Seata 初步了解

**概念：**

**首先什么是事务？？**

事务（Transaction）是访问并可能更新数据库中各种数据项的一个程序执行单元，说人话就是更新数据库数据的时候用来提交语句的一个执行单元。

**事务应该具有4个属性？？**

原子性、一致性、隔离性、持久性。

**原子性：**一个事务要么执行成功，要么执行失败

**一致性：**完成多个数据库之间的一致，事务的中间状态不能被观察到的

**隔离性：**一个事务的执行不能被其它事务所干扰

**持久性：**一个事务一旦提交，它对数据库中数据的改变就应该是永久性的

常见的分布式事务解决方案

1、seata阿里分布式事务框架

2、消息队列

3、saga

4、XA

**分布式事务的四种模式：AT、TCC、Saga、XA；**

**AT模式分成两阶段工作：**

**一阶段：**

1. 拦截并解析业务SQL，找到需要在数据表中更新的数据，将其转换为undo\_log，并且保存到提前在每个数据库中创建的undo\_log表中

**二阶段：**

1. 因为第一阶段已经提交了本地事务，数据已经更新过了，这个时候如果没有报错，那么直接删除掉undo\_log以及行锁的数据即可，但是如果发生了报错，就只需要根据undo\_log来回退数据。

AT本地事务与全局事务 角色

**TM: Transaction Manager 事务管理器**

全局事务的管理者，或者说是全局事务的发起方，再通俗一点就是标注了@GlobalTransactional的方法所在的服务

**RM: Resources Manager 资源管理**

负责分支(本地)事务注册、提交和回滚。每个服务都是一个RM，负责本地事务的管理

**TC：Transaction Coordinate 事务协调器**

全局事务的协调者，TM,RM启动的时候要向TC注册，TM创建的时候要向TC申请一个全局事务ID，所以整个事务的把控是在TC中的，但是各自事务的管理是在RM中

## TCC模式

TCC模式需要用户根据自己的业务场景实现Try、Confirm和Cancel三个操作。事务发起方在一阶段执行Try方法，在二阶段提交执行Confirm方法，二阶段回滚执行Cancel方法。

**优点**：在整个过程中基本没有锁，性能更强。

**缺点**： 侵入性比较强，并且需要用户自己实现相关事务控制逻辑。

## Seata快速开始

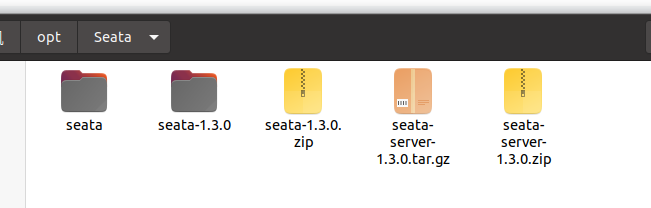
### Seata Server（TC）环境搭建

首先是版本选择

根据下面的表决定的seata版本

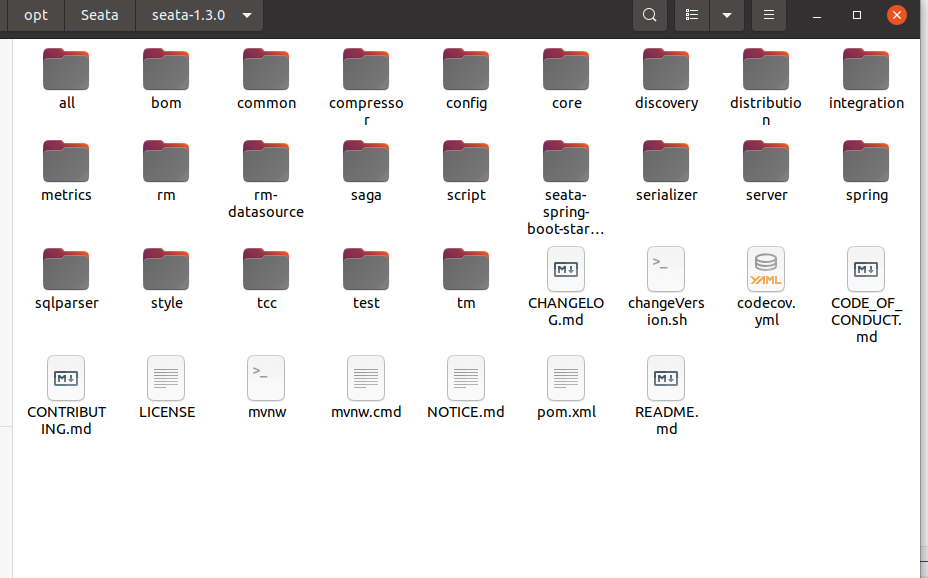


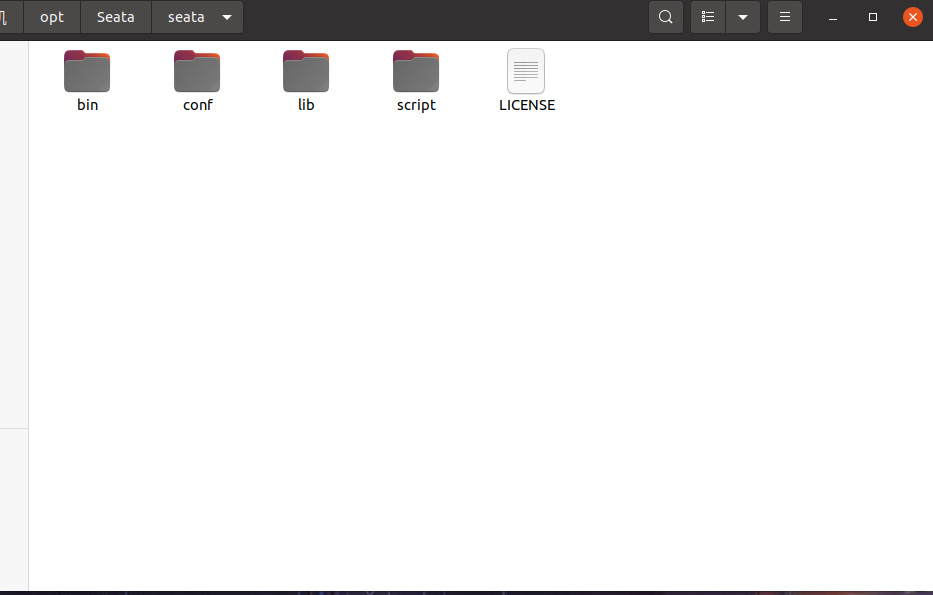
这个表中代表的就是对应的（本项目选用的是nacos2.0.3就使用seata1.3.0）



然后去官网找到seata-1.3.0 https://github.com/seata/seata/tree/1.3.0 下载下来

然后将seata-1.3.0中的scrpit移动到seata-server1.3.0中





**1.修改config.txt**

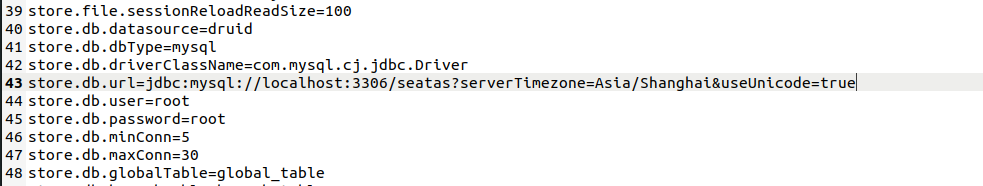
首先cd 到安装完成的目录下

然后

cd script/config-center

修改store.mode为db

store.db.dbType为mysql



注意store.db.url中的数据库需要自己先建立好

store.db.driverClassName这个为com.mysql.cj.jdbc.Driver

（注意：这个是mysql8.0以上的）

mysql5.0为com.mysql.jdbc.Driver

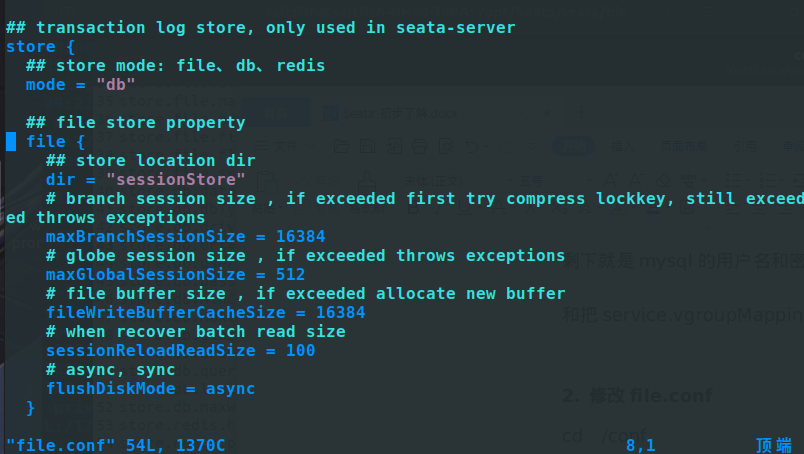
剩下就是mysql的用户名和密码

和把service.vgroupMapping.my\_test\_tx\_group=改为default

1. **修改file.conf**

cd /conf

sudo vim file.conf

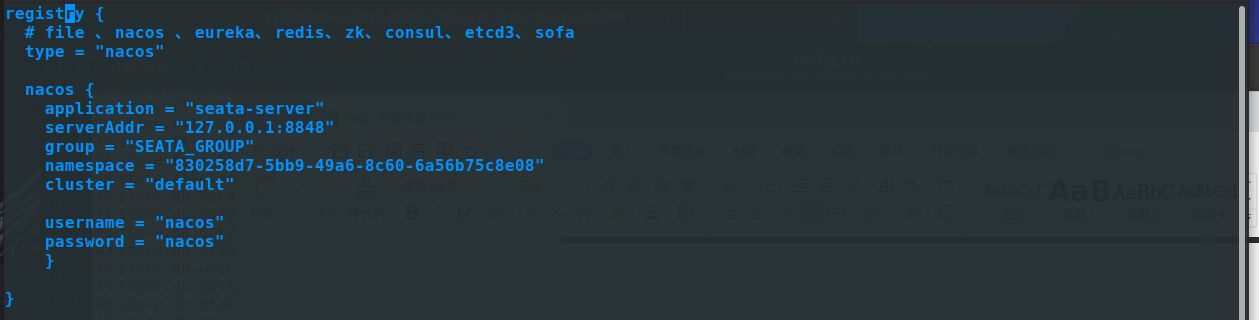


首先将mode改为db

修改下面的db为自己的密码或数据库



1. **修改registry.conf**





namespace就是nacos的命名空间这个可以不写

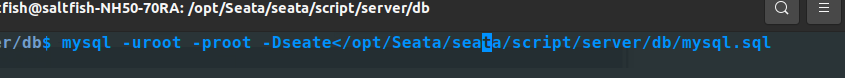
1. **执行对应的sql**

找到/script/server/db

执行下面的mysql.sql脚本

ubuntu命令执行sql脚本

mysql -u账号 -p密码 -D数据库名 < sql文件绝对路径



1. **完成nacos-config的配置**

cd ./script/config-center/nacos

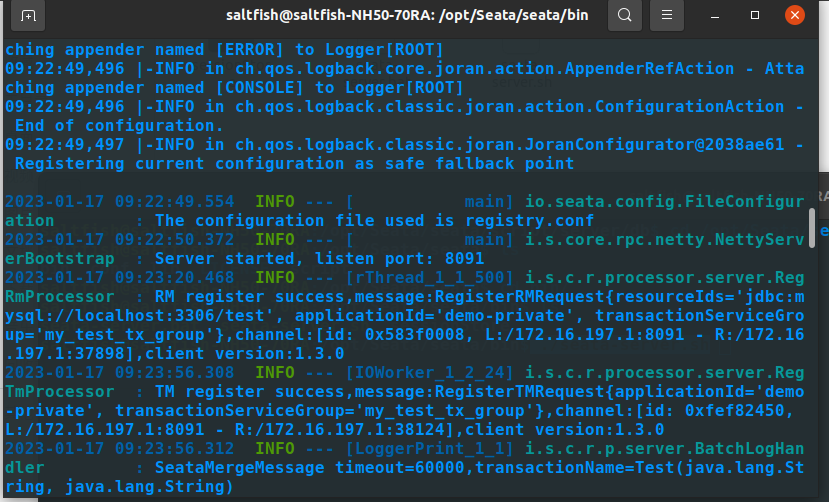
./nacos-config.sh

1. **启动seata-server**

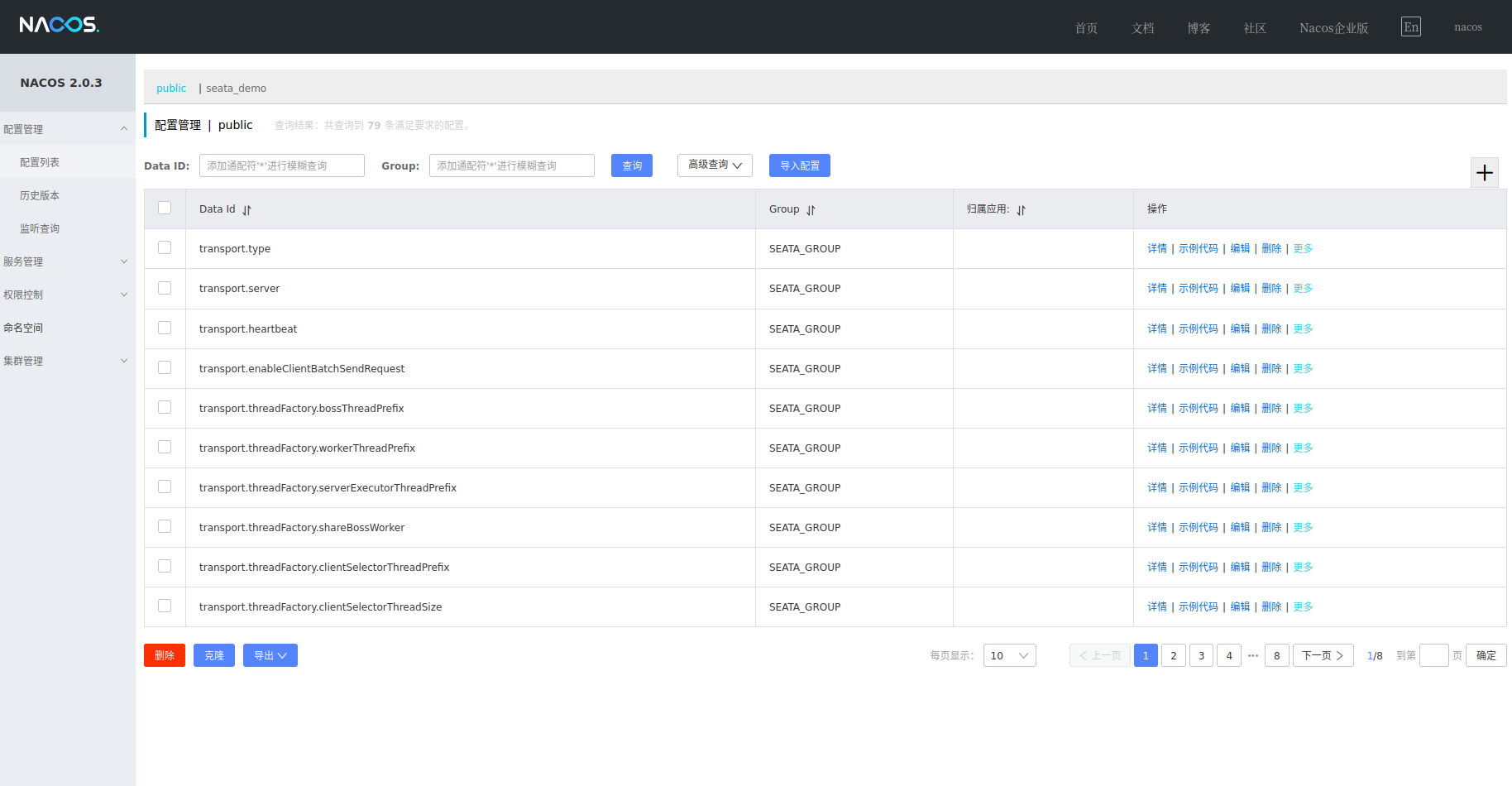
cd ./seata/bin

执行开启命令

./seata-server.sh



查看自己的nacos配置列表

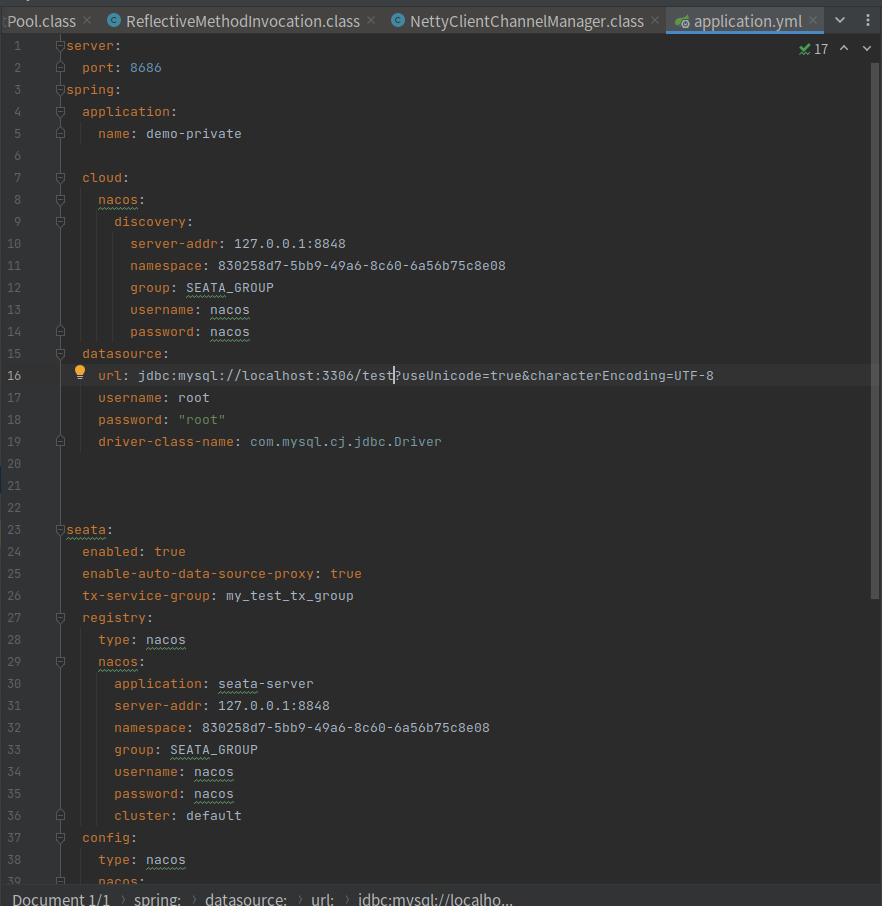


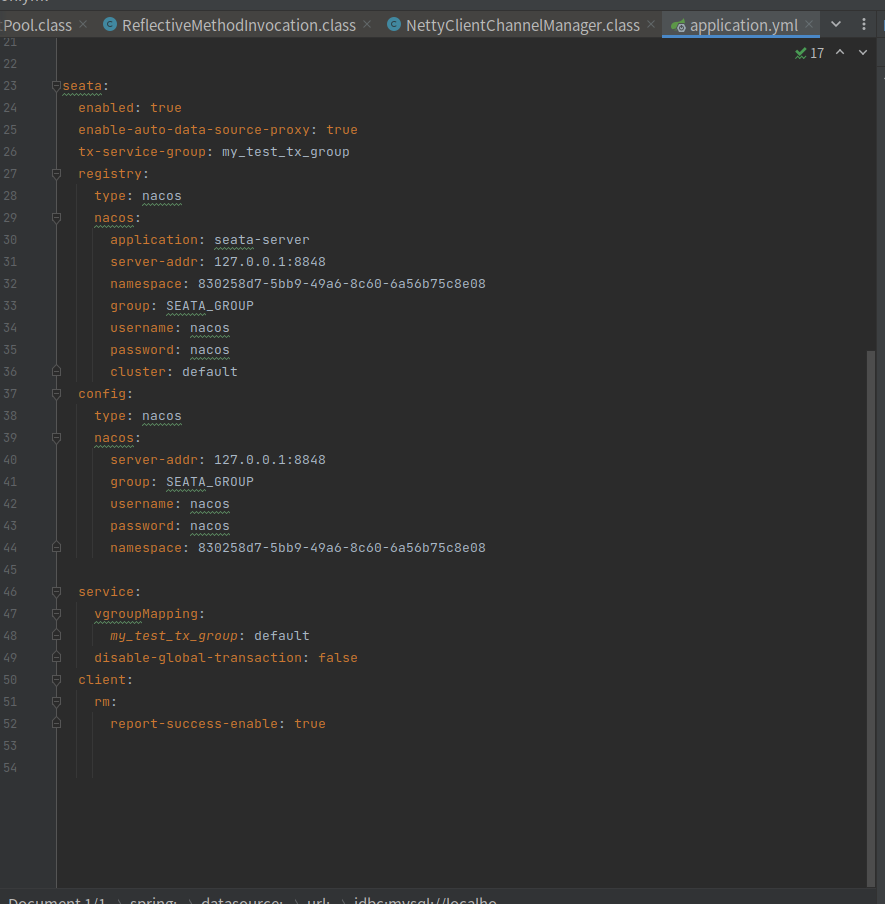
有这些之后就可以了

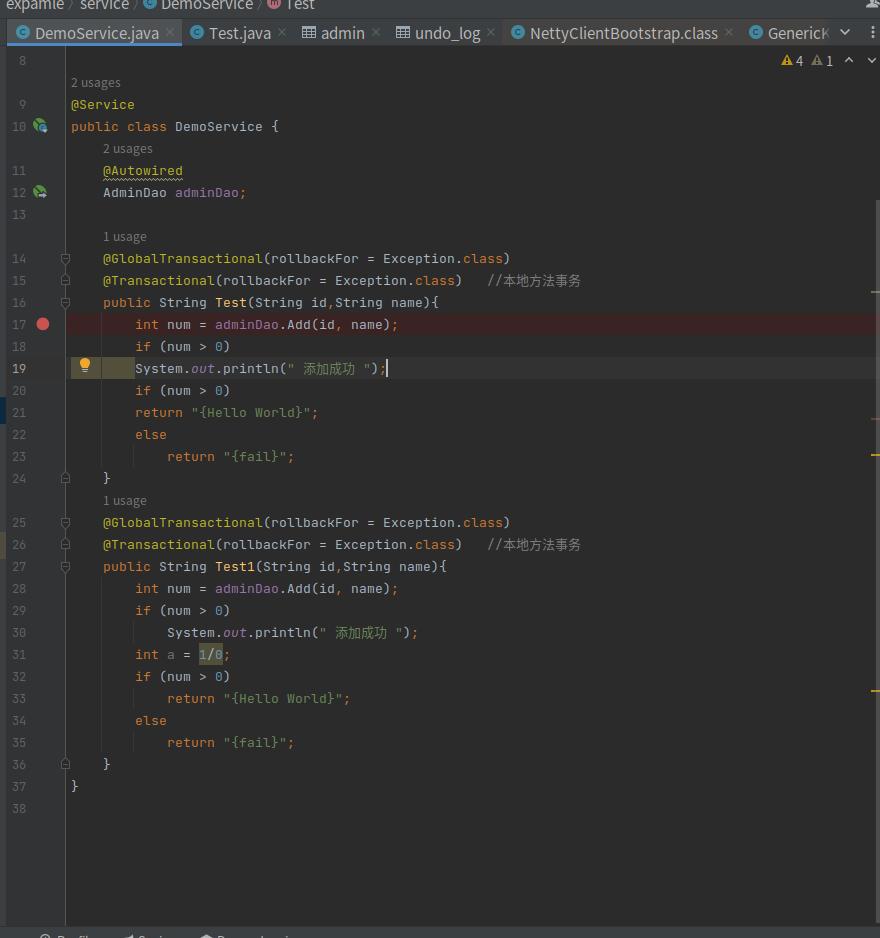
然后代码怎么写

这里举个例子

配置参考

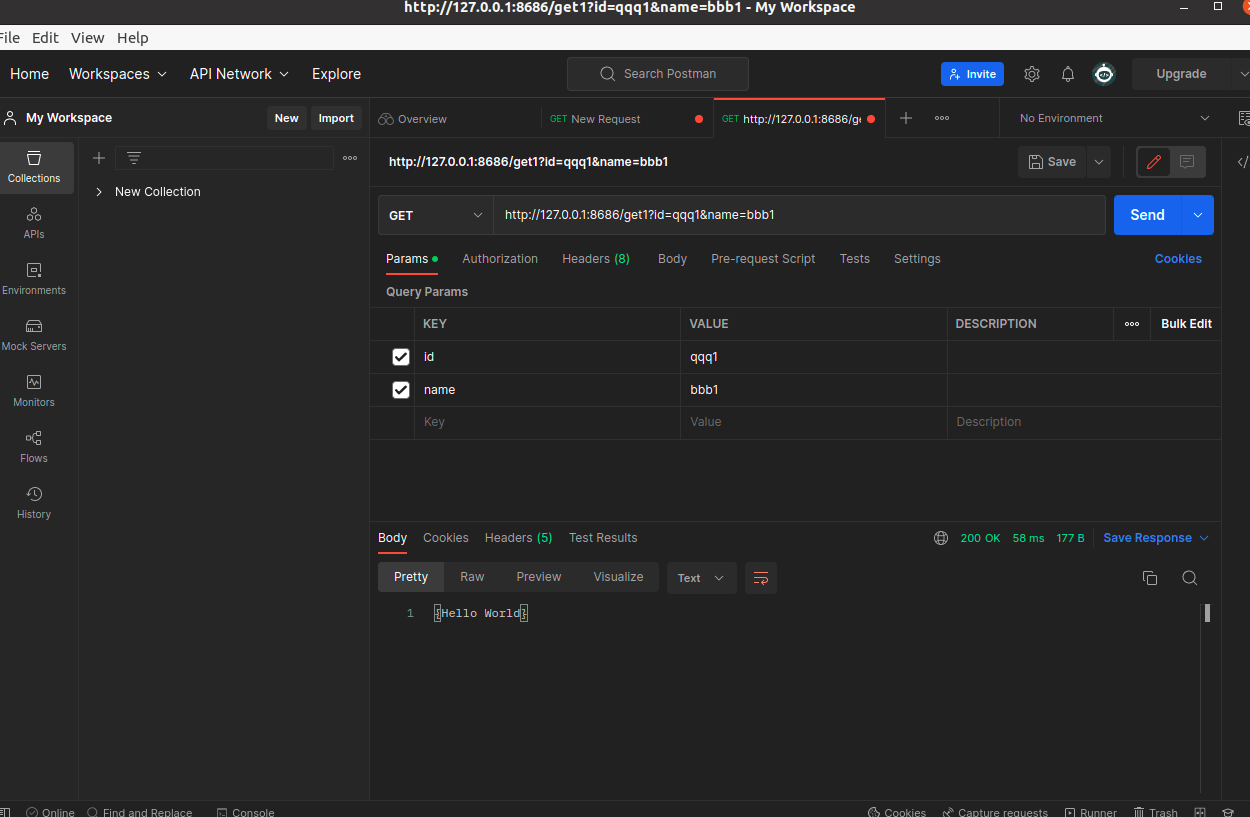




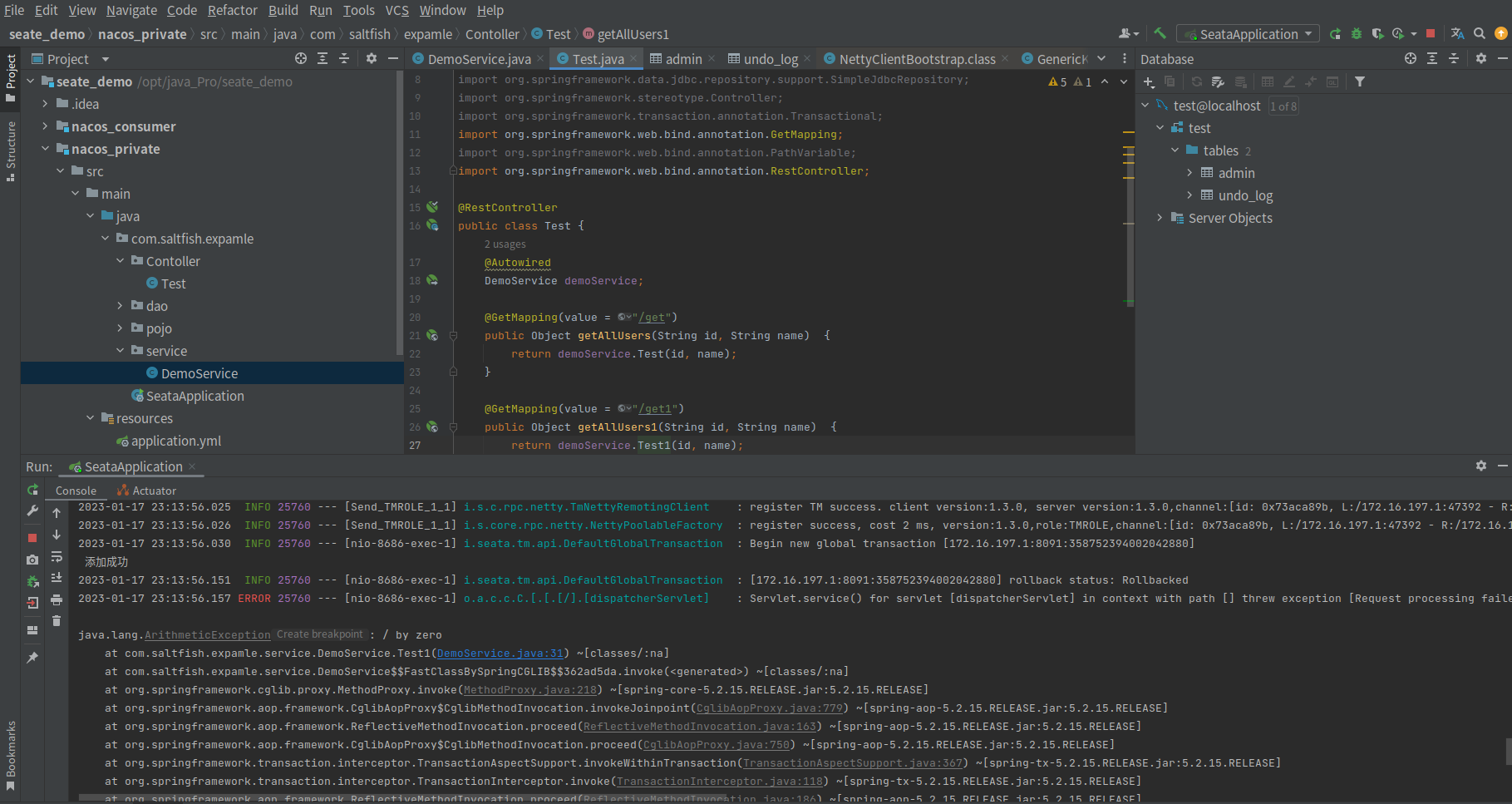


这里看到第二个Test1有一个1/0在添加完之后就造成了错误，这里开启事务管理就不会被添加到数据库里面

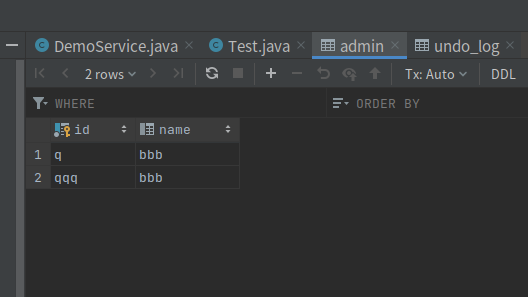
这里给一个请求



这里错误正常触发



看到数据库中并没有qqq1



这里的坑需要你去创建一个表（undo\_log）在你使用事务的地方

直接用命令行在mysql里面执行

下面的命令直接粘贴到mysql命令行就行

SET NAMES utf8mb4;

SET FOREIGN\_KEY\_CHECKS = 0;

-- ----------------------------

-- Table structure for undo\_log

-- ----------------------------

DROP TABLE IF EXISTS `undo\_log`;

CREATE TABLE `undo\_log` (

`branch\_id` bigint(20) NOT NULL COMMENT '分支事务ID',

`xid` varchar(100) CHARACTER SET utf8 COLLATE utf8\_general\_ci NOT NULL COMMENT '全局事务ID',

`context` varchar(128) CHARACTER SET utf8 COLLATE utf8\_general\_ci NOT NULL COMMENT '上下文',

`rollback\_info` longblob NOT NULL COMMENT '回滚信息',

`log\_status` int(11) NOT NULL COMMENT '状态，0正常，1全局已完成',

`log\_created` datetime(6) NOT NULL COMMENT '创建时间',

`log\_modified` datetime(6) NOT NULL COMMENT '修改时间',

UNIQUE INDEX `ux\_undo\_log`(`xid`, `branch\_id`) USING BTREE

) ENGINE = InnoDB CHARACTER SET = utf8 COLLATE = utf8\_general\_ci COMMENT = 'AT transaction mode undo table' ROW\_FORMAT = Compact;

SET FOREIGN\_KEY\_CHECKS = 1;