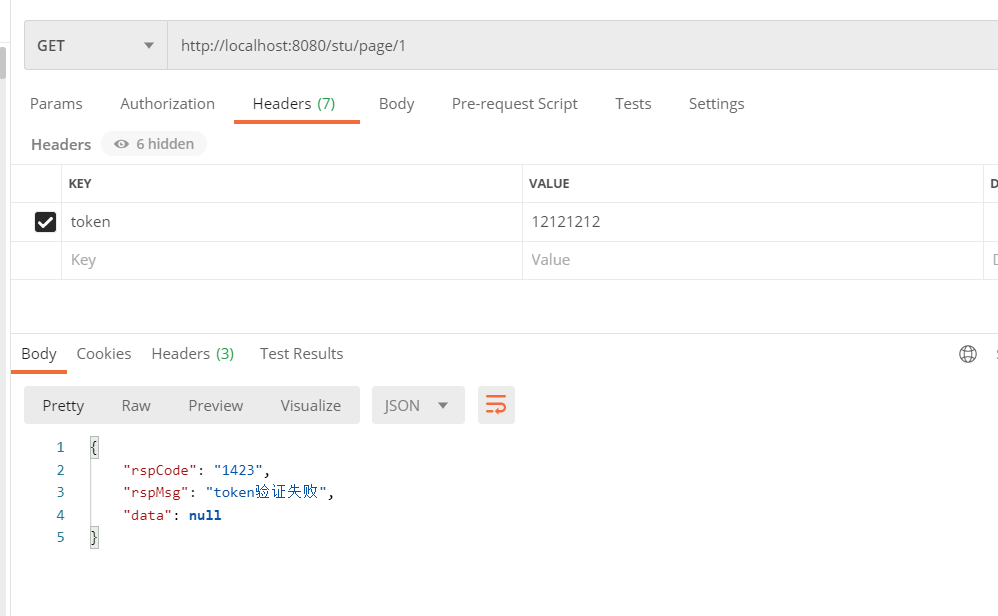


**eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiLmtYvor5UiLCJyZWxvIjoxLCJJZCI6MSwiZXhwIjoxNjIwNzIyNTcyLCJpYXQiOjE2MjA3MjI1MTJ9.LIg\_pEWoEBorXuj52lN\_NnqwkaSPaZqqHElaZORQUMc**

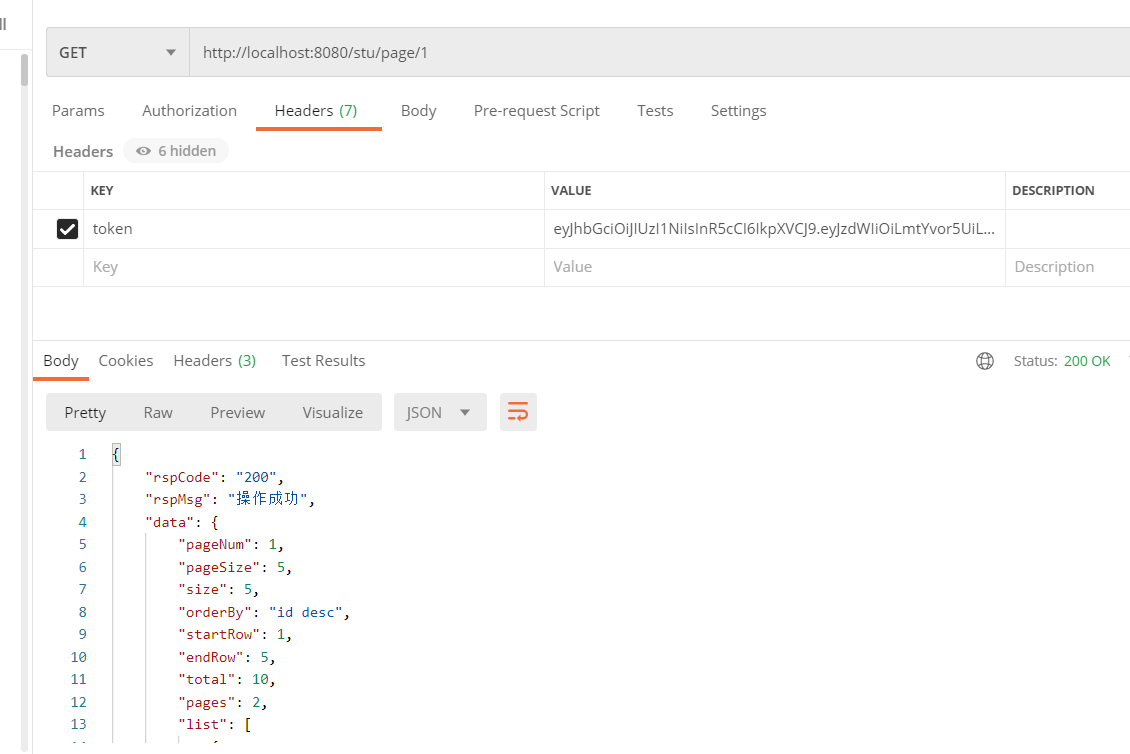
访问接口：（没有token）



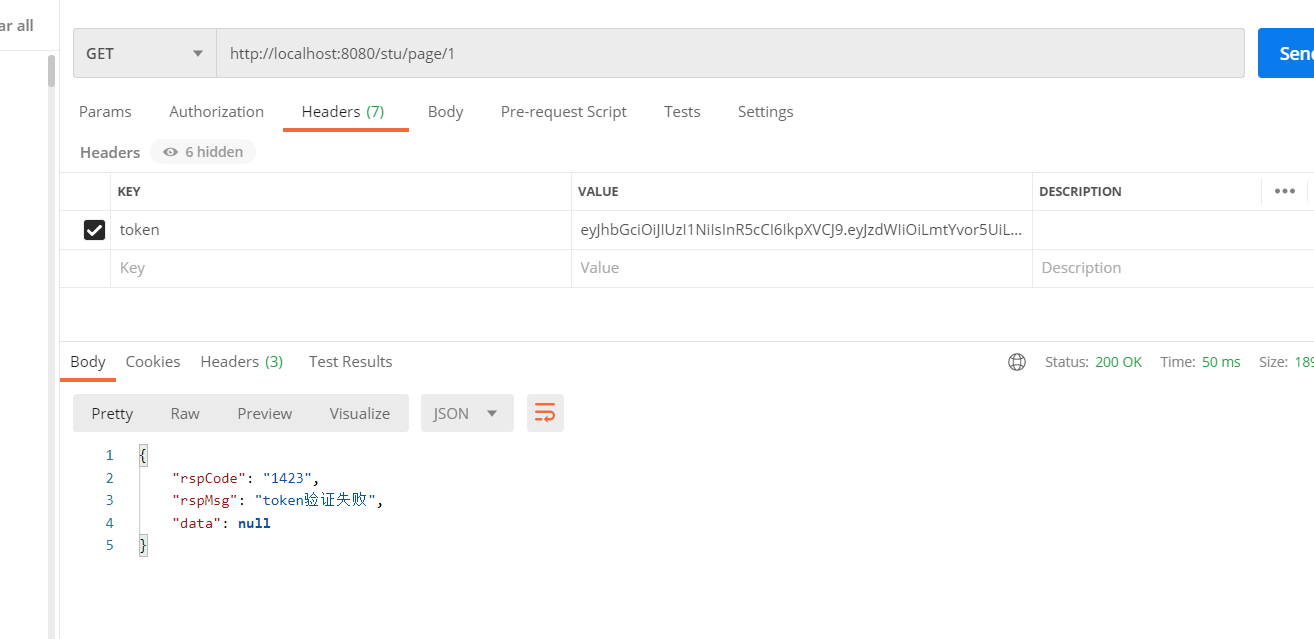
访问接口：（token错误）



访问接口：（token错误）



访问接口：（token过期）



访问接口：（token权限不足）：

