# 总结

## 电商概念

1. B2C：商家对客户，京东、当当、发展为B2C平台，天猫(B2C平台 淘宝商城)
2. B2B：商家对商家，阿里巴巴；
3. C2C：个人对个人，淘宝市场，淘宝，QQ商城；
4. O2O：线上线下，团购、电影票等；

## 分布式系统架构



面向服务架构：SOA。

## 开发模式



## maven垂直划分和水平划分

优缺点：

1. 垂直划分，功能模块明确，层次不清晰，代码重用性差；
2. 水平划分，层次清晰，代码重用性高，独立维护；

淘淘商城的后台管理系统采取水平划分；

聚合工程的打包类型是pom，运行tomcat也是要运行聚合工程（taotao-manage）

## nginx常用命令

start nginx.exe //启动、 linux：./nginx 非root不能占用80端口

nginx.exe –s stop //停止

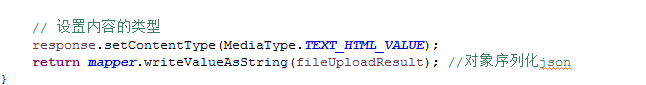
nginx.exe –s reload // 重新加载配置文件

修改配置文件：conf/nginx.conf

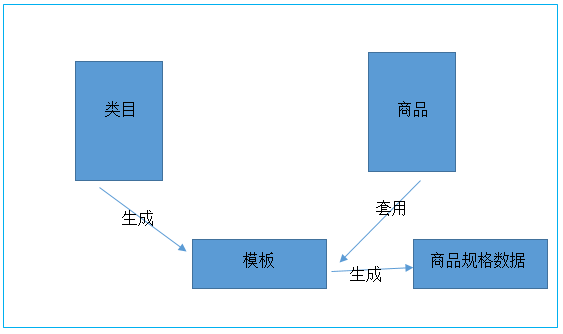
## 文件上传

需要注意的点是对上传的文件做校验。

## kindeditor上传后返回内容类型

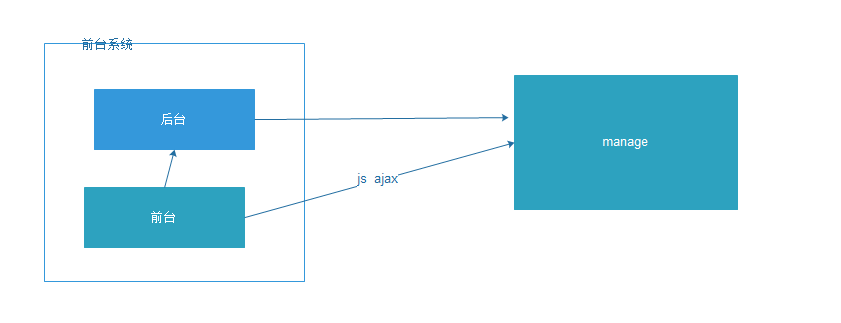


## 商品规格



重点是实现思路。

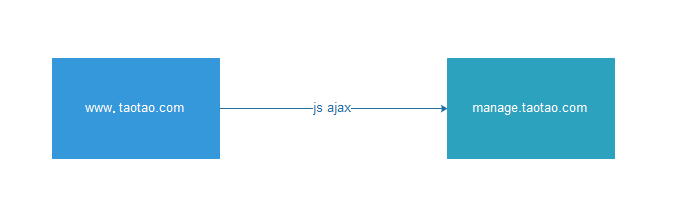
## 跨系统调用方式



2种调用方式：

1. ajax
   1. 效率
   2. 带宽
2. 后台转发请求
   1. 可以加逻辑
   2. 安全，接口不在公网公开

## 跨域



解决方案：

1. document.domain="taotao.com"
2. jsonp，部署子域名的情况

需要在SpringMVC中扩展MappingJackson2HttpMessageConverter，支持jsonp。

## CMS

重点是学习CMS系统的设计思想。

## Httpclient

1. 重点学习itcast-httpclient中的示例
2. 掌握spring和Httpclient的集成
   1. ApiService的单利还是多利的问题？

## json和java对象的互转

### java对象🡺json

objectMapper.writeValueAsString(object)

### json 🡺 java对象

#### 循环

JsonNode jsonNode = *objectMapper*.readTree(content);

ArrayNode arrayNode = (ArrayNode) jsonNode.get("rows");

List<Content> cons = **new** ArrayList<Content>();

**for** (JsonNode node : arrayNode) {

Content c = *objectMapper*.readValue(node.toString(), Content.**class**);

cons.add(c);

}

#### typefactory

String rowsData = *objectMapper*.readTree(content).get("rows").toString();

**return** *objectMapper*.readValue(rowsData, *objectMapper*.getTypeFactory().constructCollectionType(List.**class**, Content.**class**));

## Redis

1. 安装
2. 基本命令，ping、del、keys、exists、help
3. 掌握字符串数据类型，set、get、append、strlen、mset
4. 生存时间，expire、ttl
5. Jedis基本使用
6. Jedis连接池、集群jedis使用

## 单点登录系统

1. 掌握单点登录系统的设计思想以及在系统架构中的位置、作用；
2. 并发压力是在通过ticket查询用户信息；解决方案？

## 数据校验

使用Hibernate校验。

## 订单ID的生成

1. 订单ID不能重复
2. 订单ID尽可能的短（占用存储空间少，实际使用方便，客服相关）
3. 订单ID要求是全数字（客服）

订单ID的生成规则：

用户ID+当前系统的时间戳

String orderId = order.getUserId() + "" + System.currentTimeMillis();

问题1：各个服务器的时间不统一。

解决：在各个服务器上做时间的统一；（运维）

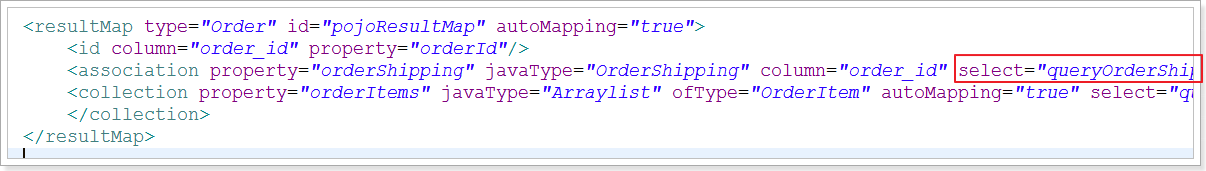
问题2：在问题1解决的基础上，可能存在毫秒级的偏差情况下？

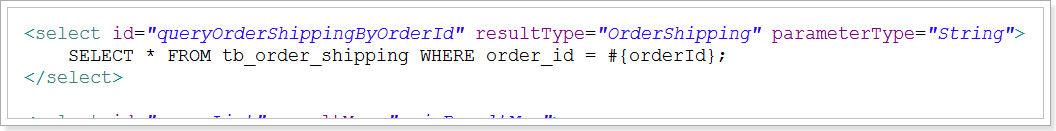
解决：修改订单生成规则：

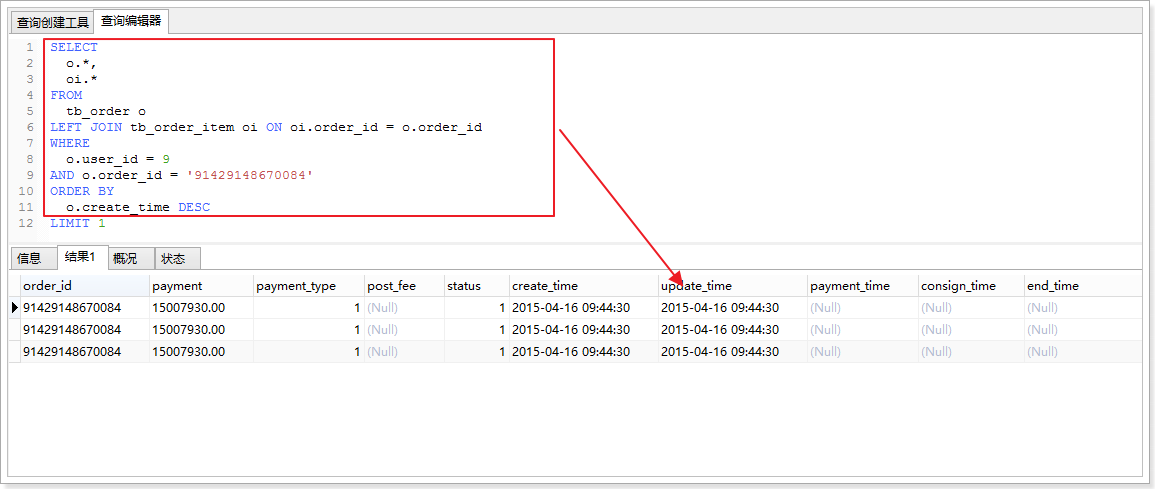
用户ID+当前系统的时间戳+随机数（3~4位） 问题：太长？ 把时间戳中的2014中的20拿掉；

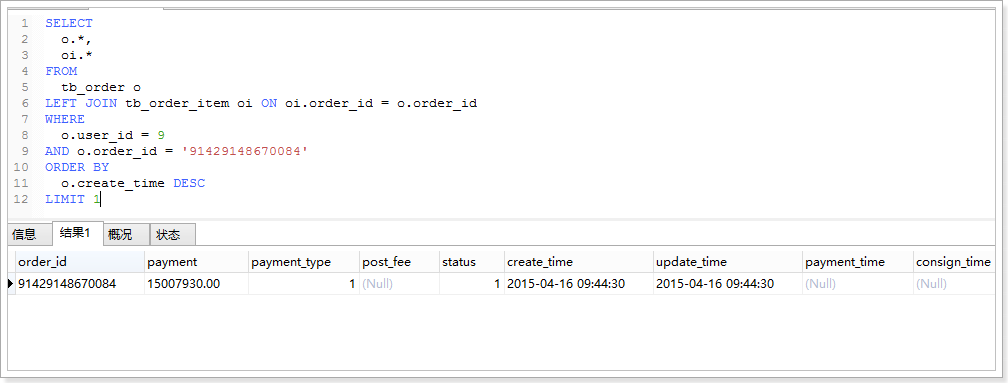
目的：使用订单ID不重复；

## 查询订单时需要使用延迟加载



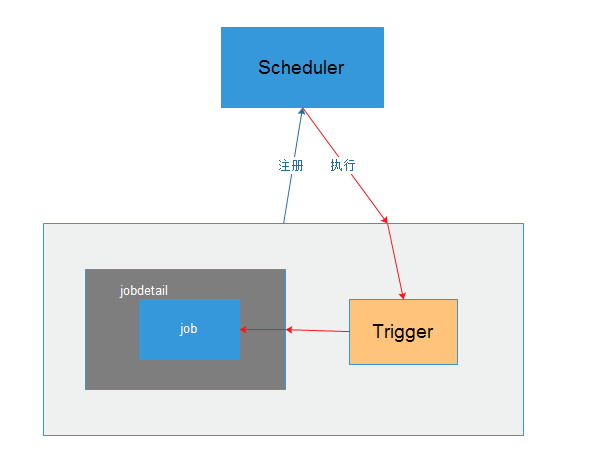






## quartz

1. 基本使用
2. 了解cron触发器的规则
3. quartz和spring的集成



## job和trigger关系

一个job可以有多个trigger；

一个trigger只能有个一个job；

# 就业指导

## 电商面试常见问题？

### 你们的架构是什么样的，能画一下吗？

### 你们的线上部署是什么样的，能画一下吗？

### 多台tomcat之间的session是怎么同步的？

### 如何解决大并发的问题？

### 你们生产环境的服务器有多少台？

### 数据库备份是怎么做的？有没有做读写分离？

## 简历指导

1. 项目名称：xxxx商城或 xxx电商
2. 项目周期：2012-12 ~ 2014-6 （这个就是大概，就说自己还参与其他模块的开发，比如说优惠、团购、秒杀等）
3. 项目介绍：xxx商城项目打造的是“社区+电商”的模式，用户不只是在社区中有自己的圈子，还可以将电商加入到社区中，整个电商网站实现的功能非常之多，采用分布式的架构设计，包括后台管理、前台系统、订单系统、单点登录系统、搜索系统、会员系统等。
4. 所用的技术：

* Spring、SpringMVC、Mybatis
* JSP、JSTL、jQuery、jQuery plugin、EasyUI、KindEditor（富文本编辑器）、CSS+DIV
* Redis（缓存服务器）
* Lucene、Solr（搜索）
* httpclient（调用系统服务）
* Mysql
* Nginx（web服务器）
* Quartz（定时任务）
* RabbitMQ（消息队列）

1. 开发环境和工具

* Eclipse 4.4.1
* Maven 3.2.3
* Tomcat 7.0.47（Maven Tomcat Plugin）
* JDK 1.7
* Mysql 5.6
* Nginx 1.5.1
* Redis 2.8.9
* Win7 操作系统
* SVN（版本管理）

1. 项目中的职责

在项目中主要负责相关系统的开发，主要有：

1. 后台管理系统，主要实现商品管理、商品规格参数管理、订单管理、会员管理等、CMS等，并且提供了跨域支持；
2. 前台系统，主要是面向用户访问，使用Httpclient和后台系统接口做交互，并且该系统在部署上采用集群的方式；
3. 单点登录系统，主要是提供集中用户登录凭证的集中解决方案，提供和用户信息相关的接口，比如说用户注册、查询等接口。
4. 订单系统，主要是提供和订单相关的业务接口，在订单系统了做了严格的数据校验以及高并发写的支持（这里可以说使用队列实现），

并且使用了Quartz定时任务实现对订单的定时扫描，比如说关闭超时未付款的订单；

1. 搜索系统，主要是提供商品的搜索，采用开源企业级系统Solr实现，采用了MQ机制保证了商品数据可以及时同步到solr中；
2. 会员系统，主要是维护用户的信息，已购买订单、优惠券、系统消息、修改密码、绑定手机等功能；
3. 缓存，主要是用Redis实现，并且对Redis做了集群来保证Redis服务的高可用。
4. 支付系统，主要是负责订单的支付、对账等功能，主要是对接了支付宝的接口；
5. 介绍项目的步骤
6. 我的项目是xxx系统，该项目是自己公司的产品，我们公司负责整个网站的运营，属于B2C平台；
7. 系统的用途，主要是提供B2C的平台，其中自营商品也有商家入住，类似天猫。
8. 系统架构，采用分布式的系统架构，其中前台系统和单点登录系统采用了集群的方式部署，在后台管理系统中采用了Maven的多模块化的管理，其中采用了水平切分的方式，将pojo、dao、service、web分层开发，这样做的好处就是可以重用性更高。

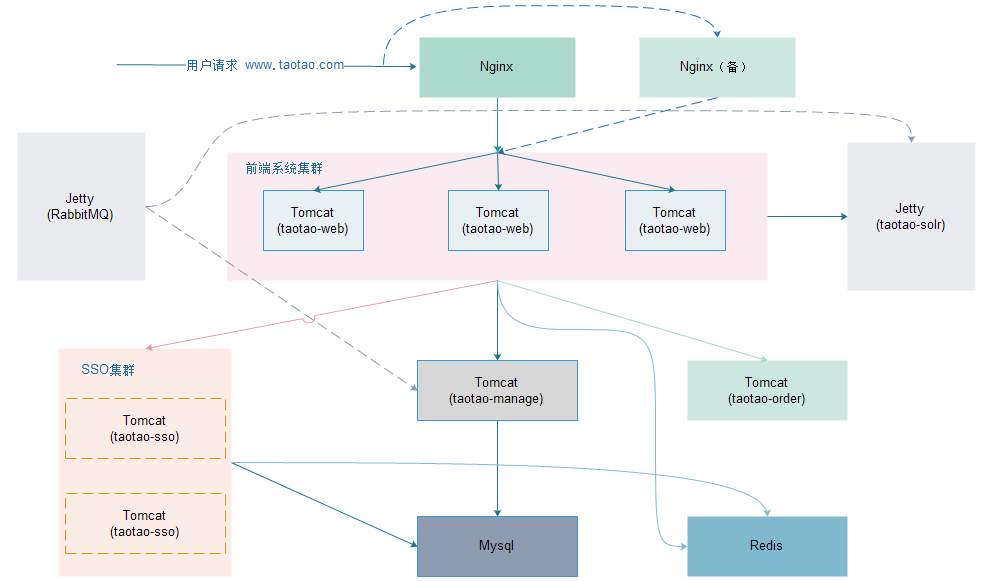
系统内部接口调用采用Httpclient，并且使用Httpclient的连接池技术，接口提供端采用RESTful方式的接口定义；

系统之间的通知机制采用MQ的方式，使用RabbitMQ的实现，使用了RabbitMQ的消息订阅模式的消息机制；

系统的接口还对JS的跨域做了支持，采用了jsonp的解决方法，在后台接口中扩展了spirng提供的jackson数据转化器实现；



1. 部署方面，采用了Nginx+tomcat的模式，其中nginx的作用一方面是做反向代理、负载均衡、另一方面是做图片等静态资源的服务器。



1. 项目中遇到的一些问题：
2. 跨域问题，因为我们是子域名访问子系统接口的，采用jsonp解决；
3. 图片上传问题，因为我们使用了KindEditor文本编辑器，它上传响应并不规范需要返回内容类型是文本不能是json格式；
4. 系统运行环境问题，有些问题是在开发环境下OK，但是到了测试环境就问题，比如说系统文件路径问题、导出报表中的中文问题（报表采用highcharts），需要在系统jdk中添加相应的中文字体才能解决；
5. 域名配置hosts问题，到测试环境或者是线上环境时需要在本机配置hosts否则接口访问不通；
6. 项目中用到了之前没有用到的技术，自己使用私人时间学习；
7. 支付时请求失败导致订单付款状态异常的问题，解决方法采用了定时任务解决、以及对账的方式解决；
8. 你的优点是什么？
   1. 不要吹技术很牛，从设计、架构、产品方面来说；
   2. 服从领导的安排；
   3. 可以自学一些技术；
   4. 能加班、能出差等；
   5. 谨慎**承诺**
   6. 有责任心、**沟通能力**强；
   7. 团队合作能力
9. 你的缺点是什么？
   1. 不能说真正的缺点；
   2. 优点说成缺点；