**VOTING-SYSTEM系统测试文档**

1. 接口概况

接口地址为localhost:3000/api/

访问其他地址会返回404，未带TOKEN 只能访问 login 和register 两个接口，访问其api下的其他页面需要进行TOKEN校验，

TOKEN采用JWT方式。后台自主配置密钥，前端使用TOKEN可以在HEADER中加入x-access-token参数或者直接在query中加入token参数，优先获取query中的。

具体接口地址如下所示，

router.get('/register', register.verficationCode);//获取验证码

router.put('/register', register.signup);//注册

router.post('/login', login.login);//工作人员登录

router.get('/candidates', candidates.getCandidates);//工作人员获取候选人信息，返回包括了投票信息

router.put('/candidates', candidates.addCandidates);//工作人员增加候选人

router.post('/candidates', candidates.modifyCandidates);//增加候选人

router.delete('/candidates', candidates.deleteCandidates);//删除候选人

router.get('/switchs', switchs.voteBegin);//工作人员开系统

router.post('/switchs', switchs.voteEnd);//工作人员关系统

router.post('/voting', voting.voteCandidates);//注册用户投票

1. 满足条件
2. 每个有效邮箱只能注册一次。
3. 每个邮箱地址会进行校验，并且发送验证码进行有效性确认。
4. 用户进行投票所选择的候选人ID不能小于2不能大于20%总候选人数，（如果20%人数小于等于2那么取2）
5. 用户进行投票需要在工作人员开启投票之后才能进行
6. 每个邮箱只能投一次票，非注册邮箱无法获取JWT无法进行投票
7. 工作人员使用预定的账号密码进行登录，同时获取对应的JWT作为身份证明
8. 工作人员可以随时开启或者关闭投票
9. 工作员可以随时查询候选人具体情况（包括得票数）
10. 工作人员可以再开启投票未进行时进行候选人的增加修改删除
11. 用户进行的投票会记录在数据库，方便以后分析。记录的格式为n1|n2|n3,其中n1n2n3为候选人的ID值（由工作人员新增时候指定）
12. 使用技术
13. 使用express框架
14. 使用restful结构规范
15. 使用Mongodb数据库
16. 使用JWT（json web token）作为身份记录
17. 其他处理

1、对于系统关键位置有日志记录，（数据库连接，开关系统时读写本地文件，其他必要处）

2、对于用户密码会进行加密处理，后端配置文件涉及密码的会进行一个掩盖处理

3、除了自身用POSTMAN测试外，提供一套Nodejs 标准测试文以供Unit Test代码

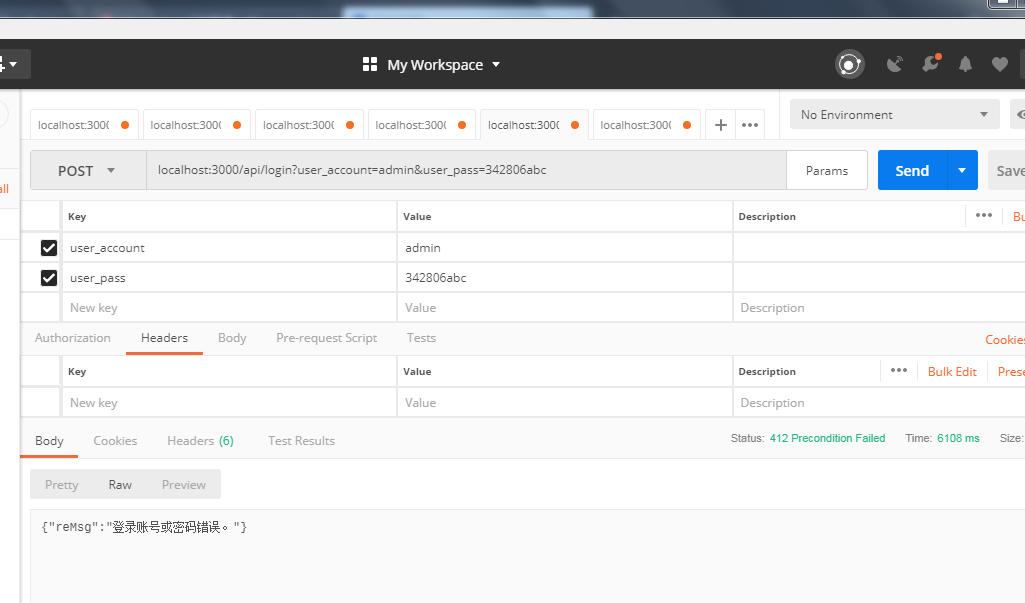
4、对于访问接口或者其他页面都有对应的措施和返回提示。

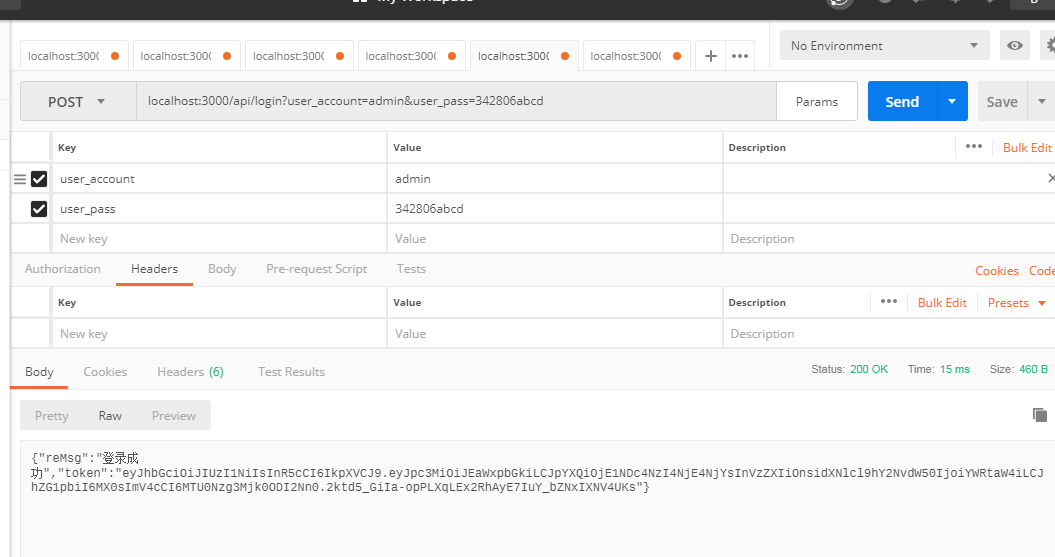
5、在返回候选人数据使用cache机制（使用global全局变量方式，每30秒过期之后访问才会从数据库中获取）。

6、代码流程中相应的注释处理

五、接口测试界面截取---测试工具POSTMAN：

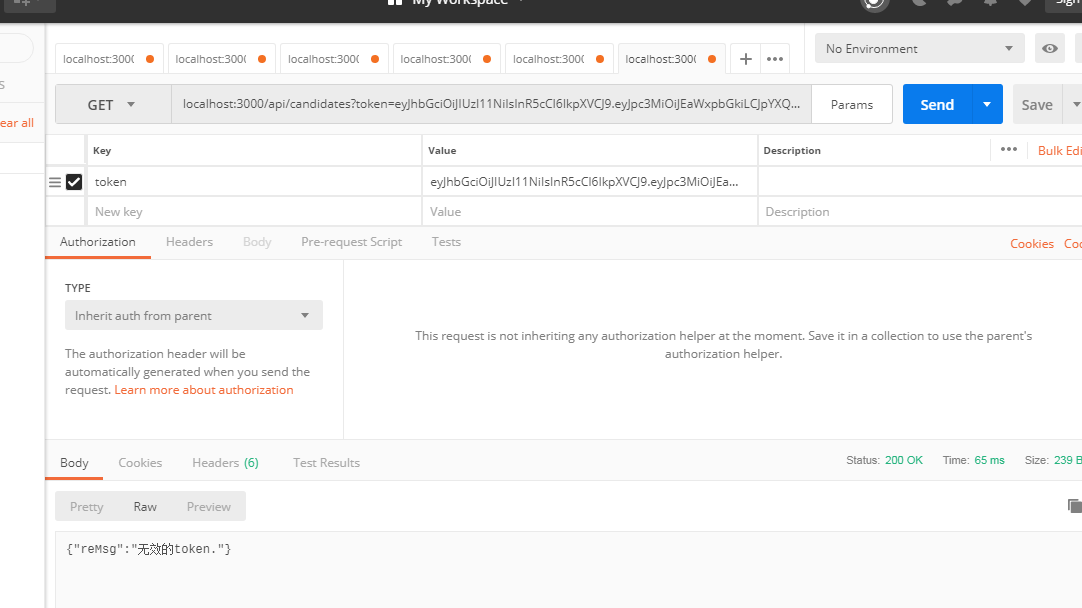
1. 工作人员登录

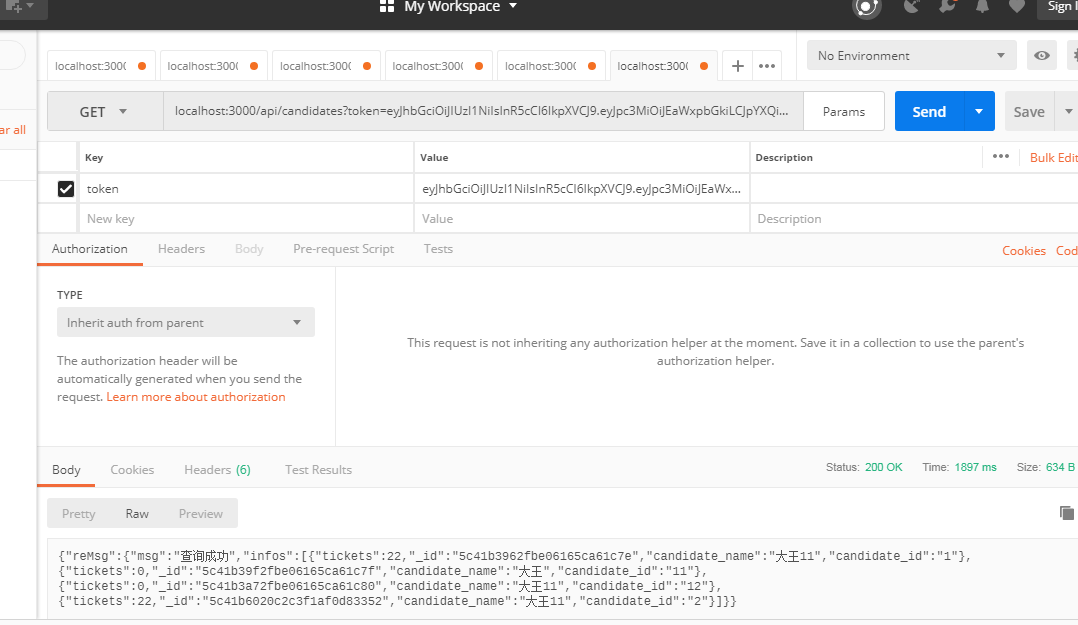




登录成功，返回JWT

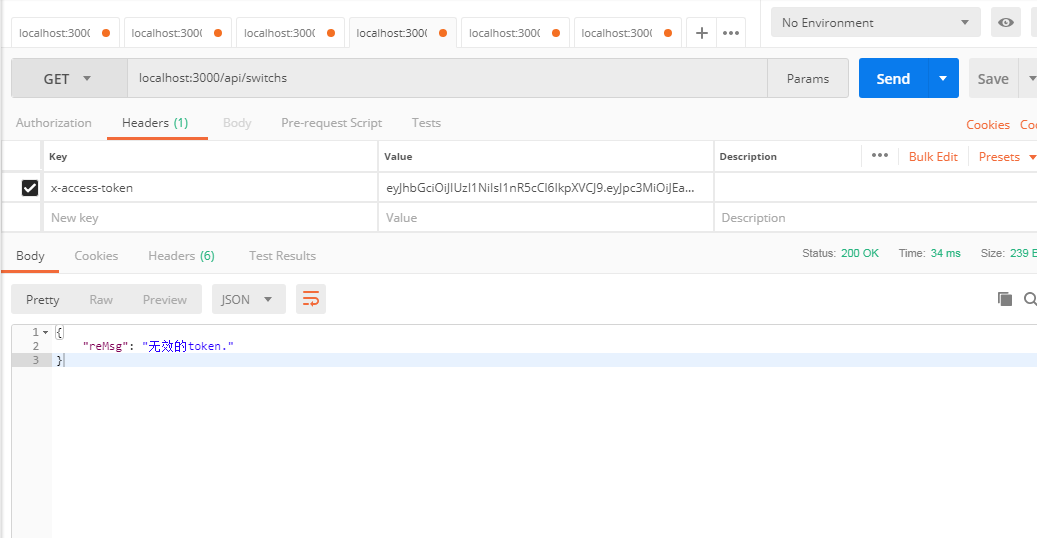
2、工作人员进行候选人信息查询



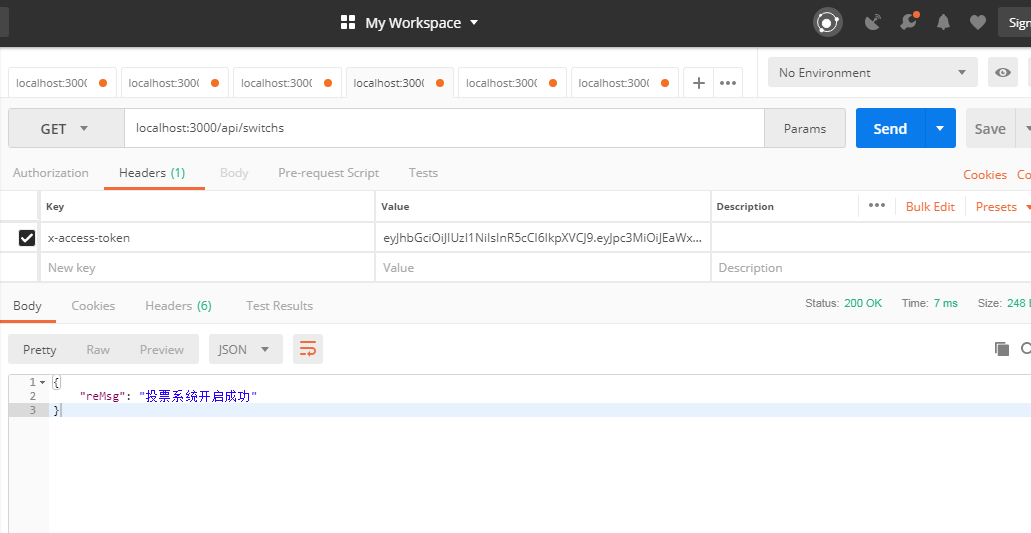


3、工作人员开启投票系统

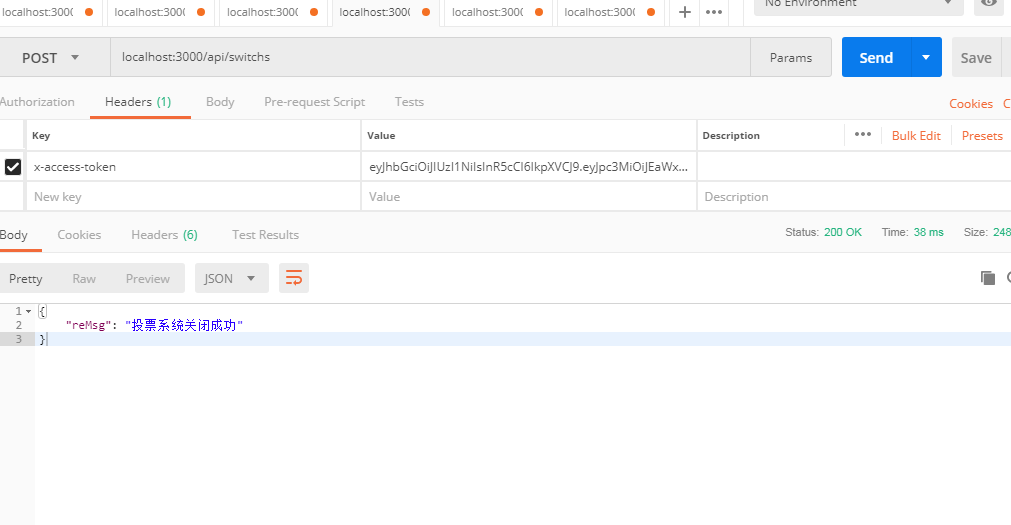
身份验证不通过



正确的TOKEN开启



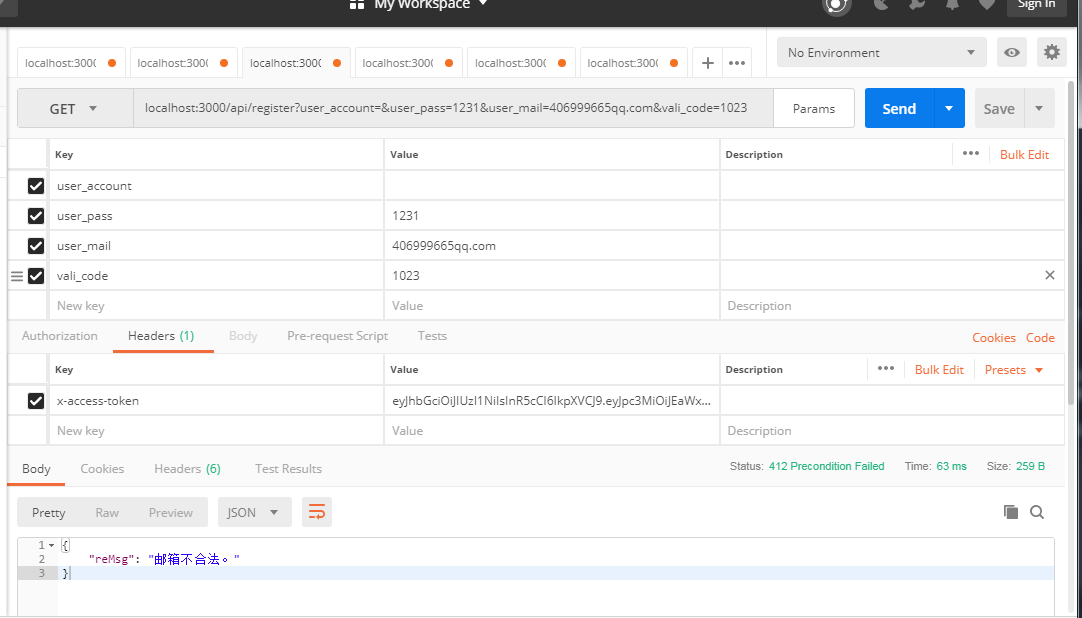
4、工作人员关闭投票系统



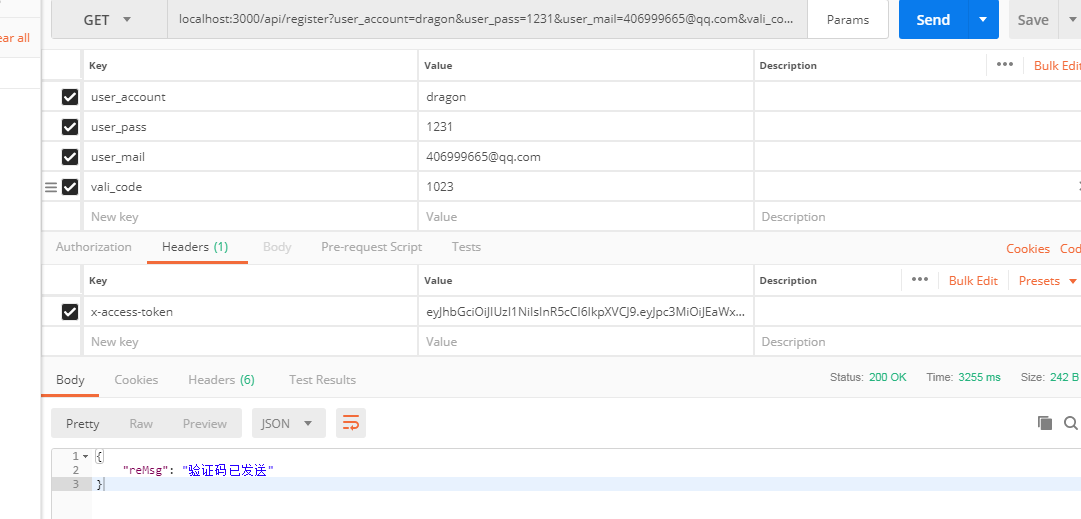
6、用户注册

获取验证码

邮箱不合法返回提示

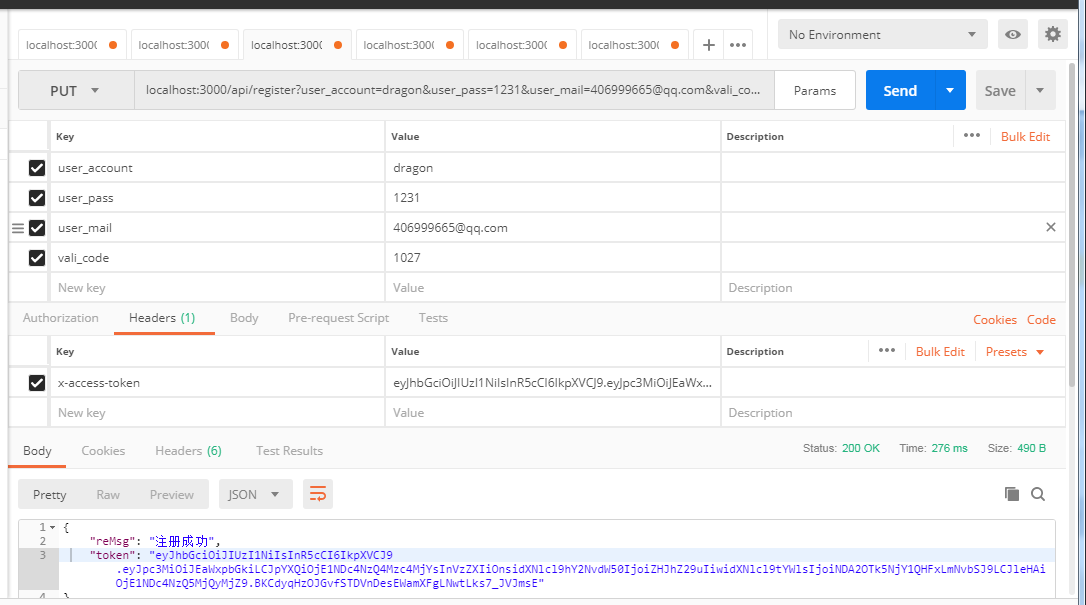


发送给邮箱对应验证码



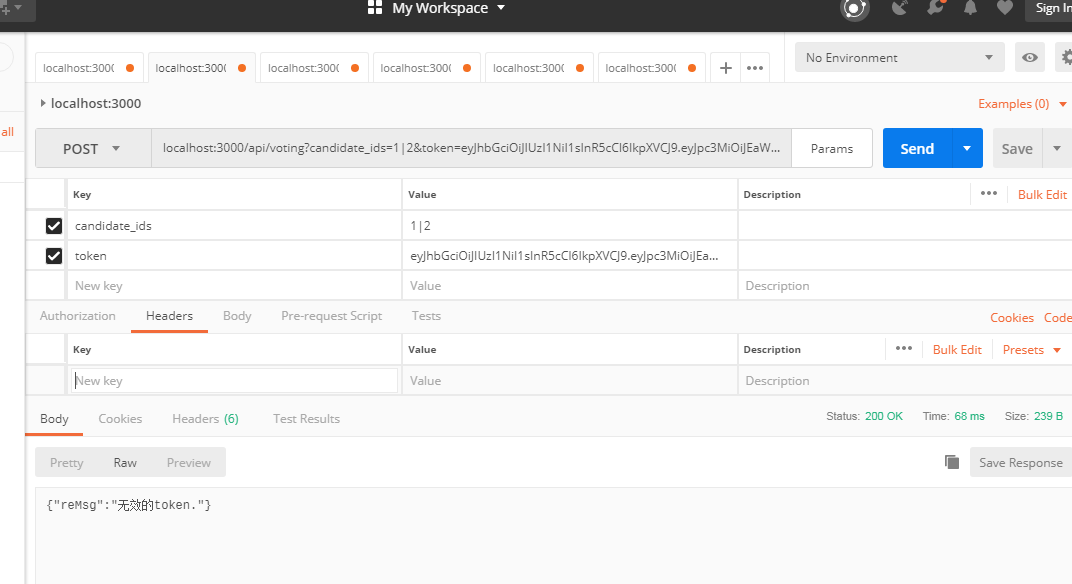


输入账号密码加上港爱的验证码进行注册，注册成功并返回JWT

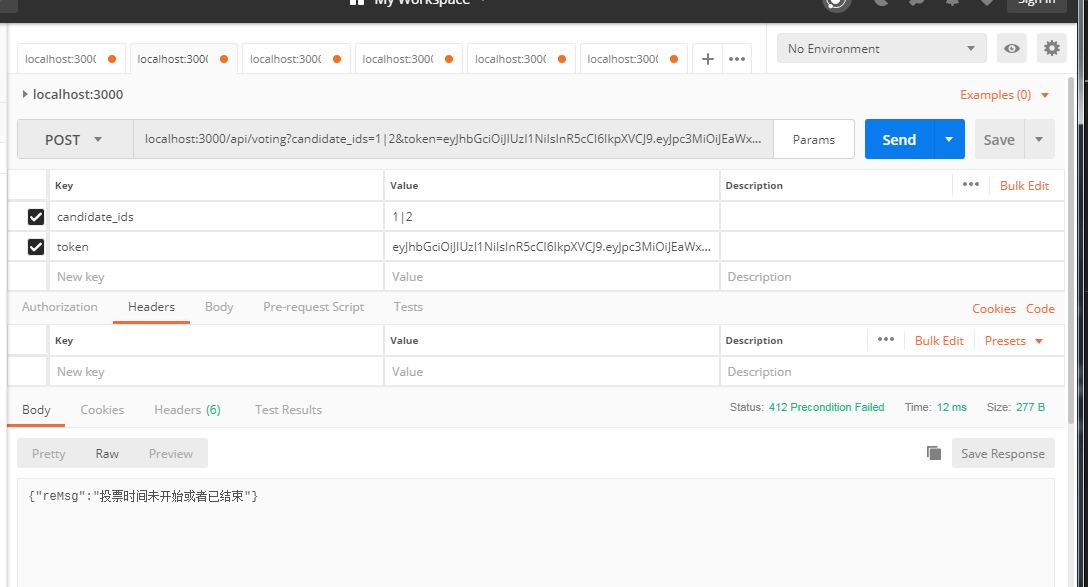


7、用户进行投票

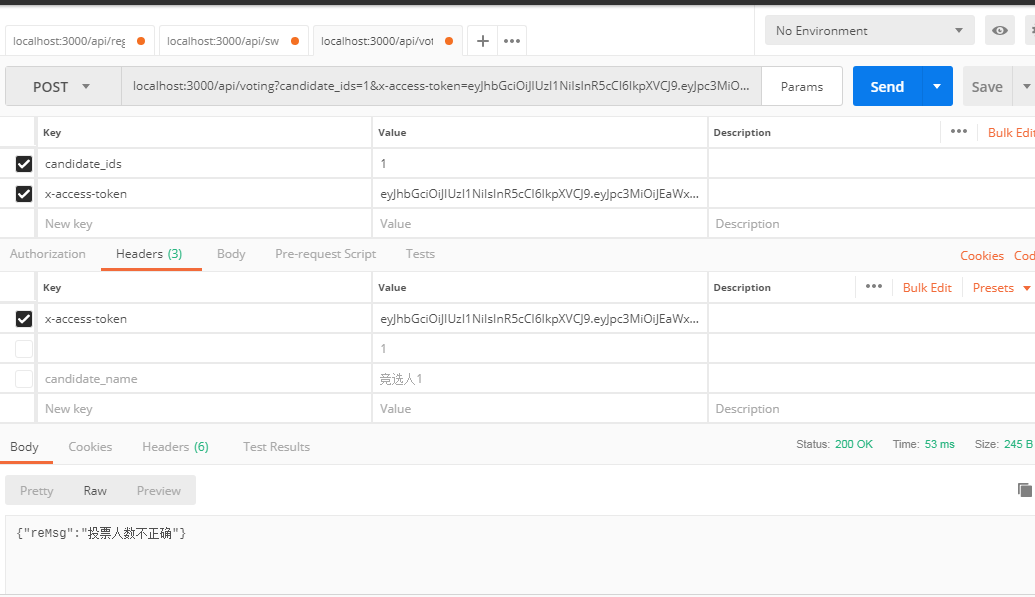
无效token返回提示。

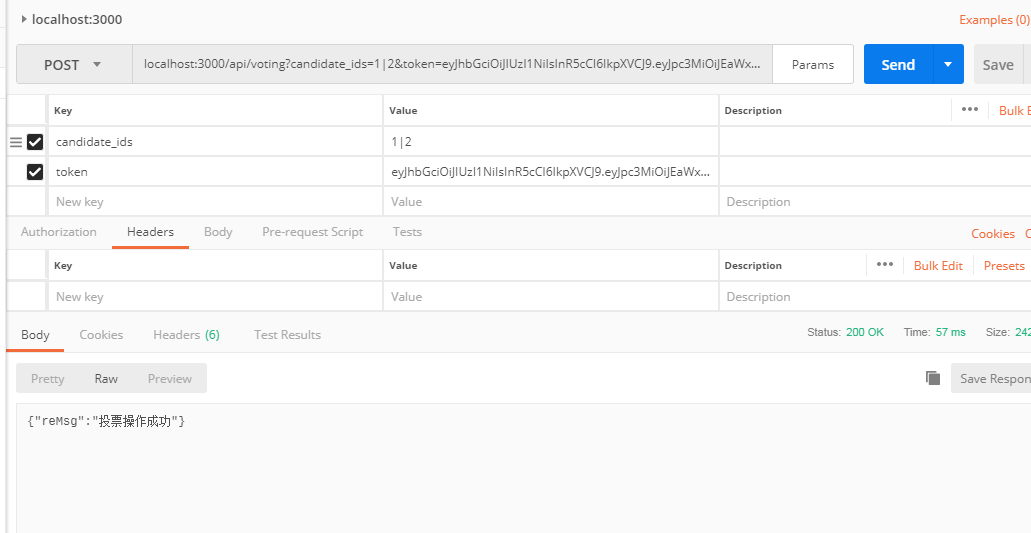


有效的TOKEN返回投票未开始

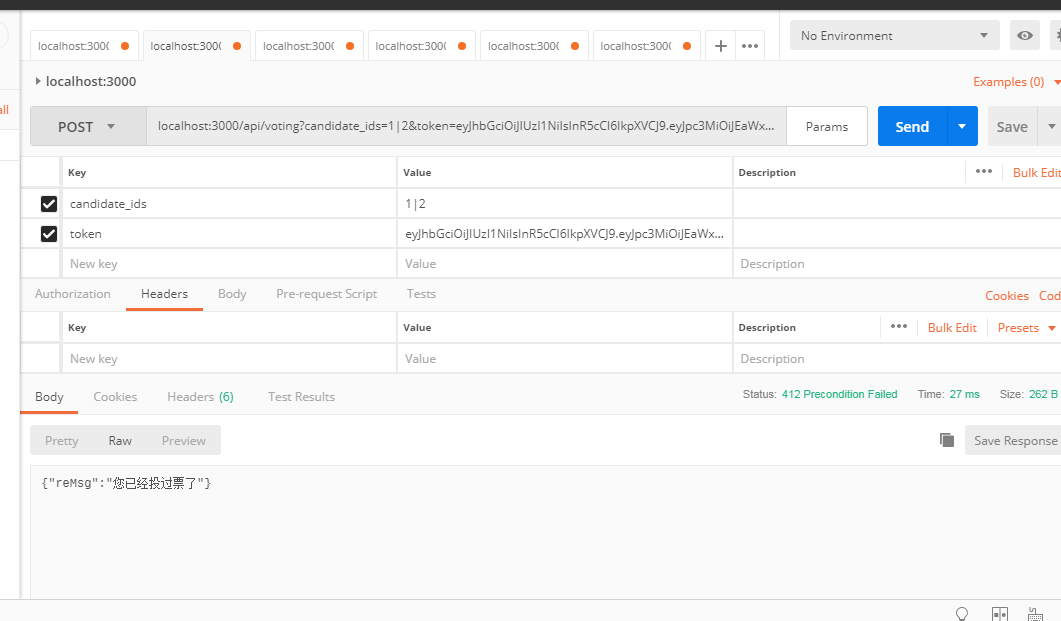


待工作人员开启之后进行投票，按照规定最少选两个候选人，最多20%个候选人的选择范围，正常返回(操作成功)。

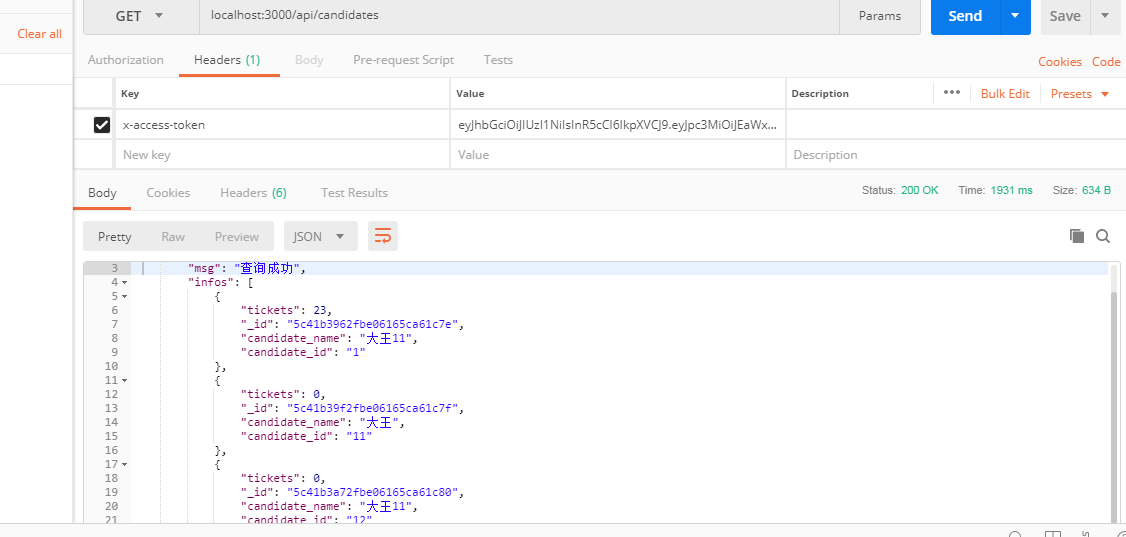
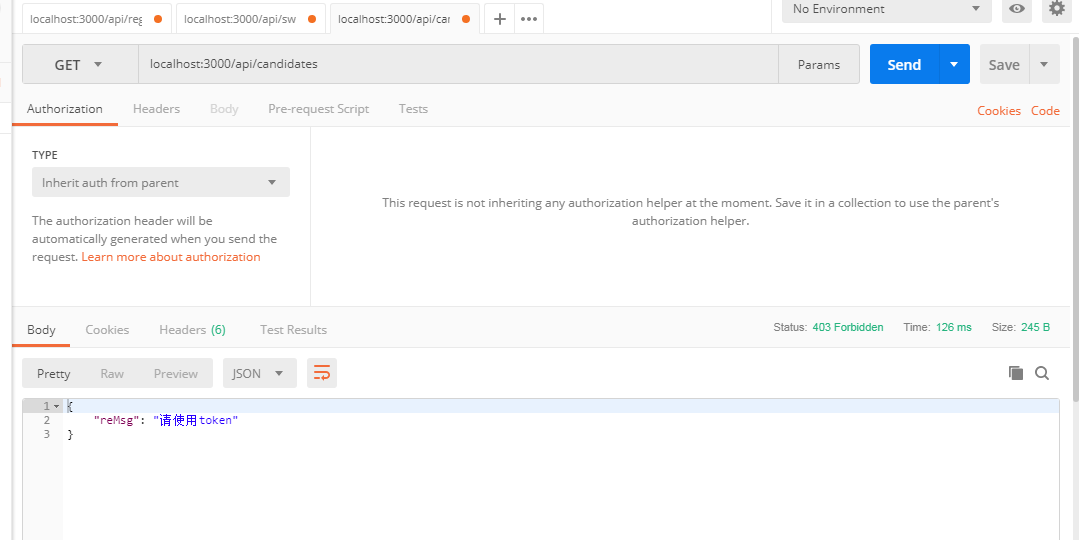




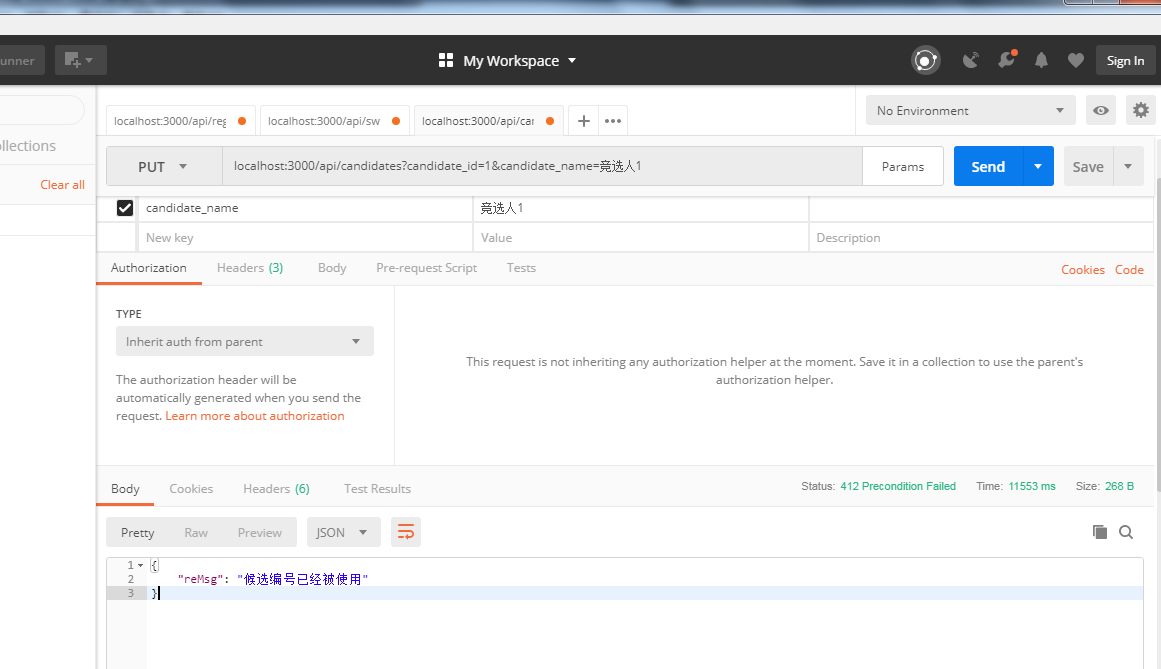
投过票之后，不能重复投

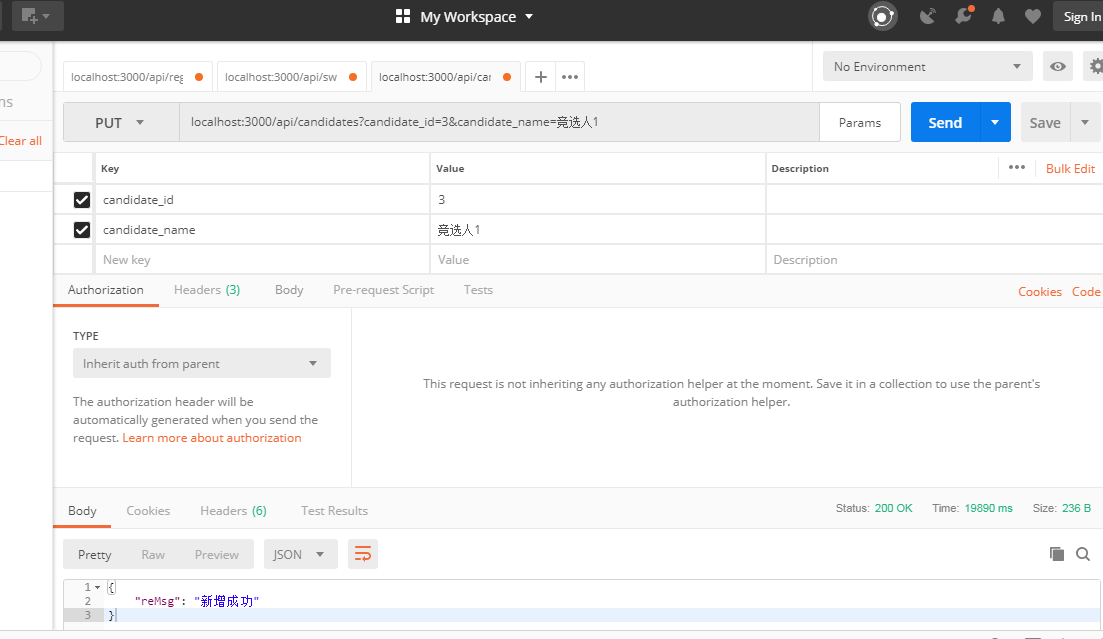


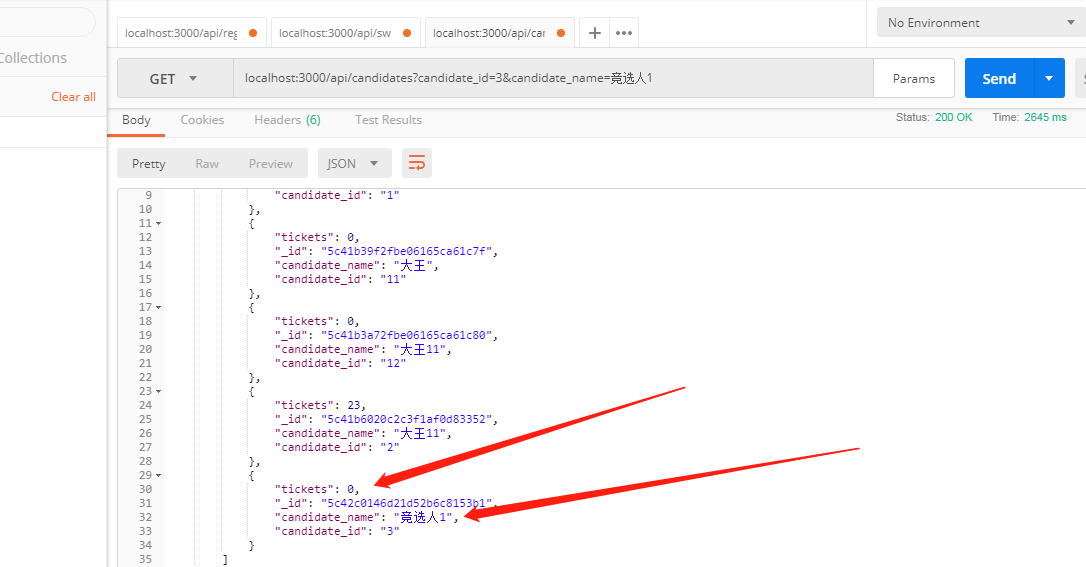
8、工作人员登录之后可以随时进行候选人的增删改以及查询候选人选票信息。



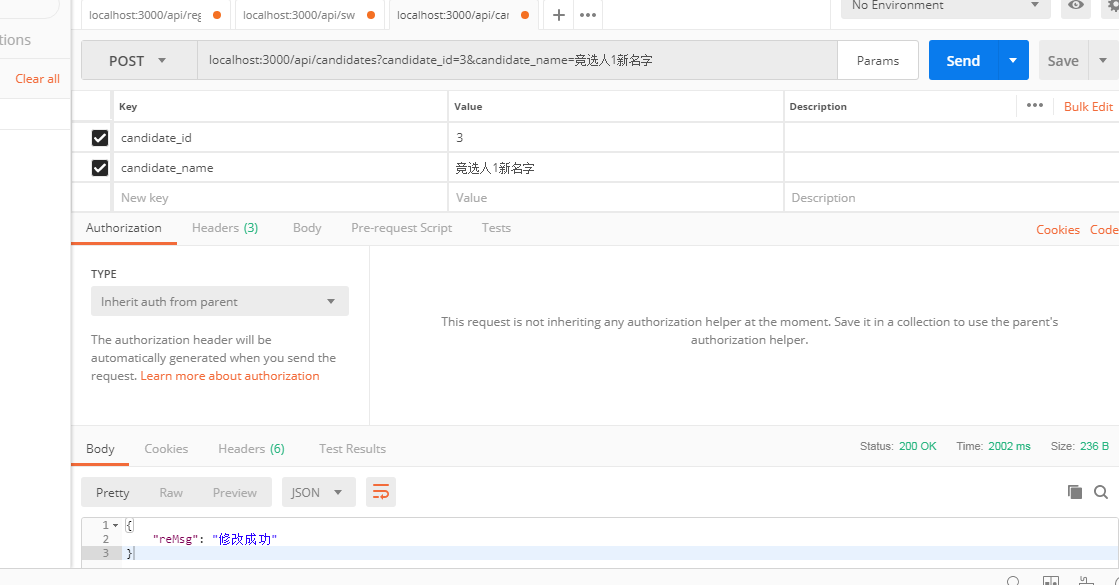
候选人增加



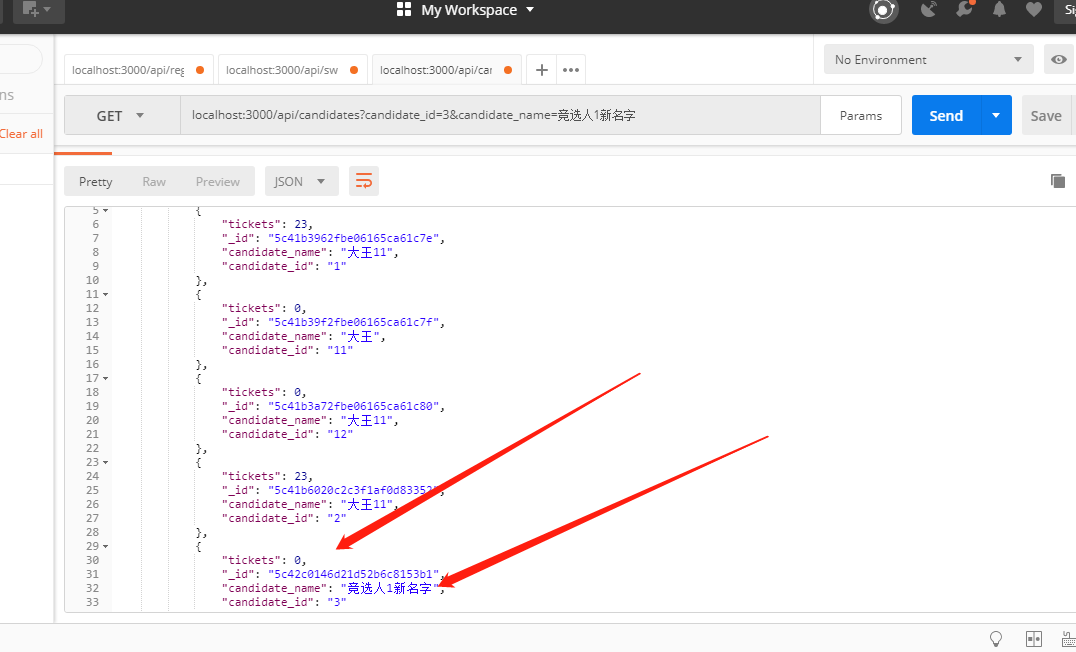




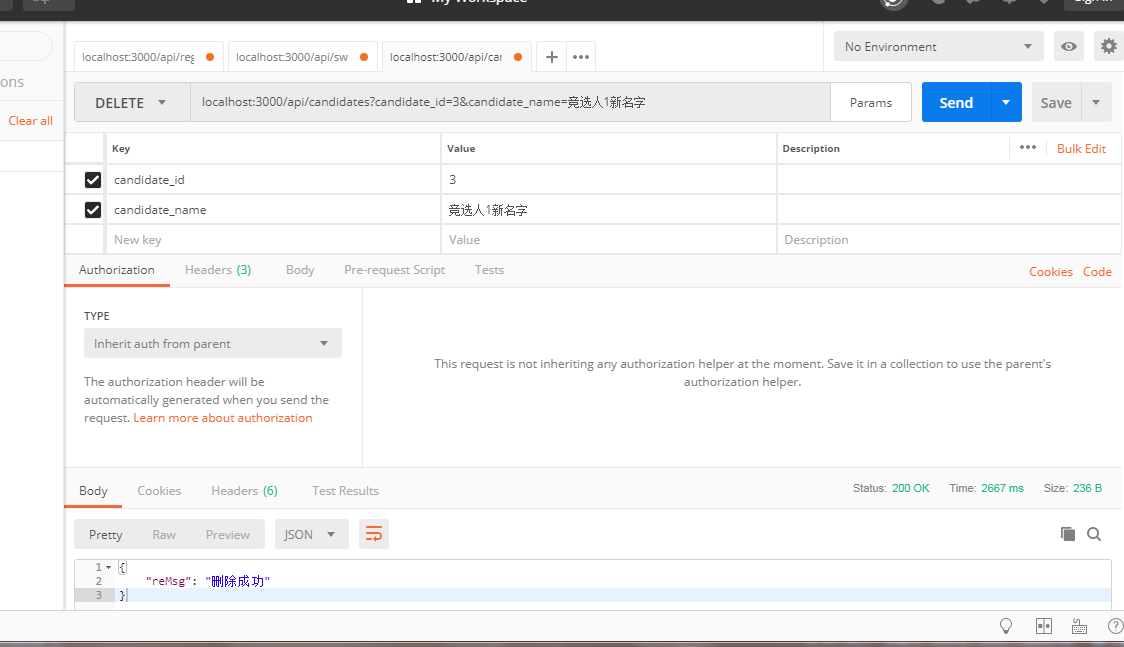
修改候选人信息



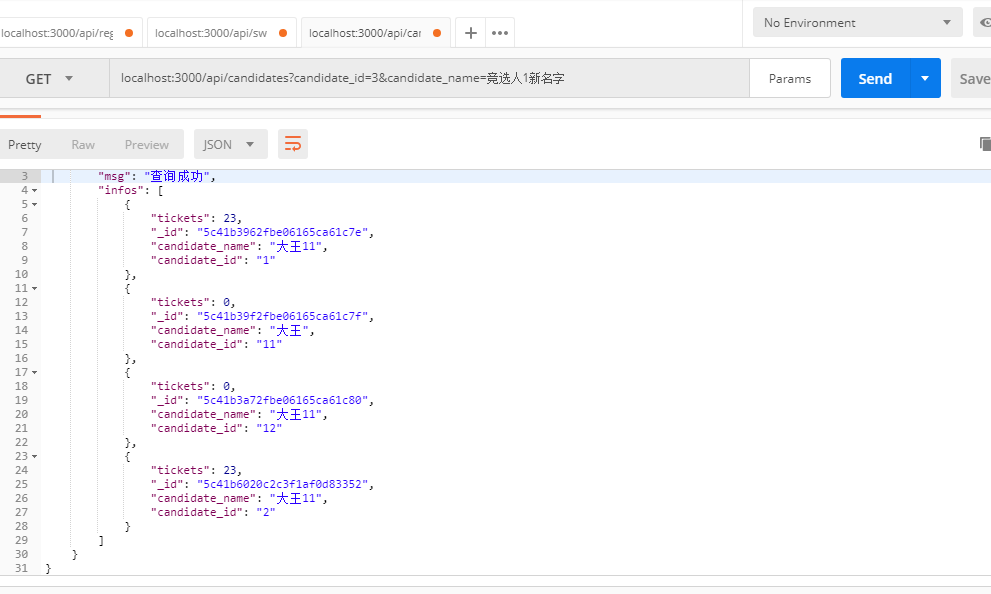
再查询



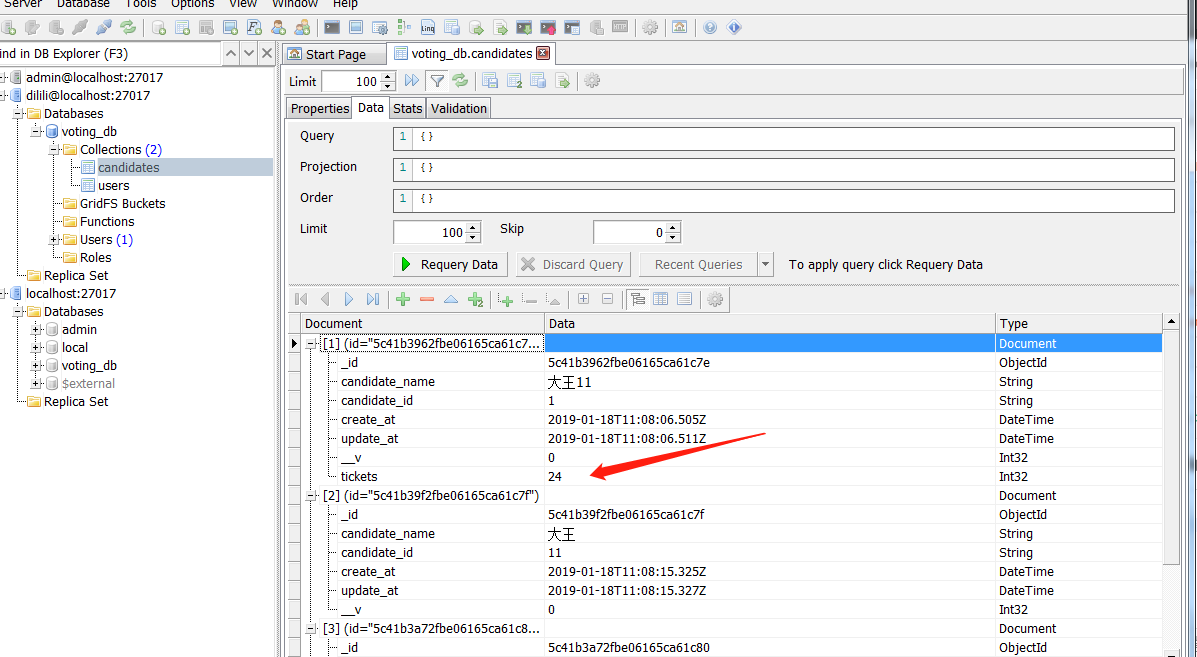
删除候选人

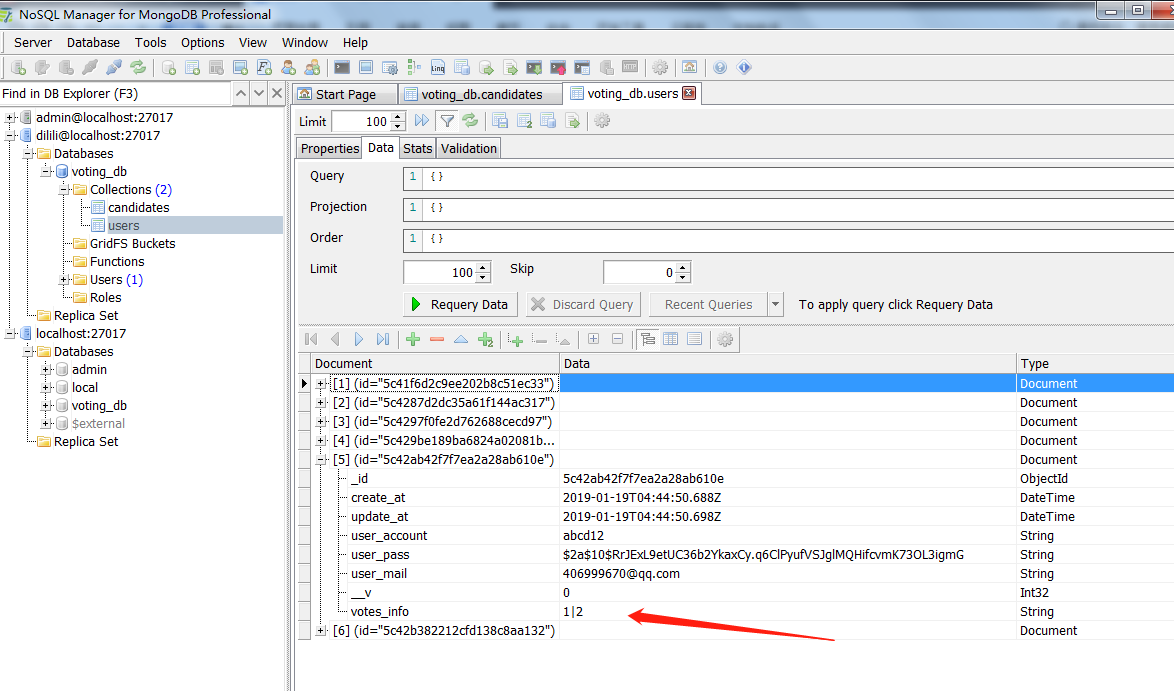


再查询（已经删除）



1. 数据的存储展示





1. 使用测试代码进行单元测试 效果图如下

