**Java高并发秒杀API之业务分析与DAO层**

# 第1章 课程介绍

## 1-1 Java高并发秒杀API之业务分析与DAO层课程介绍

01.简介

众多公司都是用这三个框架、

一、秒杀业务具有典型的“事务”特性；

二、秒杀\红包类需求越来越常见；

三、面试常见问题；

四门功课：

01.DAO层

02.Service

03.web

04.高并发优化

## 1-2 项目效果演示

# 第2章 梳理所有技术和搭建工程

## 2-1 相关技术介绍

### 01.MySQL

设计表；

sql技巧

事务和行级锁

### 02.mybatis

DAO层设计与开发；

mybatis合理使用；

mybatis与spring整合

### 03.spring

spring IOC整合 service；

声明式事务运用

### 04.SpringMVC<br>

Restful接口和使用；<br>

框架运作流程；<br>

Controller开发技巧

### 05.前段

交互设计；

BootStrap；

jQuery

### 06.高并发

高并发和高并发分析；

优化思路并实现

## 2-2 创建项目和依赖

mvn archetype:generate -DgroupId=org.seckill -DartifactId=seckill -DarchetypeArtifactId=maven-archetype-webapp -DinteractiveMode=false -DarchetypeCatalog=internal

# 第3章 秒杀业务分析

## 3-1 秒杀业务分析

一切非关系型的数据库都可以视作Nosql。

Nosql主要追求的是性能、分布式、高可用，但对事务支持的不是很好

事务机制依然是现在最可靠，最有效的数据落地方案。

## 3-2 MySQL实现秒杀难点分析

### 01.事务

开启 start transaction

修改 update库存数量

插入 insert购买明细

提交 commit

### 02.行级锁

一个用户的业务未提交，其余均是等待

### 03 难点：秒杀的难点是如何高效的处理竞争?

## 3-3 实现哪些秒杀功能

### 01.天猫的秒杀库存系统

### 02.秒杀功能(主要实现功能)

秒杀接口暴露

执行秒杀

相关查询

### 03.代码阶段

DAO设计编码

Service设计编码

Web设计编码

# 第4章 DAO层设计与开发

## 4-1 数据库设计与编码

Timestamp时间戳

KEY idx\_create\_time(create\_time) 建立索引

## 4-2 DAO实体和接口编码

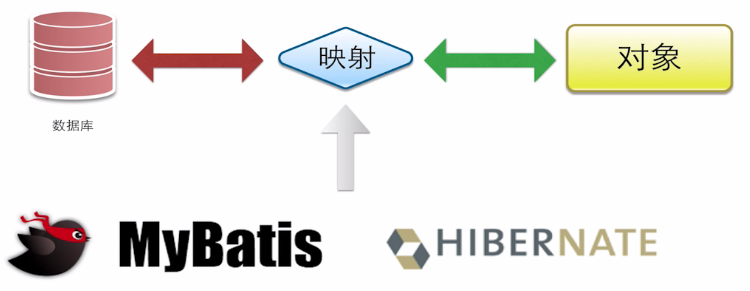
准备entity和dao接口中的方法

## 4-3 基于myBatis实现DAO理论

### 1.mybatis能做什么？

数据库---映射---对象

Mubatis 和 hibernate是工作在映射一层的，用来帮我们把数据库中的数据映射到对象，反过来再将entity中的数据映射到数据库中



### 2 Mybatis特点

参数+SQL 将最终的结果返回 = Entity/List

### 3 Mybatis怎么用呢？

**1）Sql语句写在哪？**

将SQL写到XML文件中

写到注解中(注解java 5.0后提供新的特性)

2）如何实现Dao接口呢？

**---通过mapper**自动实现Dao接口（mapper知道参数、返回类型、sql文件，可自动实现接口的实现类，去执行参数的封装，执行sql，把我们的返回结果集封装成我们想要的类型）

**---通过API编程方式**实现Dao接口（sql写在程序里面）

## 4-4 基于myBatis实现DAO编程(上)

mybtis的官方文档网址：<http://www.mybatis.org/mybatis-3/zh/getting-started.html>

dao接口是什么则对应的映射文件就是什么

驼峰命名法的转换：Table(create\_time) -> Entity(createTime)

## 4-5 基于myBatis实现DAO编程(下)

left join(左联接) 返回包括左表中的所有记录和右表中联结字段相等的记录

right join(右联接) 返回包括右表中的所有记录和左表中联结字段相等的记录

inner join(等值连接) 只返回两个表中联结字段相等的行

select

sk.seckill\_id,

sk.user\_phone,

sk.create\_time,

sk.state,

s.seckill\_id "seckill.seckill\_id",

s.name "seckill.name",

s.number "seckill.number",

s.start\_time "seckill.start\_time",

s.end\_time "seckill.end\_time",

s.create\_time "seckill.create\_time"

from success\_killed sk

inner join seckill\_id s on sk.seckill\_id = s.seckill\_id

where sk.seckill\_id = #{seckillId}

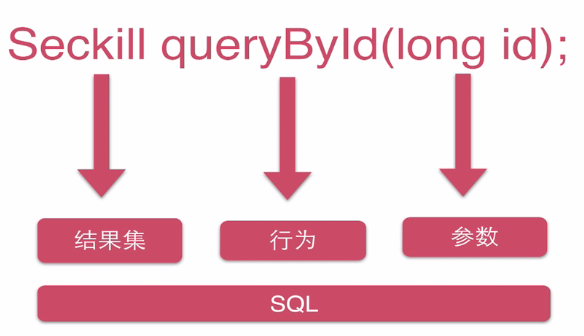
## 4-6 myBatis整合Spring理论

**整合目标**

1、更少的编码:

1）只写接口 不写实现类

2）接口能说明很多事情



2、更少的配置

1）别名：mybatis 帮我们实现了package scan(包扫描)的过程

2）配置扫描：自动扫描配置文件

3）dao实现：自动实现DAO接口，自动注入spring容器

3、足够的灵活性

1) 自由定制SQL

2) 自由传参

3）结果集自动赋值

## 4-7 mybatis整合Spring编码

spring\_mybatis整合配置文件：spring-dao.xml

配置整合mybatis过程

1 配置数据库相关参数

2 数据库连接池：配置连接池属性 c3p0连接池的私有属性 关闭理解后不自动commit 获取连接超时时间 当获取连接失败重试次数

3 配置sqlsessionfactory对象：<bean> 注入数据库连接池 配置mybatis全局配置文件 扫描entity包使用别名 扫描sql配置文件

4 配置扫描dao接口包，动态实现dao接口，注入到spring容器中：注入sqlsessionfactory 给出需要扫描dao接口包

## 4-8 DAO层单元测试编码和问题排查(上)

1、注意propertis文件中的参数配置

2、使用mabatis调用接口方法时，若参数超过两个则要在接口的方法中添加@Param(“\*\*\*”)

3、entity 属性表

**Dao 接口类** 和 mapper中的entityDao.xml文件对应

**entityDao.xml** 只写sql，不再需要配置resultMap

**spring-dao.xml** 配置整合mybatis，

配置数据库属性、

数据库连接池、

配置sqlSessionFactory(注入数据库连接池、配置mybatis-config.xml、扫描entity、扫描mapper中的sql配置文件)

扫描Dao接口包，注入到spring容器中（注入sqlSessionFactory、扫描Dao接口包）

## 4-9 DAO层单元测试编码和问题排查(下)

联合主键和ignore防止重复插入并且忽略主键冲突的错误

**总结：掌握Mybatis和Spring的整合**

Mybatis:

主要配置maybtis-config.xml和所需要的接口对应的sql配置的.xml

接口只调用，mybatis负责实现

Spring：bean配置和注入

主要配置spring-dao.xml