Java商城秒杀系统设计与实战

作者：debug
花名：阿修罗

时间：2019-06-12

追求技术，热爱分享！相信技术改变生活，技术成就梦想！

以实际业务场景为出发点，撸码实战为主，理论概念为辅，真正将讲解的理论知识要点用代码实战体现出来

各位小伙伴在学习实战的过程中，debug将全程与你交流，共同成长进步！

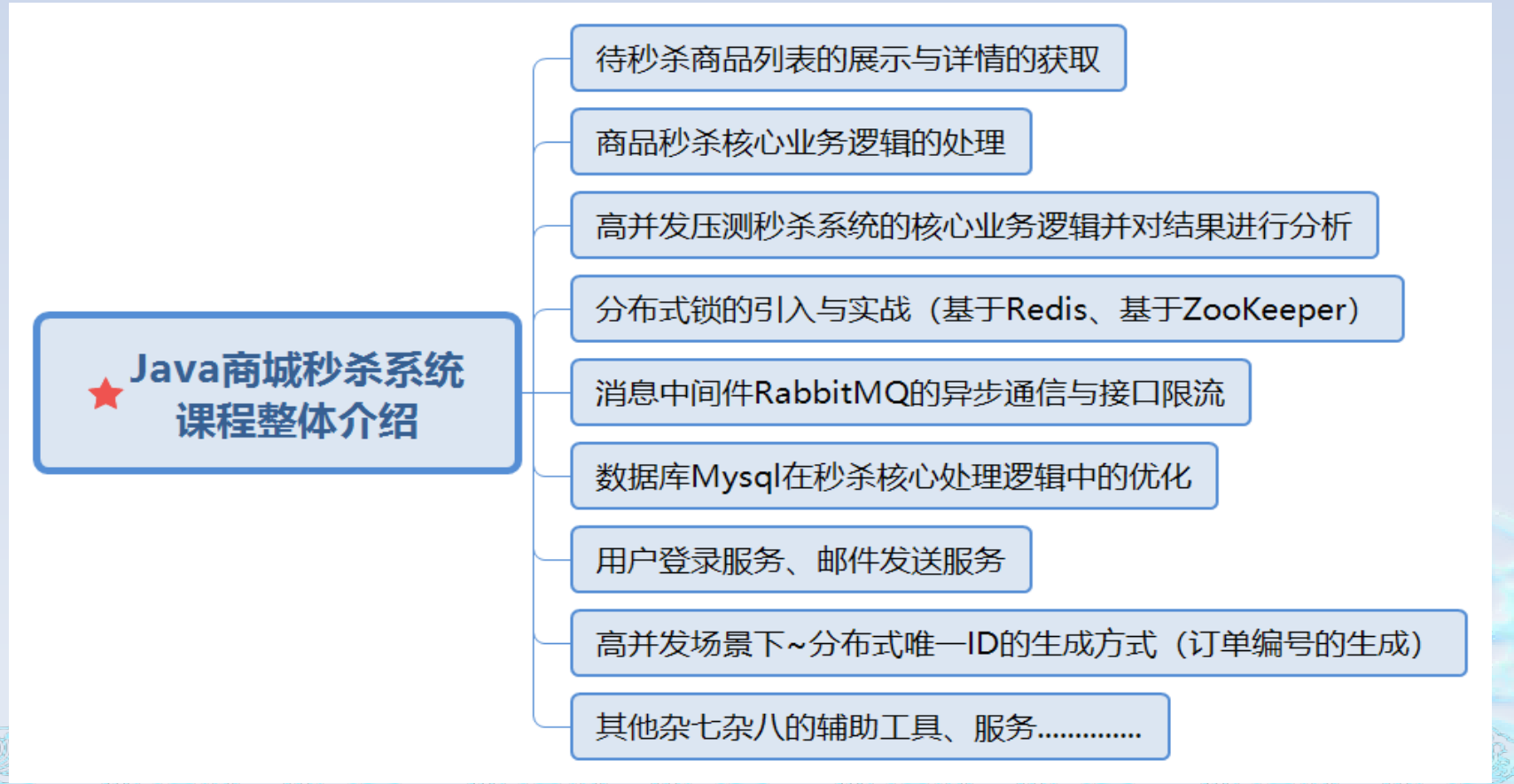
实战分享自己项目的开发经验与踩过的坑,分享自己解决问题的思路与方案!

Java商城秒杀系统设计与实战

作者：debug

课程整体介绍

介绍分享一种当下比较盛行的、典型而且常见的高并发业务场景，即商城（电商平台）商品秒杀系统的设计与实战



Java商城秒杀系统设计与实战

作者：debug

核心技术列表

Java商城秒杀系统设计与实战

作者：debug

课程要求与收益

要求：本课程属于SpringBoot+分布式锁+消息中间件等技术栈的项目实战课程，故而需要相应的技术储备

建议：可以考虑购买SpringBoot+分布式锁+消息中间件+本课程的学习套餐，从而可以更好的学习本课程

Java商城秒杀系统设计与实战

作者：debug

系统的整体演示

（1）由于秒杀系统用到了Redis、ZooKeeper跟RabbitMQ中间件，故而需要在本地安装启动相应的服务

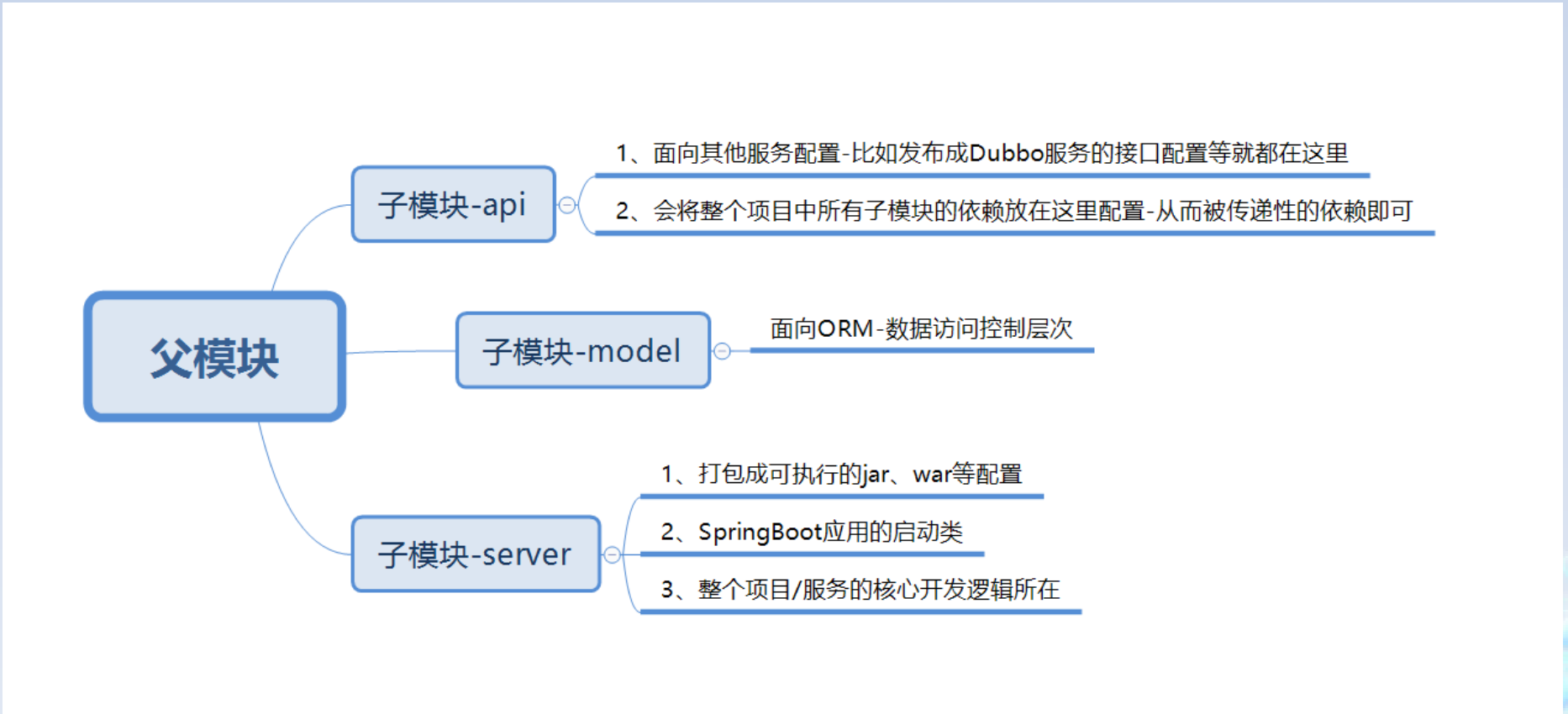
（2）下载系统源码数据库后，将项目导入IDEA中，将数据库导入Navicat Premium中

（3）采用外置的Tomcat将整个系统跑起来！

Java商城秒杀系统设计与实战

作者：debug

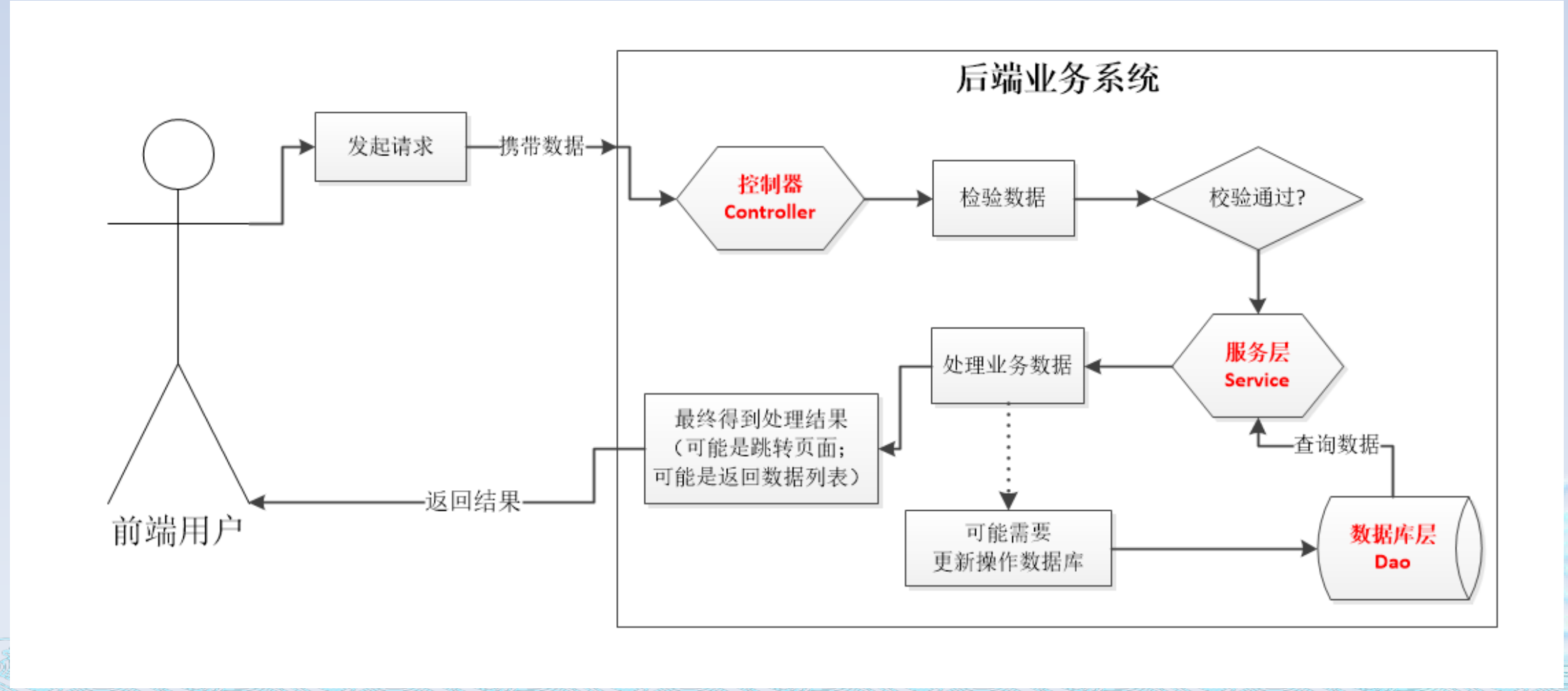
微服务项目的搭建-SpringBoot搭建多模块项目



Java商城秒杀系统设计与实战

作者：debug

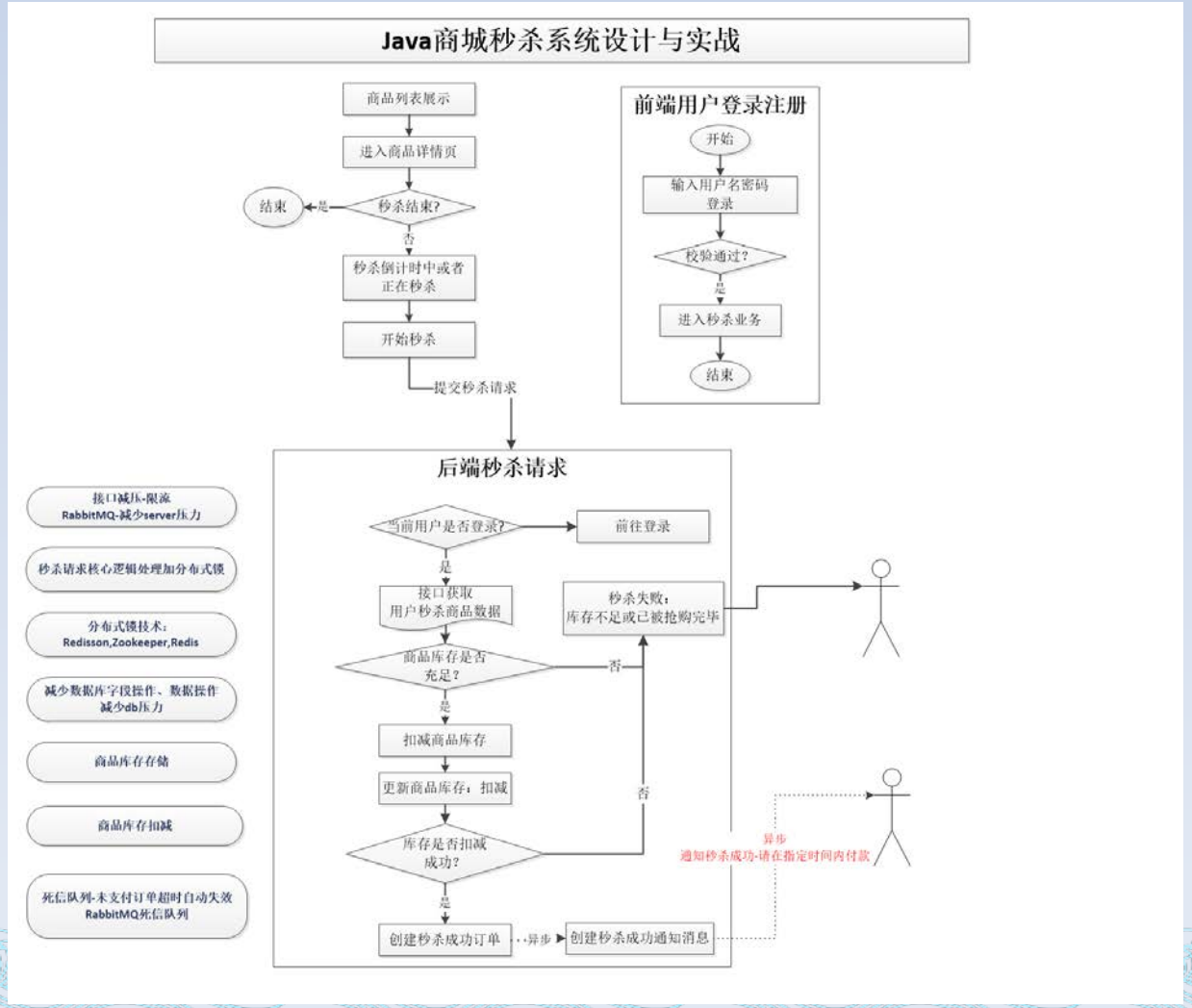
微服务项目的搭建-体验MVC的开发流程



Java商城秒杀系统设计与实战

作者：debug

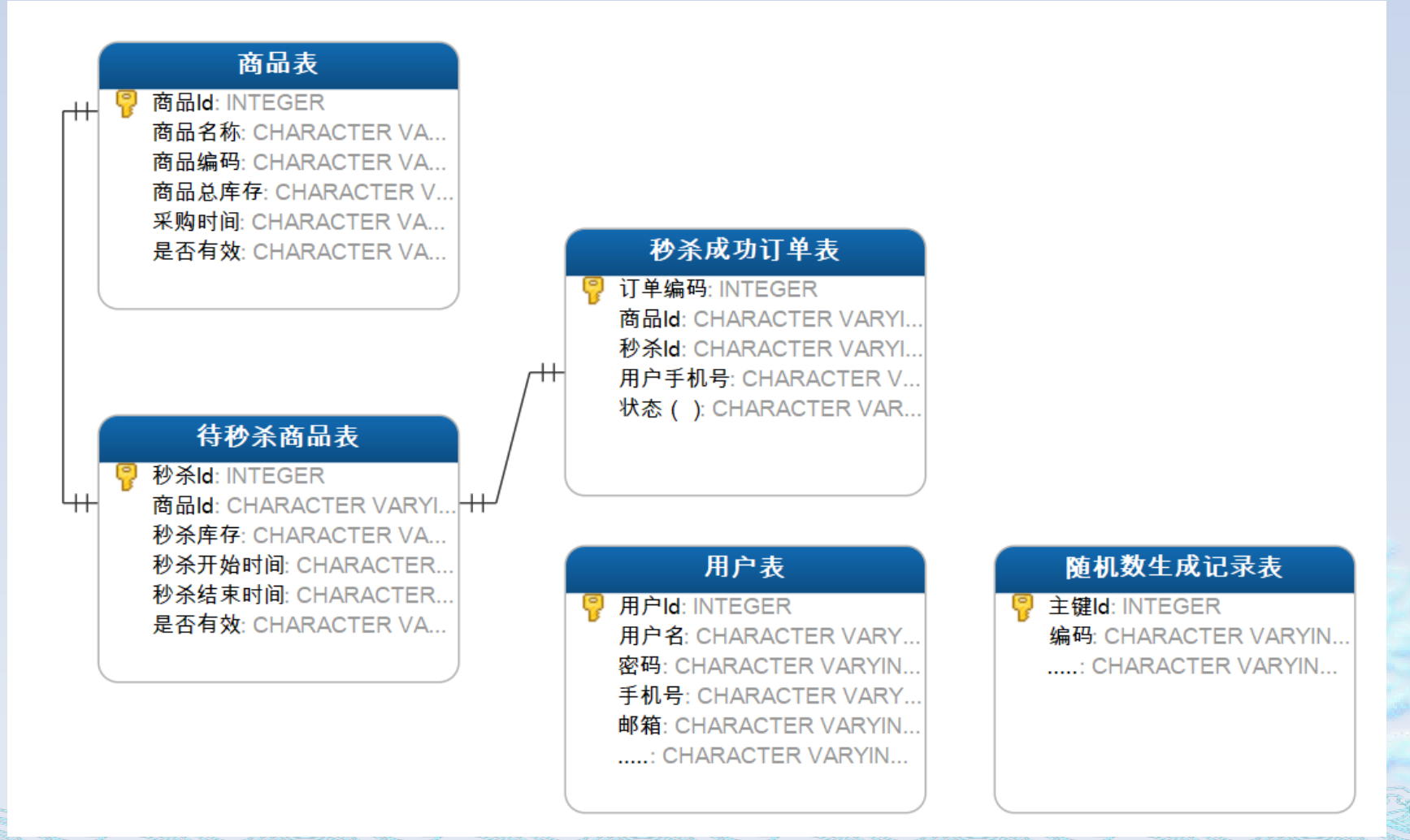
微服务项目的搭建-秒杀系统整体业务流程介绍



Java商城秒杀系统设计与实战

作者：debug

微服务项目的搭建-数据库设计与Mybatis逆向工程



Java商城秒杀系统设计与实战

作者：debug

秒杀业务代码实战-商品列表展示

开发指导：依旧是采用MVC的开发思想指导业务模块的开发：Controller -> Service -> Mapper(Dao)

需求：列出“剩余数量 > 0” 而且处于 “秒杀时间段内”的待秒杀商品列表

技巧：为了前端处理方便以及安全性，采用一字段canKill来决定商品在前端显示时是否可以秒杀！

Java商城秒杀系统设计与实战

作者：debug

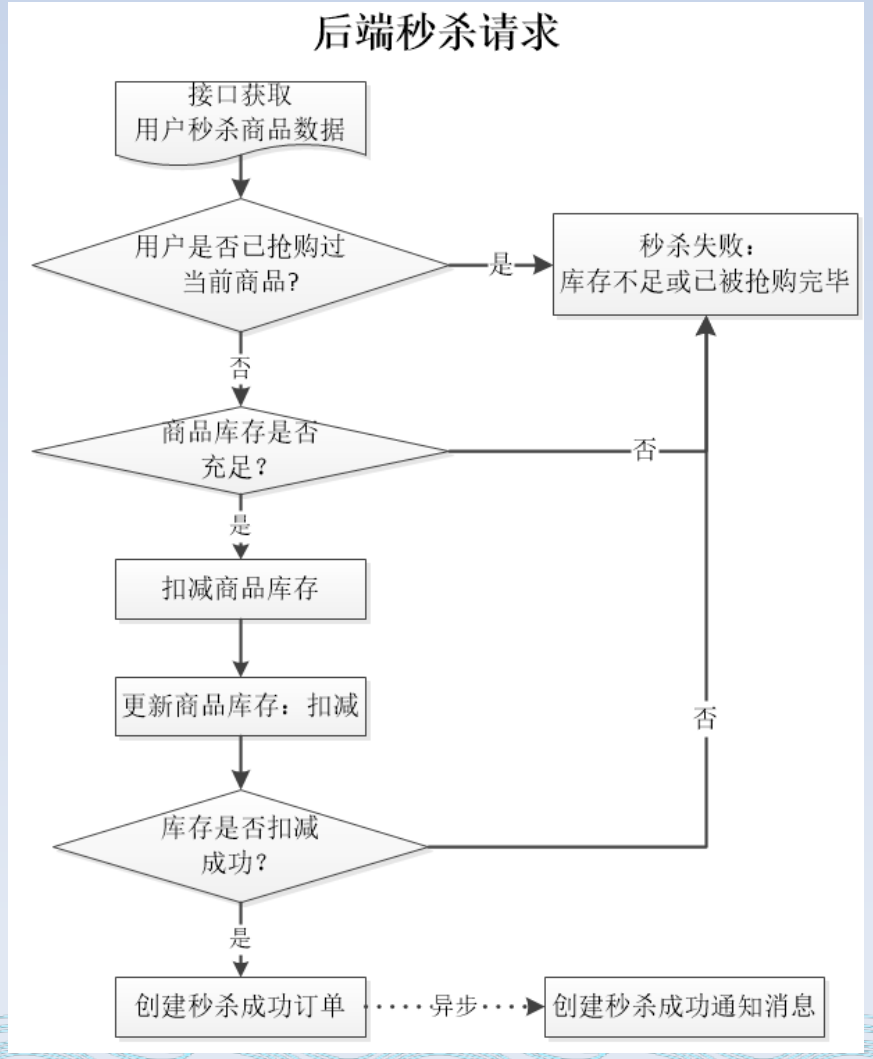
秒杀业务代码实战-商品详情展示

开发指导：依旧是采用MVC的开发思想指导业务模块的开发：Controller -> Service -> Mapper(Dao)

Java商城秒杀系统设计与实战

作者：debug

秒杀业务代码实战-商品秒杀实战



Java商城秒杀系统设计与实战

作者：debug

秒杀业务代码实战-订单编号的生成方式

（1）采用传统的时间戳 + N为流水号 （UUID也是一个道理） 构成

（2）雪花算法：高效率生成分布式唯一ID的最佳方式 https://github.com/souyunku/SnowFlake

比较一：传统的方式要么太长，没法排序；要么需要依赖中间件。

比较二：雪花算法-（整体上按照时间自增排序，并且整个分布式系统内不会产生ID碰撞，高效率）

Java商城秒杀系统设计与实战

作者：debug

秒杀业务代码实战-整合前端实现完整的秒杀逻辑

建议：站在用户角度上使用自己的系统；Take User As Fool ！！

Java商城秒杀系统设计与实战

作者：debug

秒杀业务代码实战-整合RabbitMQ实现消息异步发送

步骤一：加入RabbitMQ的依赖、配置RabbitMQ服务的相关信息

步骤二：自定义注入RabbitMQ的相关配置-RabbitmqConfig

步骤三：创建基本的消息模型，实现消息的发送和接收

建议：最好有一定的RabbitMQ应用基础（可以考虑购买学习Debug的RabbitMQ实战视频教程或者套餐）

Java商城秒杀系统设计与实战

作者：debug

秒杀业务代码实战-邮件服务发送通知信息实战

步骤一：加入邮件依赖、整合邮件配置信息

步骤二：封装统一的发送邮件的服务

步骤三：实现多种发送邮件的方式

Java商城秒杀系统设计与实战

作者：debug

秒杀业务代码实战-整体再次回顾秒杀的全过程

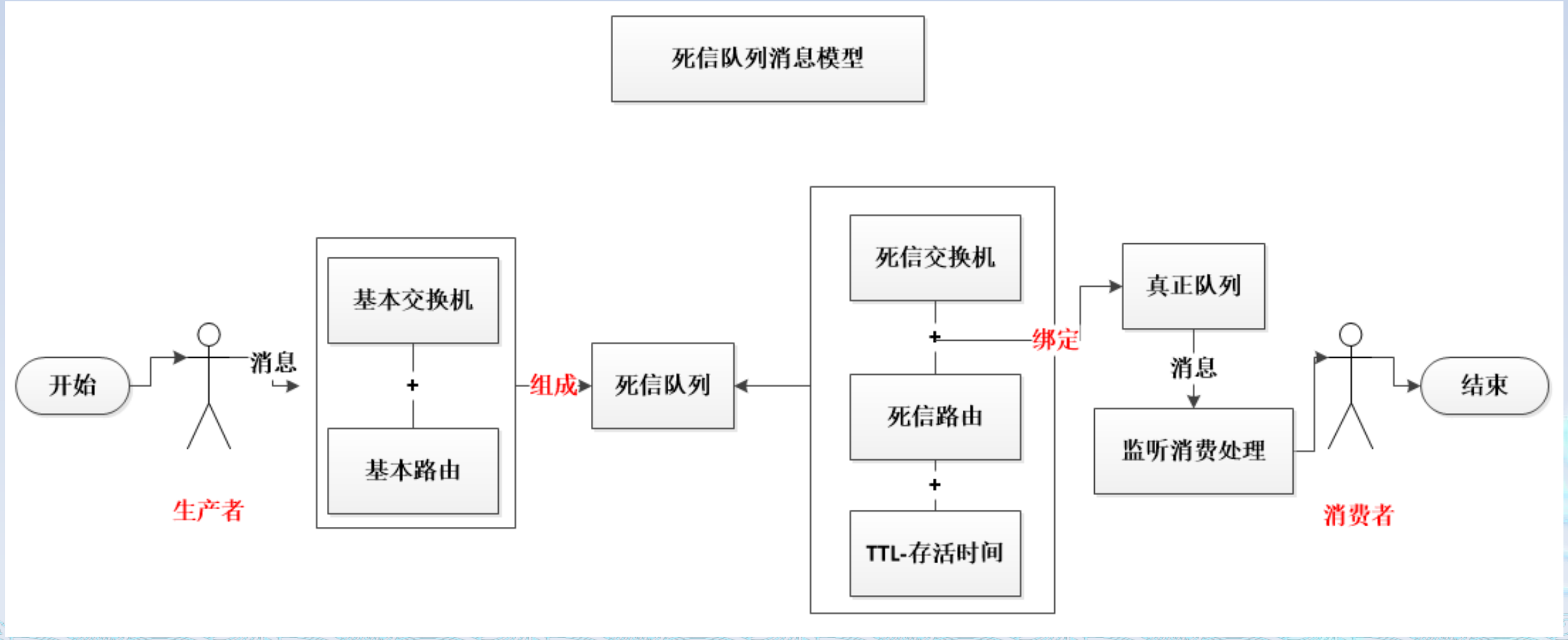
技能一：业务流程的分析与业务模块的划分

技能二：业务模块的服务化以及功能模块的解耦

Java商城秒杀系统设计与实战

作者：debug

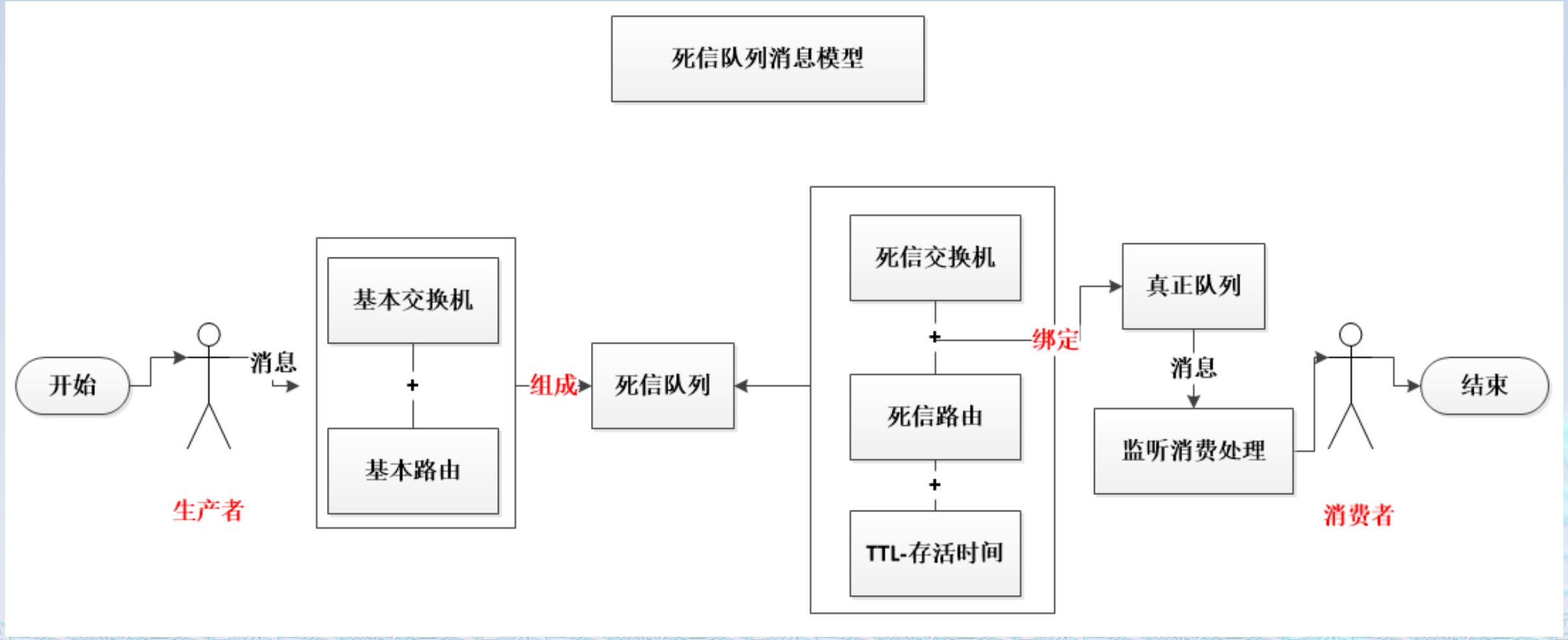
秒杀业务代码实战-死信队列失效超时未支付的订单一



Java商城秒杀系统设计与实战

作者：debug

秒杀业务代码实战-死信队列失效超时未支付的订单二



Java商城秒杀系统设计与实战

作者：debug

秒杀业务代码实战-定时任务失效超时未支付的订单

RabbitMQ死信队列缺陷：有许多订单在某个TTL集中失效，但是恰好RabbitMQ服务挂掉了。。。

补充解决方案：基于@Scheduled注解的定时任务实现-批量获取status=0的订单并判断时间超过了TTL

完整解决方案：结合线程池一起使用，规避单一线程处理多个任务调度的缺陷，支持多线程处理

Java商城秒杀系统设计与实战

作者：debug

秒杀业务代码实战-查看订单详情

步骤：根据订单编号orderNo直接获取订单详情！

Java商城秒杀系统设计与实战

作者：debug

秒杀业务代码实战-Jmeter高并发压力测试

安装Jmeter：官网下载解压即可！

Java商城秒杀系统设计与实战

作者：debug

秒杀业务代码实战-问题分析

致命点：产生的多个线程对“同一段操作共享数据的代码”进行并发操作，从而出现并发安全的问题！

核心方案：分布式锁解决共享资源在高并发访问时出现的“并发安全”问题

协助方案：对于瞬时流量、并发请求进行限流（目前是接口的限流，有条件时还能进行网关层面的限流）

辅助方案一：应用（秒杀系统）、中间件（RabbitMQ Redis…）服务做集群部署，提高高可用与稳定性！

辅助方案二：数据库Mysql做主备部署，如可以搭建一个Master写库，多个Slave读库实例的服务！！

Java商城秒杀系统设计与实战

作者：debug

秒杀逻辑优化-数据库Mysql层面优化抢单逻辑

核心SQL逻辑：“查询以及更减库存”时需要判断当前“可被更减的数量”是否仍然还大于 0

Java商城秒杀系统设计与实战

作者：debug

秒杀逻辑优化-基于Redis的分布式锁优化抢单逻辑

核心方法：SETNX + EXPIRE 联合使用

原因：Redis本身就是一个基于内存的、单线程的Key-Value存储数据库

Java商城秒杀系统设计与实战

作者：debug

秒杀逻辑优化-基于Redisson的分布式锁优化抢单逻辑

简介：基于Redis的驻内存网络数据库。。。In-Memory Data Grid

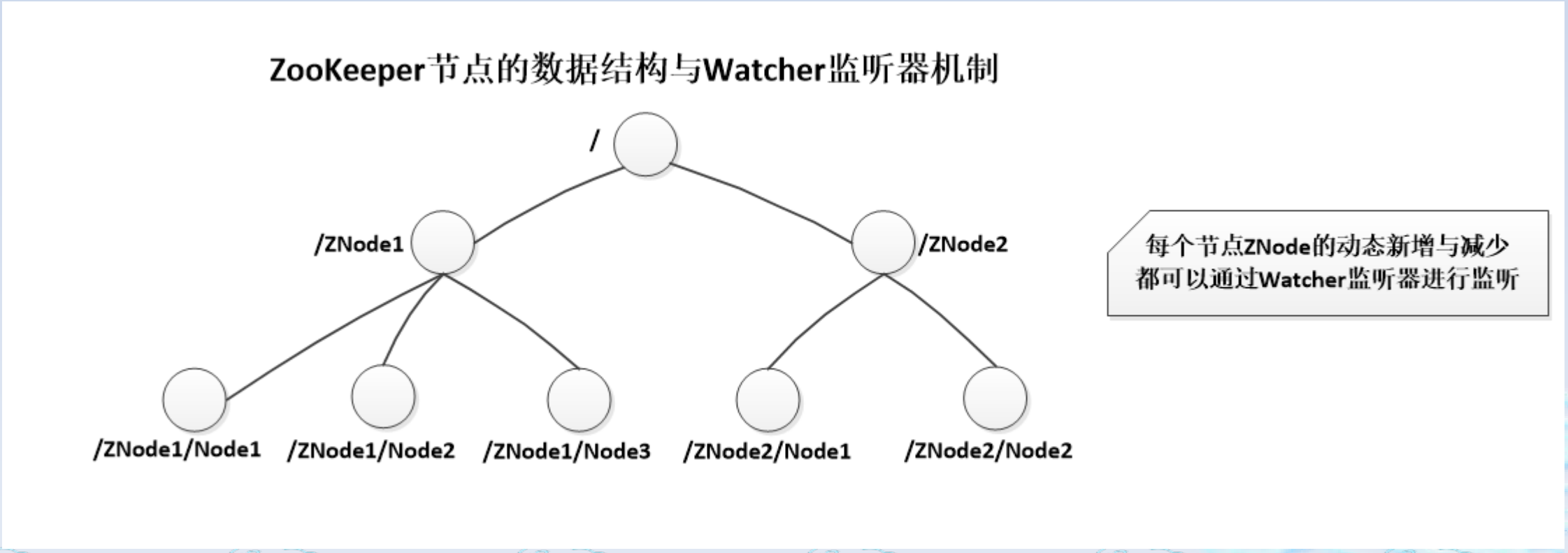
强大1：提供的功能不仅仅包含了Redis所提供的，还提供了诸如延迟队列、分布式服务、多种分布式对象等！

强大2：很亲民（很多组件是基于JavaSE核心知识体系中的数据结构来提供服务的；面向Redis实现）

Java商城秒杀系统设计与实战

作者：debug

秒杀逻辑优化-基于ZooKeeper的分布式锁优化抢单逻辑



Java商城秒杀系统设计与实战

作者：debug

秒杀逻辑优化-其他优化点介绍

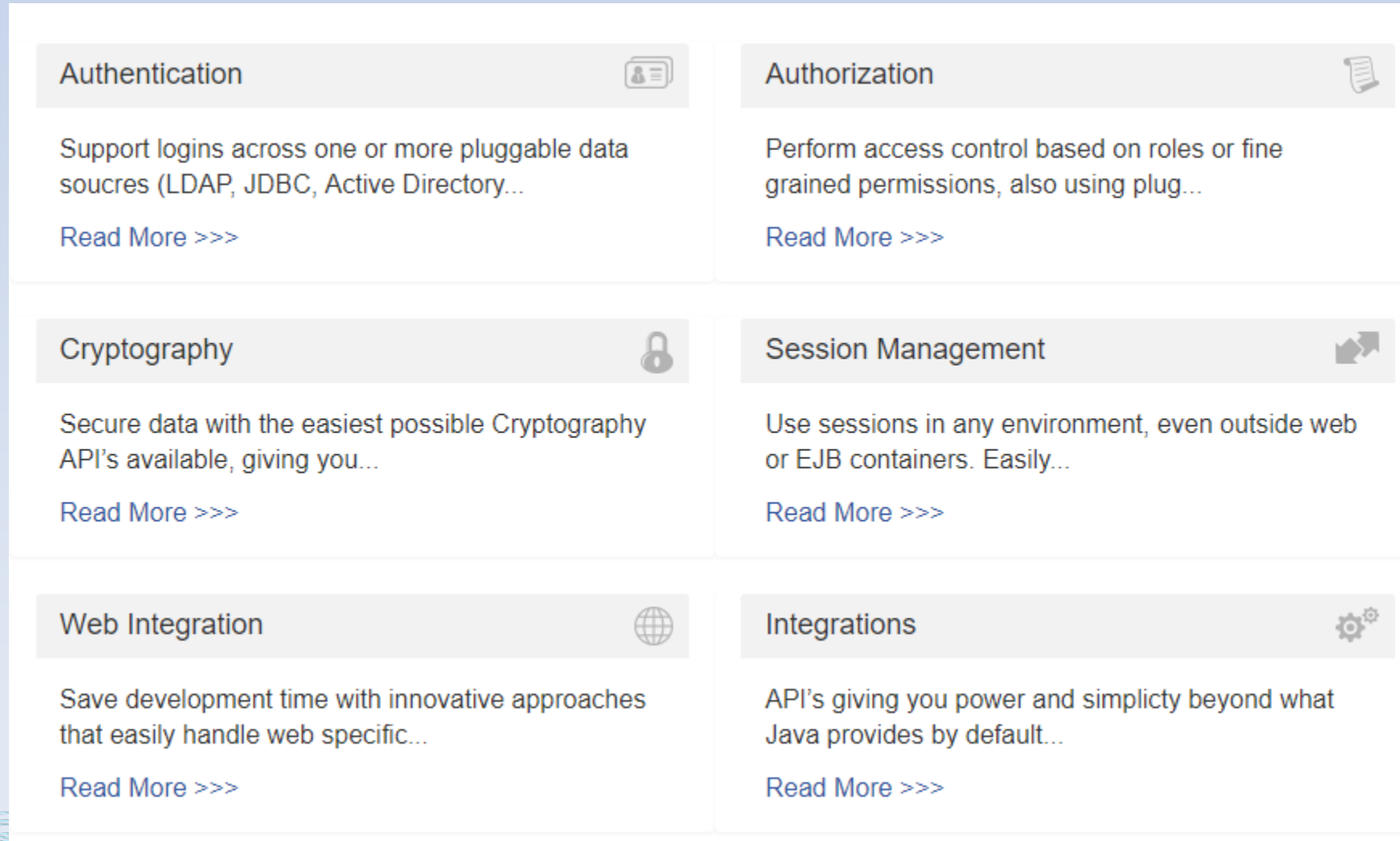
要点1：雪花算法 ~ 不采用数据库主键自增的方式，减轻DB压力；避免同一时刻生成相同的订单号

要点2：RabbitMQ业务模块解耦、异步通信 ~ 提高了接口的整体响应时间

Java商城秒杀系统设计与实战

作者：debug

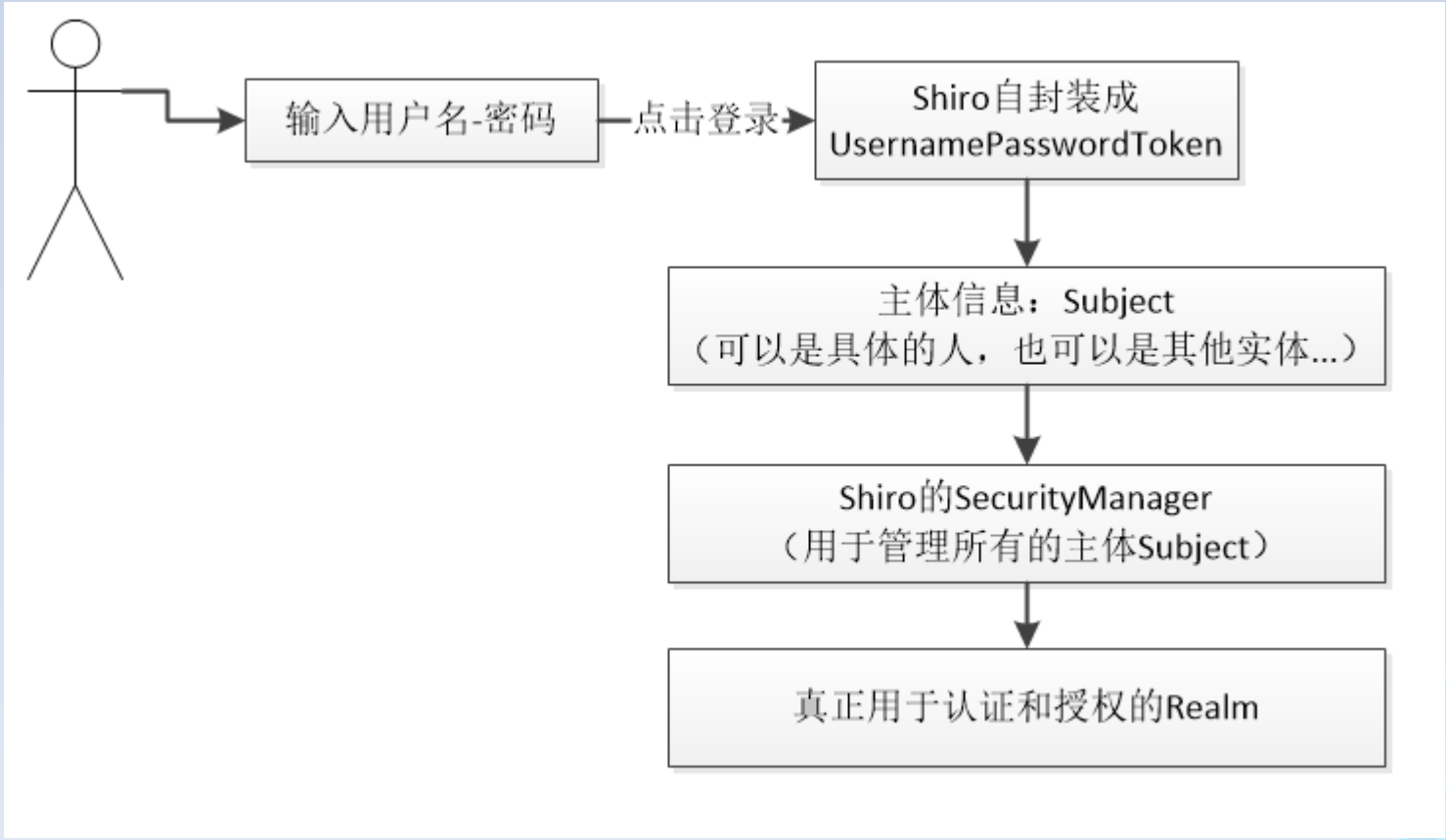
秒杀逻辑优化-整合Shiro实现用户登录



Java商城秒杀系统设计与实战

作者：debug

秒杀逻辑优化-整合Shiro实现用户登录



Java商城秒杀系统设计与实战

作者：debug

课程总结与建议

内容一：介绍了商城秒杀活动业务场景的整体业务流程，并由此业务流程进行了业务模块划分、数据库设计与系统搭建！

内容二：按照划分后的业务模块及其流程用代码实战实现！

内容三：重现了高并发业务场景下出现的一系列问题并由此介绍了相应的解决方案以及代码实战实现相应的解决方案！

内容四：整合了其他常见的第三方服务，如邮件服务、登录认证服务、订单编号全局唯一生成服务等等

Java商城秒杀系统设计与实战

作者：debug

课程总结与建议

建议一：纸上得来终觉浅，绝知此事要躬行，一定要多敲，边敲边理解，边敲边思考

建议二：整个课程完结之后自己要进行整体的回顾，包括整体的业务流程、出现的常见问题以及问题的解决方案！

建议三：对于“秒杀业务场景”中出现的问题的解决方案，还有很多，比如应用集群、应用服务器集群、中间件集群等等

建议四：别想太多，有想法那就去实践（条件允许的前提下），只有实践才能出真知，只有实践才能学得更多、更有成长