《33\_SpringCloudAlibaba之Seata分布式事务原理剖析（2）》

TCC框架，Saga框架，Seata支持好几种分布式事务方案，全面、完善的分布式事务解决方案，成熟框架

TCC方案，ByteTCC，搜索一下，维护这个框架的，QQ群

你的核心接口，全部要写三套，每个接口都要拆分为3个接口，Try、Commit、Cancel，一个接口，chooseCandidate接口，tryChooseCandidate，commitChooseCandidate，cancelChooseCandidate，每个接口要写3套逻辑

try里面，可以在数据库添加一批评审员，状态都设置为INVALID，无效的；commit，就可以把他们的状态设置为VALID；cancel，就可以把插入的那批评审员给删除了

分布式事务这个技术以及思想，过去在国内一直没有受到重视的，大厂还是小公司，只要你不是金融级跟钱直接相关的系统，普通的互联网系统，哪怕是订单系统，都不会上分布式事务，TCC思想，太麻烦了

Atomikos框架，但是在国内也是很少很少用的

Seata，支持多种分布式事务方案：TCC、XA、AT、Saga

TCC而言，你不需要写TCC三个接口的，你的业务代码，就跟以前是一样的，就是一个接口，你的接口以前是什么样子的，现在也可以是什么样的，接口都没有任何的变化的话，分布式事务到底是怎么做的

Saga，在国内很少很少用，长事务，N多个服务串联在一起执行，补偿

TC、TM、RM

TC，Seata自己独立部署的一个server，他用于全面的管理每个分布式事务；TM，用于对单个分布式事务进行管理和注册；RM，是对一个分布式事务内的每个服务本地分支事务进行管理的

分布式事务框架的角度来思考，TM这个东西他是怎么运作起来的

TM这个东西，完全是可以基于一个注解来进行驱动的，Spring，AOP机制，切面机制，Spring技术体系里，还有一些其他的拦截机制，监听机制，回调机制，完全可以针对TM设计一个注解，@ShishanTransaction，加在你的举报服务的Service组件的方法上，依托Spring的机制去对方法调用做一个拦截，如果你发现这个方法加了你指定的分布式事务的注解

提取一下本次请求里带的一些请求头或者是请求附加的内置参数，有没有一个全局事务id，xid，这个带着@ShishanTransaction注解的方法，他其实是一个分布式事务的起始方法，TM这样的一个组件的业务逻辑就可以开始运作起来了

引入依赖，还会在spring boot配置文件里配置一下分布式事务的一些配置，分布式事务server端的地址，暴露出来的都是RESTful API接口，基于HTTP请求就可以了

TM如果要找TC注册一个全局事务，此时就可以通过HTTP通信组件，发送HTTP请求到指定地址的TC server的接口就可以了，TC server可以注册一个全局事务，生成一个唯一的txid，返回给你的TM

RM如何可以拦截你的本地数据库的操作呢？

代理你的数据源，操作数据库，必须要有一个数据库连接，JDBC接口规范里就是一个Connection对象，数据库连接池，Druid、C3P0、DBCP，维护一个数据库连接池，一定会从数据库连接池里获取一个数据库连接

依托这个数据库连接去对数据库执行增删改的操作

你可以针对你的数据库连接去做一个代理，也就是说，业务系统拿到的数据库连接是被你代理过的，他基于你的代理数据库连接执行增删改操作，代码会先执行到你手上，此时你就可以做一些操作了

增删改的语句，执行一些查询，DELETE语句，UPDATE，生成一个逆向的UPDATE语句，想要把一个字段改成1100，900，生成一个INSERT语句

在一个本地事务里，让他执行增删改，把你生成的undo log插入到数据库的undo\_log表里去，发送HTTP请求到TC去注册一个分支事务，提交本地事务，把增删改操作和undo log插入都放一个本地事务里，他们会一起成功或者失败