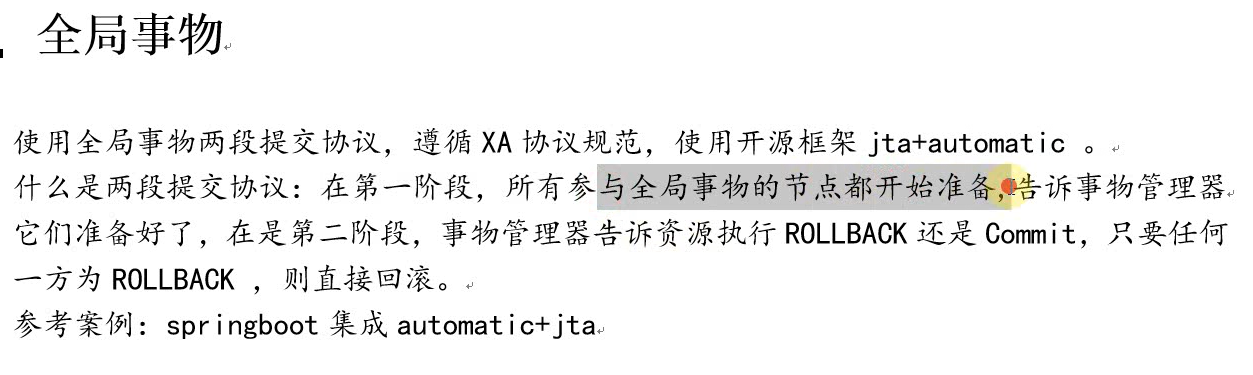
分布式事务



# 1 Fescar

下载Fescar-server ,修改register.conf 和 file.conf

## 启动项目

linux下：

sh fescar-server.sh 8091 /home/admin/fescar/data/

win操作系统直接启动

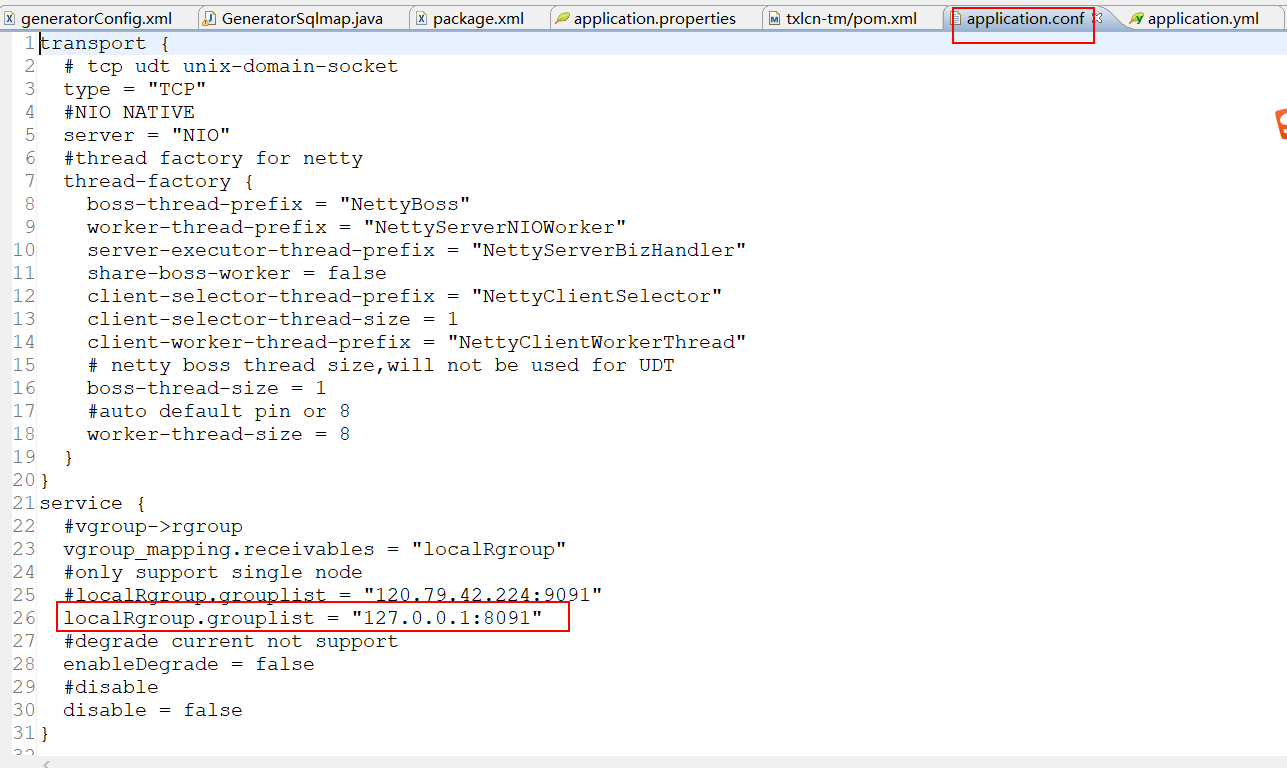
fescar-server.bat

## 启动zookeeper

## client搭建：

查看 自带示例：

整合springboot需要修改的地方：

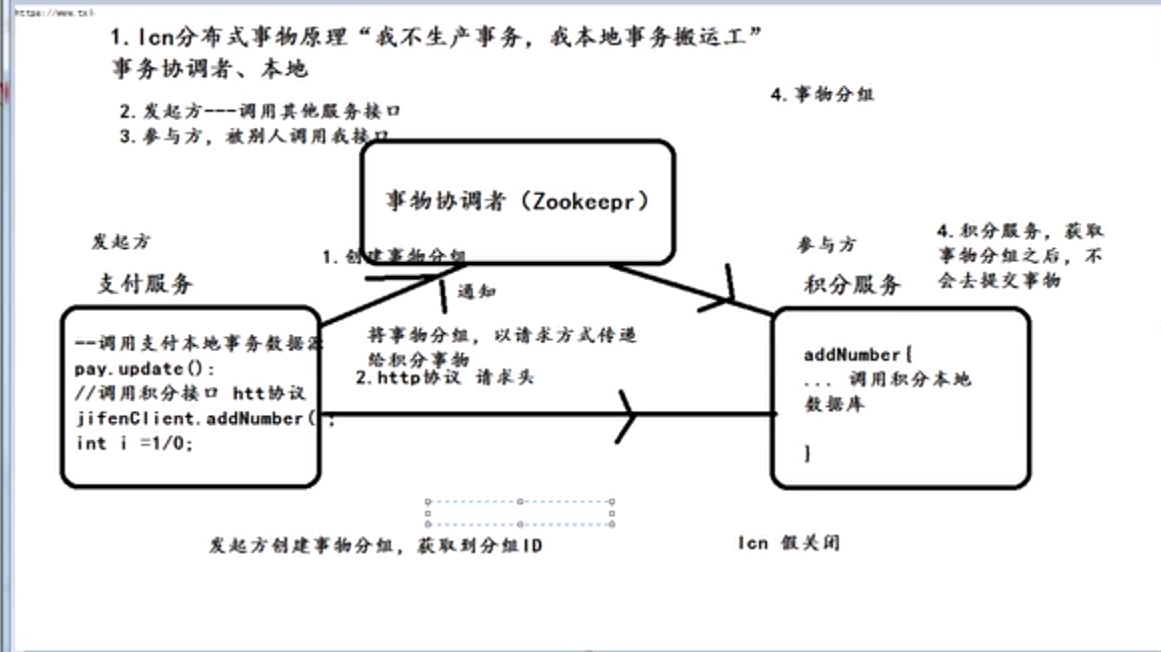




## 注意事项

所有需要全局事务的地方都要加上

@GlobalTransactional 否则会不被当作全局事务处理



# 2 LCN 5.x使用

## 2.1配置TM （5.0.2）

### 2.1.1操作前准备

需要redis 和 mysql（创建好表格）

运行TM之前要先启动redis和mysql。

sql表格

|  |
| --- |
| /\*  Navicat Premium Data Transfer  Source Server : local  Source Server Type : MySQL  Source Server Version : 100309  Source Host : localhost:3306  Source Schema : tx-manager  Target Server Type : MySQL  Target Server Version : 100309  File Encoding : 65001  Date: 29/12/2018 18:35:59  \*/  CREATE DATABASE IF NOT EXISTS `tx-manager` DEFAULT CHARSET utf8 COLLATE utf8\_general\_ci;  USE `tx-manager`;  SET NAMES utf8mb4;  SET FOREIGN\_KEY\_CHECKS = 0;  -- ----------------------------  -- Table structure for t\_tx\_exception  -- ----------------------------  DROP TABLE IF EXISTS `t\_tx\_exception`;  CREATE TABLE `t\_tx\_exception` (  `id` bigint(20) NOT NULL AUTO\_INCREMENT,  `group\_id` varchar(64) CHARACTER SET utf8mb4 COLLATE utf8mb4\_general\_ci NULL DEFAULT NULL,  `unit\_id` varchar(32) CHARACTER SET utf8mb4 COLLATE utf8mb4\_general\_ci NULL DEFAULT NULL,  `mod\_id` varchar(128) CHARACTER SET utf8mb4 COLLATE utf8mb4\_general\_ci NULL DEFAULT NULL,  `transaction\_state` tinyint(4) NULL DEFAULT NULL,  `registrar` tinyint(4) NULL DEFAULT NULL,  `ex\_state` tinyint(4) NULL DEFAULT NULL COMMENT '0 待处理 1已处理',  `remark` varchar(10240) NULL DEFAULT NULL COMMENT '备注',  `create\_time` datetime(0) NULL DEFAULT NULL,  PRIMARY KEY (`id`) USING BTREE  ) ENGINE = InnoDB AUTO\_INCREMENT = 967 CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_general\_ci ROW\_FORMAT = Dynamic;  SET FOREIGN\_KEY\_CHECKS = 1; |

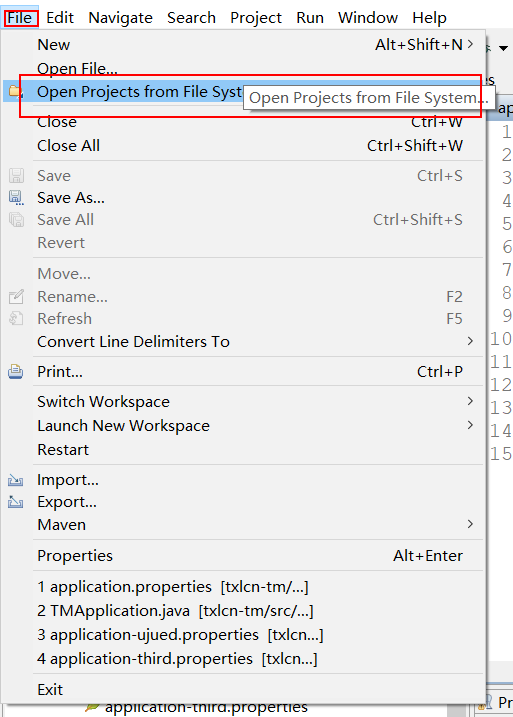
### 2.1.2下载源码包：

<https://github.com/codingapi/tx-lcn/releases>

### 2.1.3解压如下：



### 2.1.4将源码导入MyEclipse：



### 2.1.5可能存在问题：

源码中大量的log无法通过@Slf4j注解找到log变量，而且大量的set/get方法注解也都失效了。

### 2.1.6解决：

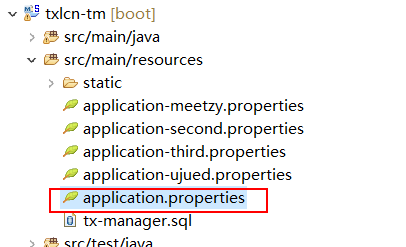
1 下载lombok.jar

https://projectlombok.org/download.html

**2. 将 lombok.jar 复制到 myeclipse.ini / eclipse.ini /sts.ini所在的文件夹目录下  
3. 打开 eclipse.ini / myeclipse.ini/sts.ini，在最后面插入以下两行并保存：  
    -Xbootclasspath/a:lombok.jar  
     -javaagent:lombok.jar       
4.重启 eclipse / myeclipse/STS**

### 2.1.7重启后仍无效，则删除项目，重新导入即可。

### 2.1.8配置配置文件



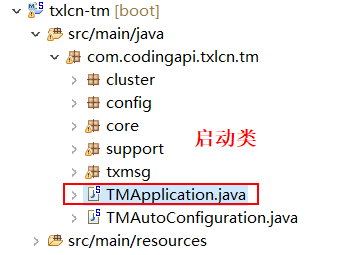
#### 2.1.8.1基本内容：

|  |
| --- |
| spring.application.name=TransactionManager  server.port=7970  #数据库配置  spring.datasource.driver-class-name=com.mysql.jdbc.Driver  spring.datasource.url=jdbc:mysql://127.0.0.1:3306/tx-manager?characterEncoding=UTF-8  spring.datasource.username=root  spring.datasource.password=xxxx  #方言配置  spring.jpa.database-platform=org.hibernate.dialect.MySQL5InnoDBDialect  spring.jpa.hibernate.ddl-auto=update  #Redis配置  spring.redis.host=192.168.153.136  spring.redis.port=6379  spring.redis.password= |

#### 2.1.8.2官方示例：

|  |
| --- |
| **spring.application.name**=TransactionManager  **server.port**=7970  *# JDBC 数据库配置*  **spring.datasource.driver-class-name**=com.mysql.jdbc.Driver  **spring.datasource.url**=jdbc:mysql://127.0.0.1:3306/tx-manager?characterEncoding=UTF-8  **spring.datasource.username**=root  **spring.datasource.password**=123456  *# 数据库方言*  **spring.jpa.database-platform**=org.hibernate.dialect.MySQL5InnoDBDialect  *# 第一次运行可以设置为: create, 为TM创建持久化数据库表*  **spring.jpa.hibernate.ddl-auto**=validate  *# TM监听IP. 默认为 127.0.0.1*  **tx-lcn.manager.host**=127.0.0.1  *# TM监听Socket端口. 默认为 ${server.port} - 100*  **tx-lcn.manager.port**=8070  *# 心跳检测时间(ms). 默认为 300000*  **tx-lcn.manager.heart-time**=300000  *# 分布式事务执行总时间(ms). 默认为36000*  **tx-lcn.manager.dtx-time**=8000  *# 参数延迟删除时间单位ms 默认为dtx-time值*  **tx-lcn.message.netty.attr-delay-time**=${tx-lcn.manager.dtx-time}  *# 事务处理并发等级. 默认为机器逻辑核心数5倍*  **tx-lcn.manager.concurrent-level**=160  *# TM后台登陆密码，默认值为codingapi*  **tx-lcn.manager.admin-key**=codingapi  *# 分布式事务锁超时时间 默认为-1，当-1时会用tx-lcn.manager.dtx-time的时间*  **tx-lcn.manager.dtx-lock-time**=${tx-lcn.manager.dtx-time}  *# 雪花算法的sequence位长度，默认为12位.*  **tx-lcn.manager.seq-len**=12  *# 异常回调开关。开启时请制定ex-url*  **tx-lcn.manager.ex-url-enabled**=false  *# 事务异常通知（任何http协议地址。未指定协议时，为TM提供内置功能接口）。默认是邮件通知*  **tx-lcn.manager.ex-url**=/provider/email-to/\*\*\*@\*\*.com  *# 开启日志,默认为false*  **tx-lcn.logger.enabled**=true  **tx-lcn.logger.enabled**=false  **tx-lcn.logger.driver-class-name**=${spring.datasource.driver-class-name}  **tx-lcn.logger.jdbc-url**=${spring.datasource.url}  **tx-lcn.logger.username**=${spring.datasource.username}  **tx-lcn.logger.password**=${spring.datasource.password}  *# redis 的设置信息. 线上请用Redis Cluster*  **spring.redis.host**=127.0.0.1  **spring.redis.port**=6379  **spring.redis.password**=  注意（NOTE）  (1) TxManager所有配置均有默认配置，请按需覆盖默认配置。  (2) 特别注意 TxManager进程会监听两个端口号，一个为TxManager端口，另一个是事务消息端口。TxClient默认连接事务消息端口是8070， 所以，为保证TX-LCN基于默认配置运行良好，请设置TxManager端口号为8069 或者指定事务消息端口为8070  (3) 分布式事务执行总时间 a 与 TxClient通讯最大等待时间 b、TxManager通讯最大等待时间 c、微服务间通讯时间 d、微服务调用链长度 e 几个时间存在着依赖关系。 a >= 2c + (b + c + d) \* (e - 1), 特别地，b、c、d 一致时，a >= (3e-1)b。你也可以在此理论上适当在减小a的值，发生异常时能更快得到自动补偿，即 a >= (3e-1)b - Δ（[原因](https://www.txlcn.org/zh-cn/docs/fqa.html)）。 最后，调用链小于等于3时，将基于默认配置运行良好  (4) 若用tx-lcn.manager.ex-url=/provider/email-to/xxx@xx.xxx 这个配置，配置管理员邮箱信息(如QQ邮箱)：  **spring.mail.host**=smtp.qq.com  **spring.mail.port**=587  **spring.mail.username**=xxxxx@\*\*.com  **spring.mail.password**=\*\*\*\*\*\*\*\*\* |

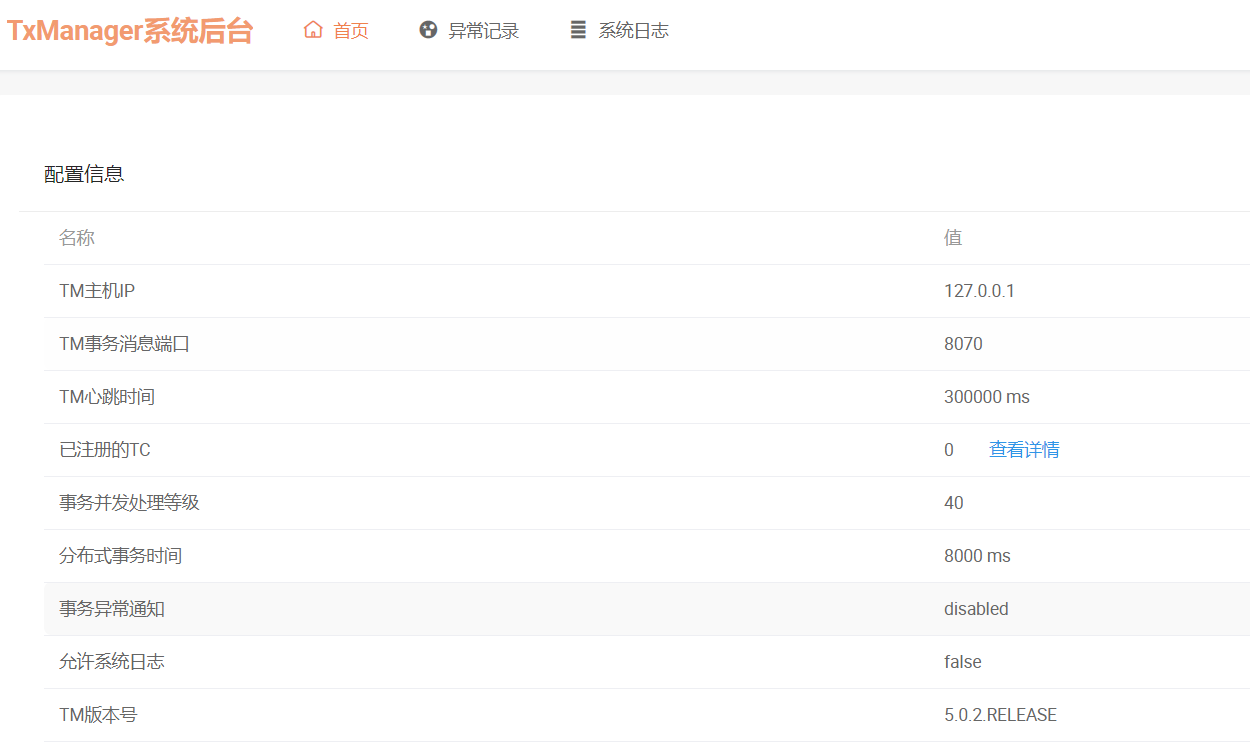
### 2.1.9项目启动



### 2.1.10 访问：<http://localhost:7970>

初始密码：codingapi

如下成功



## 2.2从源码编译打包为jar

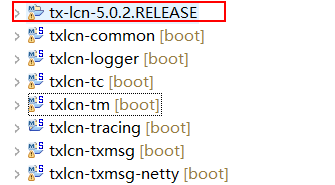
1 解压获取到的源码包，并成功导入到myeclipse中。

2 进入 txlcn-tm项目中，修改pom.xml 的打包插件：

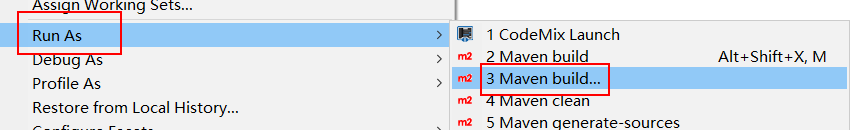


3 解开这个注解，并注释掉docker的那个插件（图片中未显示出来，初始时事那个）。

4 不要单独打包这个项目，容易出现bug，直接右键tx-lcn-5.0.2.RELEASE，对整个源码包进行打包。

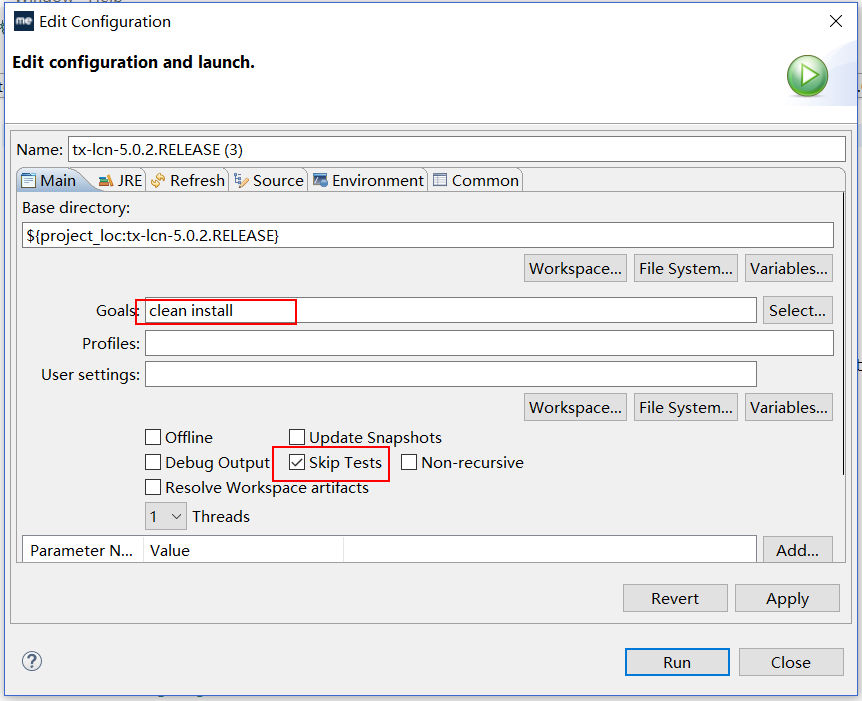


5 右键tx-lcn 选则：

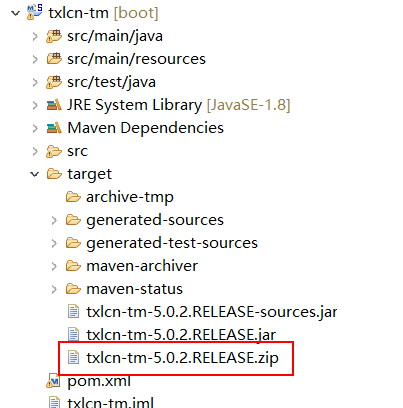


6 输入任务代码 ： clean install

7 同时勾选跳过测试test，点击运行即可（可能需要等待一段时间）



8 打包结束后，在txlcn-tm的target下会多出来几个文件，其中txlcn-tm-5.0.2.zip就是我们需要的文件，她包含了TM的单独项目源码，jar包和源码jar包，三个部分。jar包可直接运行。（运行前先启动reids和mysql）



## 2.2配置TC

### 2.2.1TC引入pom依赖

注意：不能直接引用Maven仓库中的依赖，目前仓库中的5.0.2版本依赖有bug。

导致DataSourceAspect 创建bean的name冲突。

所以要直接引用5.0.2 源码包中的下面两个依赖。（将上述打包好的jar包，放入本地仓库就行了）

<dependency>

<groupId>com.codingapi.txlcn</groupId>

<artifactId>txlcn-tc</artifactId>

<version>5.0.2.RELEASE</version>

</dependency>

<dependency>

<groupId>com.codingapi.txlcn</groupId>

<artifactId>txlcn-txmsg-netty</artifactId>

<version>5.0.2.RELEASE</version>

</dependency>

### 2.2.2TC开启分布式事务注解

在主类上使用@EnableDistributedTransaction

**@SpringBootApplication**

**@EnableDistributedTransaction**

**public** **class** **DemoAApplication** {

**public** **static** **void** **main**(String[] args) {

SpringApplication.run(DemoDubboClientApplication.class, args);

}

}

### 2.2.3TC微服务A业务方法配置

**@Service**

**public** **class** **ServiceA** {

**@Autowired**

**private** ValueDao valueDao; *//本地db操作*

**@Autowired**

**private** ServiceB serviceB;*//远程B模块业务*

**@LcnTransaction** *//分布式事务注解*

**@Transactional** *//本地事务注解*

**public** String **execute**(String value) **throws** BusinessException {

*// step1. call remote service B*

String result = serviceB.rpc(value); *// (1)*

*// step2. local store operate. DTX commit if save success, rollback if not.*

valueDao.save(value); *// (2)*

valueDao.saveBackup(value); *// (3)*

**return** result + " > " + "ok-A";

}

}

### 2.2.4TC微服务B业务方法配置

**@Service**

**public** **class** **ServiceB** {

**@Autowired**

**private** ValueDao valueDao; *//本地db操作*

**@LcnTransaction** *//分布式事务注解*

**@Transactional** *//本地事务注解*

**public** String **rpc**(String value) **throws** BusinessException {

valueDao.save(value); *// (4)*

valueDao.saveBackup(value); *// (5)*

**return** "ok-B";

}

}

### 2.2.5TC配置信息说明

# 默认之配置为TM的本机默认端口

tx-lcn.client.manager-address=127.0.0.1:8070

# 3 MQ实现分布式事务



