1. 接口开发 首页接口，登录接口，

|  |
| --- |
| **package** net.xdclass.xdclassshiro.controller;  **import** net.xdclass.xdclassshiro.domain.JsonData; **import** net.xdclass.xdclassshiro.domain.UserQuery; **import** net.xdclass.xdclassshiro.service.UserService; **import** org.apache.commons.collections.MapUtils; **import** org.apache.shiro.SecurityUtils; **import** org.apache.shiro.authc.UsernamePasswordToken; **import** org.apache.shiro.subject.Subject; **import** org.springframework.beans.factory.annotation.Autowired; **import** org.springframework.web.bind.annotation.\*;  **import** javax.servlet.http.HttpServletRequest; **import** javax.servlet.http.HttpServletResponse; **import** java.util.ArrayList; **import** java.util.HashMap; **import** java.util.List; **import** java.util.Map; **import** java.util.function.ToLongBiFunction;  @RestController @RequestMapping(**"/pub"**) **public class** PublicController {   @Autowired  **private** UserService **userService**;   *//http://localhost:8080/pub/find\_user\_info?username=jack* @GetMapping(**"/find\_user\_info"**)  **public** Object queryUserInfo(@RequestParam(**"username"**) String username) {  **return userService**.findAllUserInfoByUsername(username);  }   @RequestMapping(**"/need\_login"**)  **public** JsonData needLogin(){  **return** JsonData.*buildError*(**"请登录"**,-2);  }  *// shiroconfig配置类里面配置的路径* @RequestMapping(**"/not\_permit"**)  **public** JsonData noPermit(){  **return** JsonData.*buildError*(**"没有权限访问"**,-3);  }   *//首页,任何角色的都可以访问* @RequestMapping(**"/index"**)  **public** JsonData index(){  List<String> videoList = **new** ArrayList<>();  videoList.add(**"Mysql零基础入门到实战 数据库教程"**);  videoList.add(**"Redis高并发高可用集群百万级秒杀实战"**);  videoList.add(**"Zookeeper+Dubbo视频教程 微服务教程分布式教程"**);  videoList.add(**"2019年新版本RocketMQ4.X教程消息队列教程"**);  videoList.add(**"微服务SpringCloud+Docker入门到高级实战"**);  **return** JsonData.*buildSuccess*(videoList);  }    */\*\*  \* 登录接口  \** ***@param userQuery*** *\** ***@param request*** *\** ***@param response*** *\** ***@return*** *\*/* @PostMapping(**"/login"**)  **public** JsonData login(@RequestBody UserQuery userQuery, HttpServletRequest request,  HttpServletResponse response){  Map<String,Object> info = **new** HashMap<>();  **try** {  Subject subject = SecurityUtils.*getSubject*();  UsernamePasswordToken token = **new** UsernamePasswordToken(userQuery.getName(),userQuery.getPwd());  subject.login(token);   info.put(**"msg"**, **"登录成功"**);  info.put(**"session\_id"**,subject.getSession().getId());  **return** JsonData.*buildSuccess*(info);  } **catch** (Exception e){  e.printStackTrace();  **return** JsonData.*buildError*(**"登录失败"**);  }  }  } |

1. 视频观看列表接口

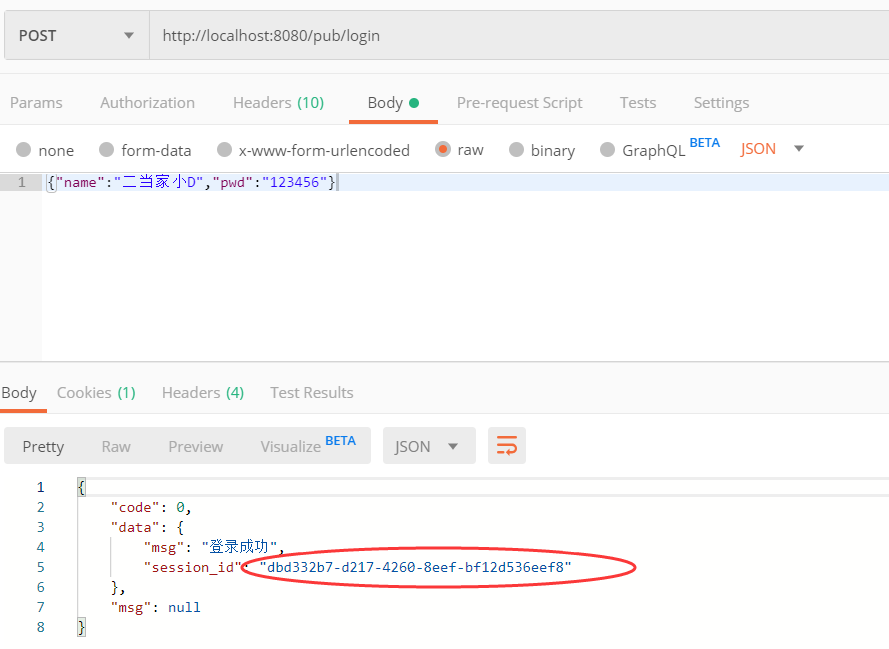
|  |
| --- |
| **package** net.xdclass.xdclassshiro.controller;  **import** net.xdclass.xdclassshiro.domain.JsonData; **import** org.springframework.web.bind.annotation.PostMapping; **import** org.springframework.web.bind.annotation.RequestMapping; **import** org.springframework.web.bind.annotation.RestController;  **import** java.util.HashMap; **import** java.util.Map;  */\*\*  \* 登录才能访问的接口，使用@RestController，返回json数据到前端  \*/* @RestController  @RequestMapping(**"/authc"**) **public class** OrderController {   */\*\*  \* 视频播放列表接口  \** ***@return*** *\*/* @PostMapping(**"/video/play\_record"**)  **public** JsonData findPlayRecord(){  Map<String ,String> recordMap = **new** HashMap<>();  recordMap.put(**"SpringBoot入门到高级实战"**,**"第8章第1集"**);  recordMap.put(**"Cloud微服务入门到高级实战"**,**"第4章第10集"**);  recordMap.put(**"分布式缓存Redis"**,**"第10章第3集"**);  **return** JsonData.*buildSuccess*(recordMap);  }  } |

Psotman测试：

1. 首先将net.xdclass.xdclassshiro.config.ShiroConfig 类中的customRealm方法中的

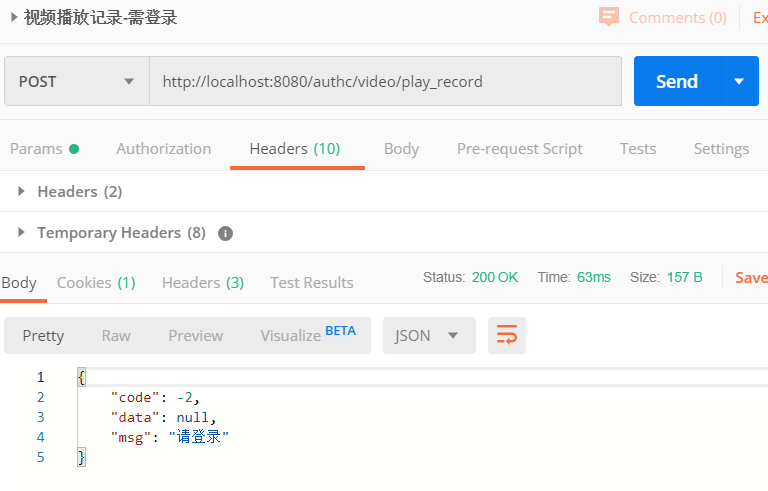
*customRealm.setCredentialsMatcher(hashedCredentialsMatcher()); 注释掉，否则使用明文密码登录时会进行密码解密验证，导致登录失败*

1. 测试登录接口：<http://localhost:8080/pub/login> 请求体： {"name":"二当家小D","pwd":"123456"}

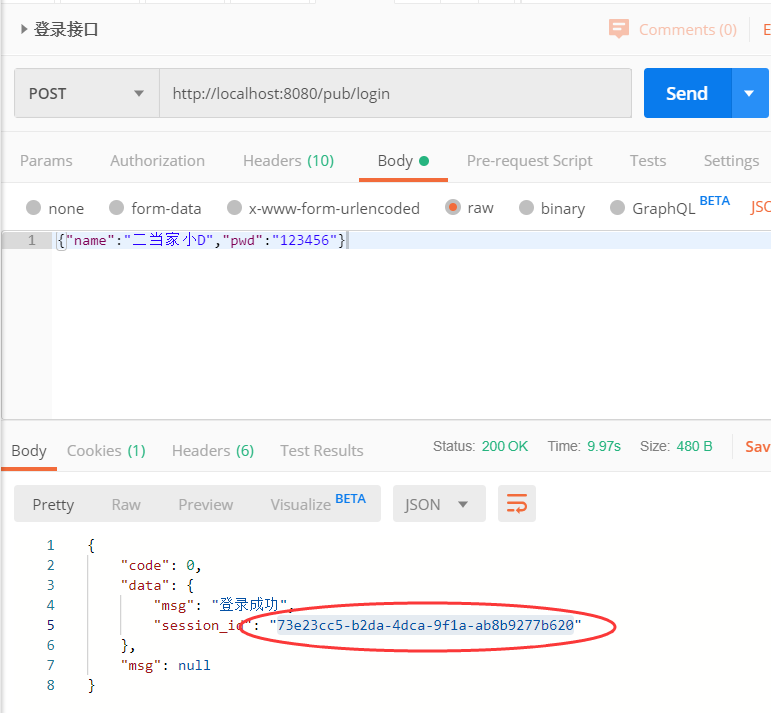


1. 测试需要登录的接口

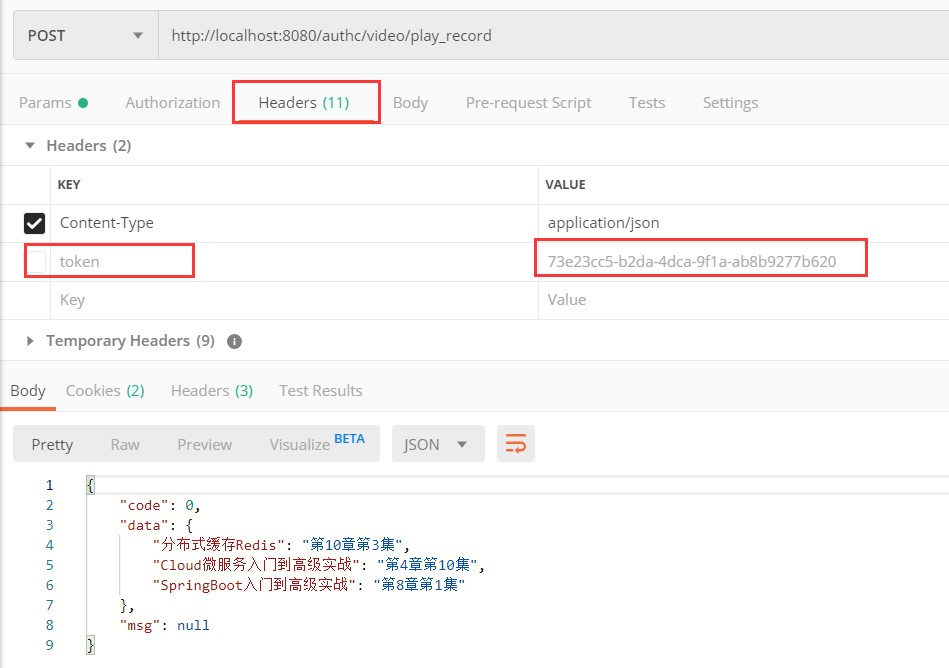
不登录，直接请求[http://localhost:8080/authc/video/play\_record](http://localhost:8080/video/play_record)接口：



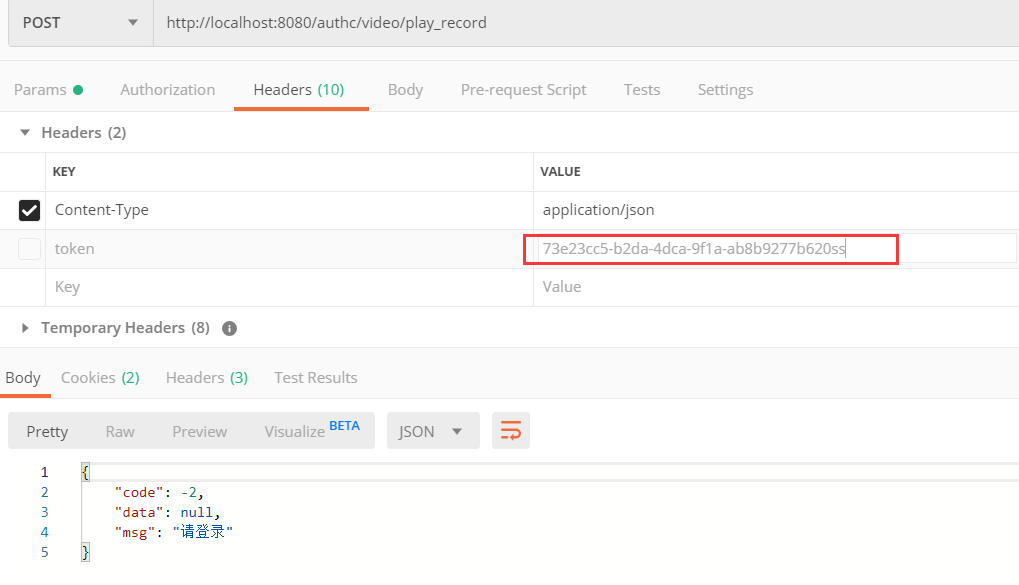
登录：



请求头带上登录获取到的sessionId请求 [http://localhost:8080/authc/video/play\_record](http://localhost:8080/video/play_record) 接口



篡改sessionId, 提示需要登陆



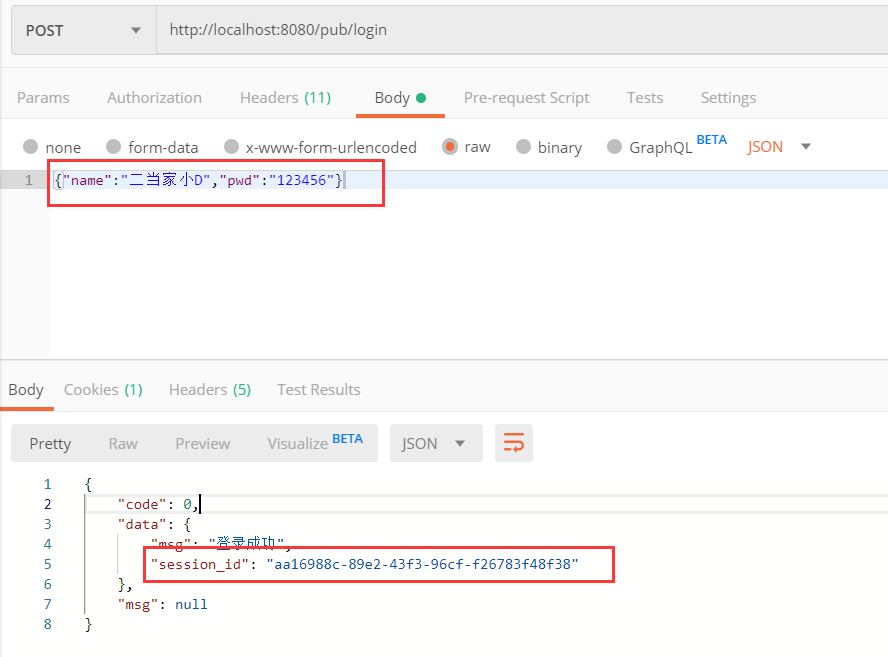
登录之后过20s，再去请求[http://localhost:8080/authc/video/play\_record](http://localhost:8080/video/play_record) 接口，也会提示请登录，这是因为之前设置的session超时时间为20秒（net.xdclass.xdclassshiro.config.ShiroConfig#sessionManager中设置session超时时间）

1. 测试需要admin角色才可以访问的接口：

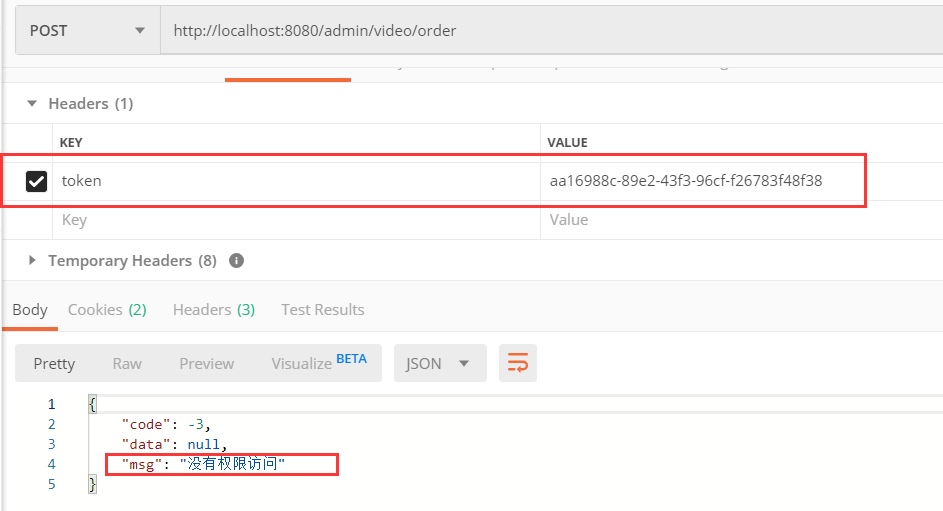
新增controller

|  |
| --- |
| **package** net.xdclass.xdclassshiro.controller;  **import** net.xdclass.xdclassshiro.domain.JsonData; **import** org.springframework.web.bind.annotation.PostMapping; **import** org.springframework.web.bind.annotation.RequestMapping; **import** org.springframework.web.bind.annotation.RestController;  **import** java.util.HashMap; **import** java.util.Map;  */\*\*  \* 只有admin管理员角色才能访问的接口  \*/* @RestController @RequestMapping(**"admin"**) **public class** AdminController {   */\*\*  \* 视频播放列表接口  \** ***@return*** *\*/* @PostMapping(**"/video/order"**)  **public** JsonData findPlayRecord(){  Map<String ,String> recordMap = **new** HashMap<>();  recordMap.put(**"SpringBoot入门到高级实战"**,**"300元"**);  recordMap.put(**"Cloud微服务入门到高级实战"**,**"100元"**);  recordMap.put(**"分布式缓存Redis"**,**"90元"**);  **return** JsonData.*buildSuccess*(recordMap);  } } |

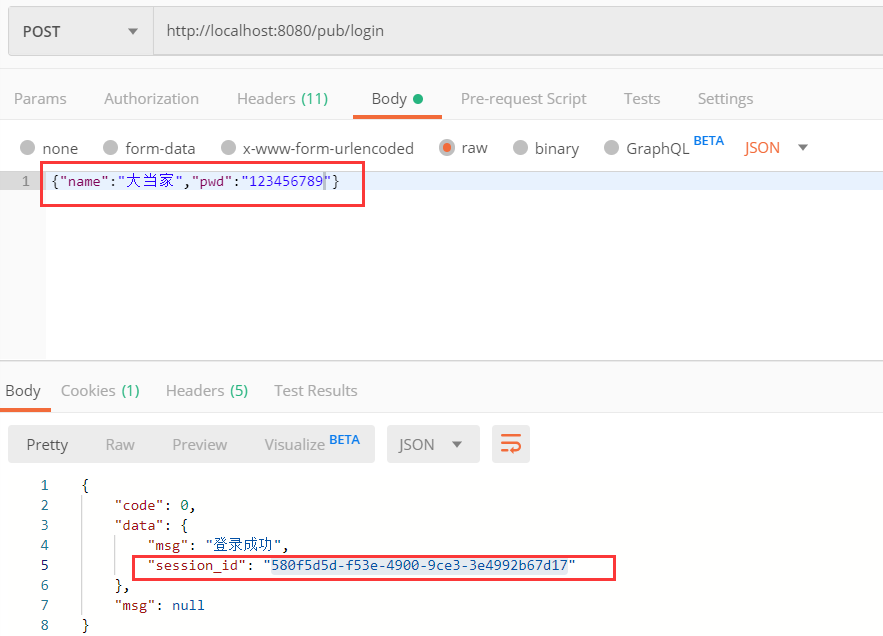
然后使用普通用户登录：

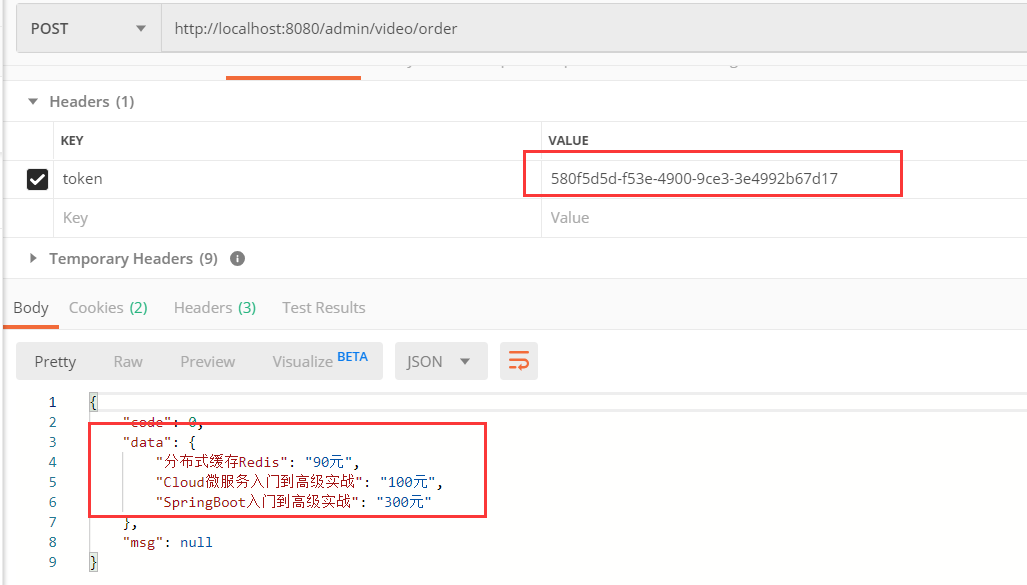


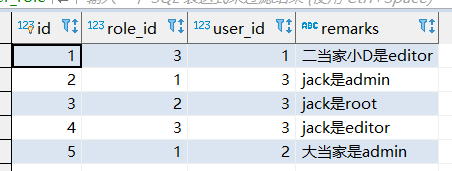
用这个sessionId去请求/admin/**video/order接口** http://localhost:8080/admin/video/order



然后更换为admin账号登录再去访问该接口：







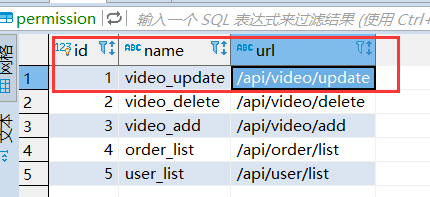
用户大当家具有admin角色，才能访问该接口

1. 具有具体权限的用户才能访问的接口：

filterChainDefinitionMap.put(**"/video/update"**, **"perms[video\_update]"**);

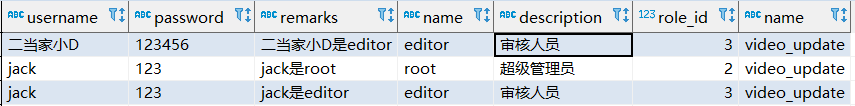
新增测试controller

|  |
| --- |
| **package** net.xdclass.xdclassshiro.controller;  **import** net.xdclass.xdclassshiro.domain.JsonData; **import** org.springframework.web.bind.annotation.PostMapping; **import** org.springframework.web.bind.annotation.RequestMapping; **import** org.springframework.web.bind.annotation.RestController;  **import** java.util.HashMap; **import** java.util.Map;  */\*\*  \* 只有具有video/update权限才能访问的接口  \*/* @RestController @RequestMapping(**"video"**) **public class** VideoController {  @PostMapping(**"/update"**)  **public** JsonData findPlayRecord(){  **return** JsonData.*buildSuccess*(**"video更新成功"**);  } } |

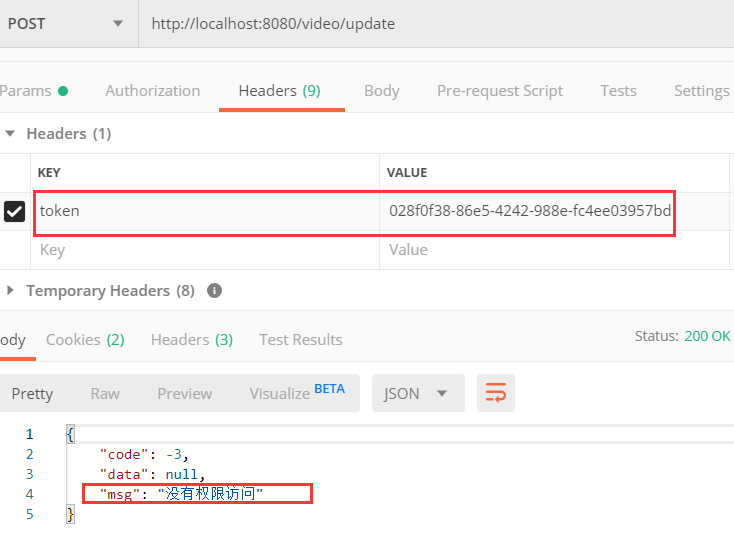
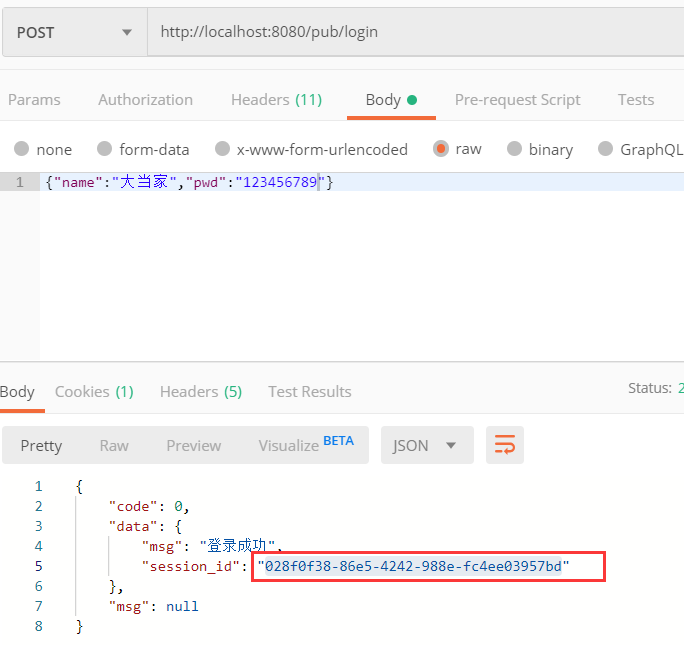


查询数据库可以看到，具有video\_update权限的用户如下：

|  |
| --- |
| **select** u.username,u.**password**,ur.remarks,r.name,r.description,rp.role\_id,p.name **from** **user** u  **left** **join** user\_role ur **on** u.id = ur.user\_id  **left** **join** **role** r **on** r.id = ur.role\_id  **left** **join** role\_permission rp **on** rp.role\_id = r.id  **left** **join** permission p **on** p.id = rp.permission\_id  **where** p.id = 1 |



使用大当家账号登录并访问，提示没有权限



使用jack账号登录并访问

