Atitit 单元测试修改

目录

[1. 单元测试目录新增tset目录 maven结构 1](#_Toc15109)

[2. 测试数据源问题 1](#_Toc22144)

[3. Quarz对配置干扰问题，全部移除 1](#_Toc17527)

[3.1. 日志问题，将log4j。Xml 1](#_Toc9484)

[4. Redis 问题 1](#_Toc205)

# 单元测试目录新增tset目录 maven结构

使用单独对apptest。Xml测试。。

# 测试数据源问题

修改 spring mybatis 新增加测试数据源，修改指向到此数据源

# Quarz对配置干扰问题，全部移除

## 日志问题，将log4j。Xml

挪到更目录解决

# Redis 问题

redis中 Could not get a resource from the pool

java.util.NoSuchElementException: Unable to validate object

调整了redis配置但是不生效貌似 /quartz/src/test/resources/spring/spring-redis.xml

配置繁琐，重新下载redis使用默认配置 ，还是不生效

简单测试是ok的

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

//连接本地的 Redis 服务

Jedis jedis = **new** Jedis("localhost");

// 如果 Redis 服务设置来密码，需要下面这行，没有就不需要

// jedis.auth("123456");

System.***out***.println("连接成功");

//查看服务是否运行

System.***out***.println("服务正在运行: "+jedis.ping());

}

修改类下代码可以了。。。

**private** **static** ShardedJedis getJedis\_ShardedJedis() {

**try** {

**return** *shardedJedisPool*.getResource();

} **catch** (Exception e) { //for test

// 池基本配置

JedisPoolConfig config = **new** JedisPoolConfig();

// config.setMaxActive(20);

// config.setMaxIdle(5);

// config.setMaxWait(1000l);

config.setTestOnBorrow(**false**);

// slave链接

List<JedisShardInfo> shards = **new** ArrayList<JedisShardInfo>();

shards.add(**new** JedisShardInfo("127.0.0.1", 6379, "master"));

// 构造池

ShardedJedisPool shardedJedisPool = **new** ShardedJedisPool(config, shards);

**return** shardedJedisPool.getResource();

}

}

**private** **static** <T> T exceute(IRedisFunction<ShardedJedis, T> fun) {

ShardedJedis shardedJedis = **null**;

**try** {

// 从连接池中获取到jedis分片对象

shardedJedis = *getJedis\_ShardedJedis*();

**return** fun.callBack(shardedJedis);

} **catch** (Exception e) {

***log***.error(CacheHelper.**class**, e);

} **finally** {

**if** (**null** != shardedJedis) {

shardedJedis.close();

}

}

**return** **null**;

}