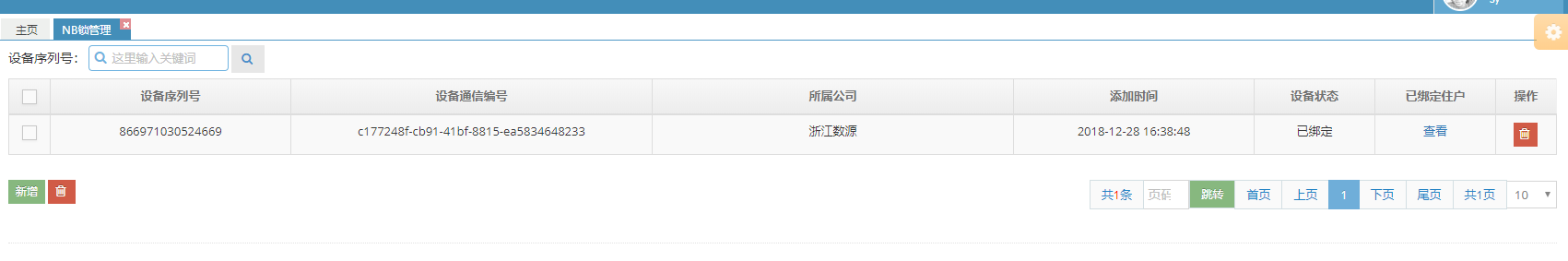


中正锁管理:

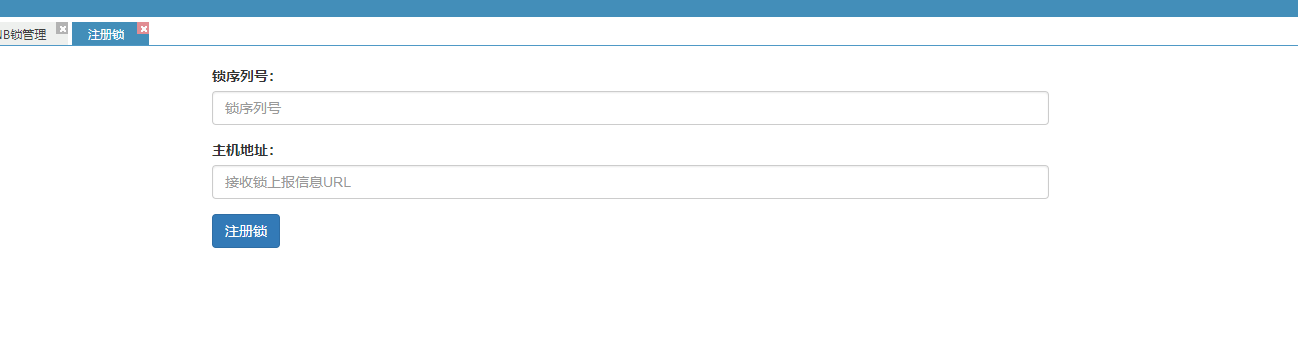
新增:



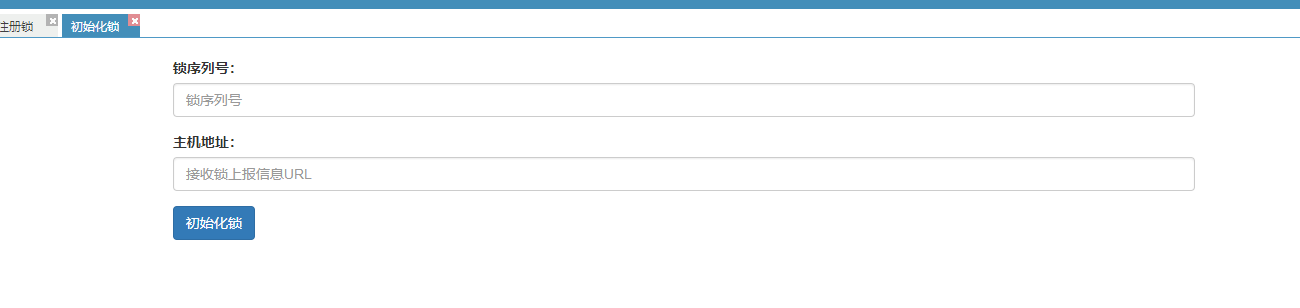
查询:



注册锁:



初始化锁:



Map.put() //键值对格式存储参数

HttpClient client=new HttpClient(请求的 url 地址);

client.setHttps(true);//是否是 https 协议

client.setXmlParam(xmlParam);//发送的 xml 数据

client.post();//执行 post 请求

String result = client.getContent(); //获取结果

TxManager:事务管理器

CAP: consistency, availability, partition-tolerance

BASE: Basically Available, Soft-state, Eventual Consistency

XA:

两阶段: 准备阶段(投票阶段)和提交阶段（执行阶段）

LCN:(lock, confirm, notify) 分布式框架

事务补偿机制，服务故障或挂机再启动时可恢复事务

创建事务组:发起方(调用者)在代码执行之前通知TxManager创建事务组,拿到groupid

添加事务组:参与方执行完代码,将参与方(被调用方)的事务信息添加到TxManager的事务组

关闭事务组:发起方执行完所有代码,将执行结果通知TxManager. TxManager会根据事务组中的信息来通知各参与方模块的提交或回滚.

Lcn框架依赖redis实现事务组队事务信息的存储

事务补偿机制: 事务补偿机制是在执行某个业务方法时,本应该执行成功的操作,因为服务器挂机或者网络等问题导致事务没有正常提交,这种场景就需要通过补偿来完成事务,从而达到事务的一致性.

补偿机制的触发条件？

当执行关闭事务组步骤时，若发起方接受到失败的状态后将会把该次事务识别为待补偿事务，然后发起方将该次事务数据异步通知给TxManager。TxManager接受到补偿事务以后先通知补偿回调地址，然后再根据是否开启自动补偿事务状态来补偿或保存该次切面事务数据。

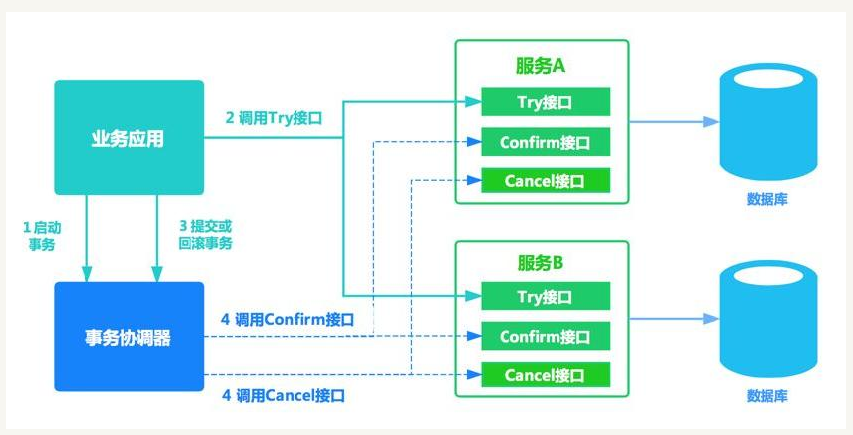
TCC:(try,confirm, cancel) 实现最终一致性

一个接口分成三个逻辑(三段代码): try , confirm, cancel

Try: 执行try预操作逻辑代码

Confirm: 当try成功,tcc框架会感知并执行confirm逻辑代码

Cancel: 当try抛出异常,tcc框架感知并执行cancel逻辑代码回滚



TXC: (Taobao Transaction Constructor)

Mvn install:install-file

-DgroupId=com.codingapi.txlcn

-DartifactId=txlcn-tm -Dversion=5.0.2

-Dfile=D:\java\IdeaProjects\tx-lcn\txlcn-tm\target\txlcn-tm-5.0.2.RELEASE.jar

-Dpackaging=jar