淘淘商城第一天

# 课程计划

共十二天的内容。

第一天：了解电商行业、了解淘淘商城。后台工程搭建。框架整合ssm。

第二天：商品管理。商品列表展示。商品添加中的商品类目选择。

第三天：商品添加：上传图片。Nginx、FastDFS。

第四天：商品添加实现：富文本编辑器的使用、商品的规格参数实现。

第五天：前台系统搭建。展示首页、展示商品类目。

第六天、第七天：首页大广告位的展示。Cms系统实现、redis缓存（集群）首页大广告位展示。

第八天：搜索功能的实现。使用solr实现（solr集群）。

第九天：商品详情页面实现，网页静态化freemaker实现。

第十天：单点登录系统实现，session共享。

第十一天：购物车、订单系统

第十二天：Quartz任务调度框架。项目部署、项目总结、面试中的问题。

第一天的课程内容：

1. 了解电商行业
2. 项目介绍
   1. 功能介绍
   2. 架构介绍
3. 后台工程搭建
   1. 使用maven搭建。
   2. 搭建聚合工程
4. 代码上传到svn
5. Ssm框架整合

# 了解电商行业

## 电商行业技术特点

* 技术新
* 技术范围广
* 分布式
* 高并发、集群、负载均衡、高可用
* 海量数据
* 业务复杂
* 系统安全

# 淘淘商城介绍

## 电商模式：

b2b：商家到商家。例如：阿里巴巴。1688.com。慧聪网

B2c：商家到用户。例如京东商城、天猫商城（b2b2c）

C2c：用户到用户。淘宝集市。

O2o：线上到线下。

## 商城简介

淘淘网上商城是一个综合性的B2C平台，类似京东商城、天猫商城。会员可以在商城浏览商品、下订单，以及参加各种活动。

管理员、运营可以在平台后台管理系统中管理商品、订单、会员等。

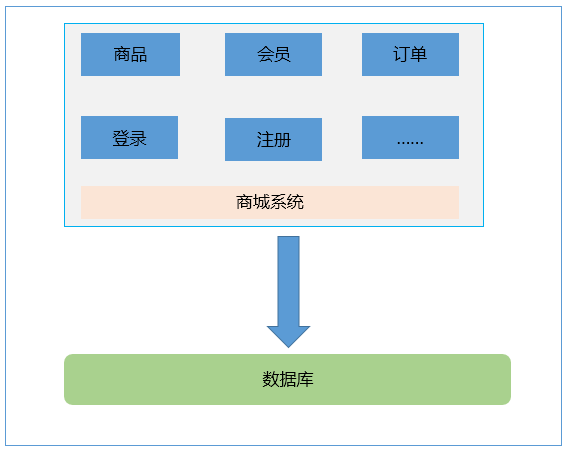
客服可以在后台管理系统中处理用户的询问以及投诉。

## 功能架构



## 技术架构

### 传统架构



1、Lamp:linux apache mysql php

2、小型机（ibm）+oracle+emc

3、廉价设备+分布式+java+mysql+缓存

## 分布式系统架构



分布式架构：

把系统按照模块拆分成多个子系统。

优点：

1. 把模块拆分，使用接口通信，降低模块之间的耦合度。
2. 把项目拆分成若干个子项目，不同的团队负责不同的子项目。
3. 增加功能时只需要再增加一个子项目，调用其他系统的接口就可以。
4. 可以灵活的进行分布式部署。

缺点：

系统之间交互需要使用远程通信，接口开发增加工作量。

## 技术选型

数据库：mysql

Dao层：mybatis、数据库连接池（德鲁伊druid）

缓存：redis

搜索：solr

Service层：spring

表现层：springmvc、jstl、EasyUI、jsp、freemaker

图片服务器：FastDFS（分布式文件系统）

反向代理服务器：nginx

定时器：Quartz

Web服务器：tomcat

工程管理：maven

## 开发工具和环境

Eclipse 4.5.0(Mars)，自带maven插件，需要手工安装svn插件。

Maven 3.3.3（开发工具自带）

Tomcat 7.0.53（Maven Tomcat Plugin）

JDK 1.7

Mysql 5.6

Nginx 1.8.0

Redis 3.0.0

Win7 操作系统

SVN（版本管理）

# 后台管理系统搭建

## 工程分析

网站一般分前台和后台。前台给用户看后台是管理人员使用。

使用maven管理工程。

1. 依赖管理
2. 项目构建

后台创建一个独立的web工程。创建一个maven工程打包方式是war包。

可以创建一个聚合工程，每个模块都是一个jar包，可以被其他系统依赖。

Taotao-parent（父工程管理jar包的版本）

|--Taotao-common（把通用的工具类打包）

|--Taotao-manager（继承父工程）

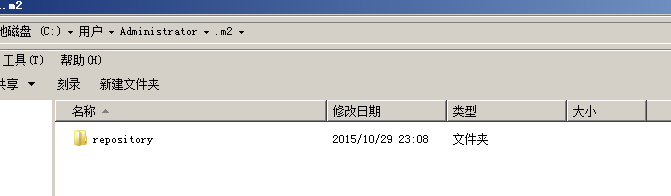
|-taotao-manager-pojo

|-taotao-manager-dao

|-taotao-manager-service

|-taotao-manager-web（war包）

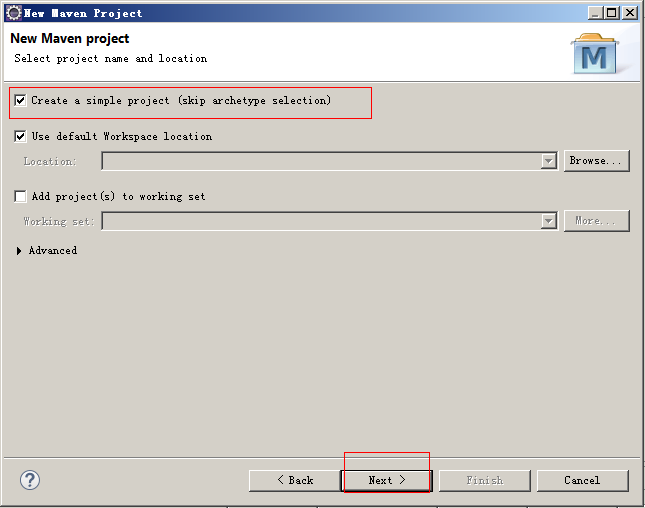
## Maven的配置

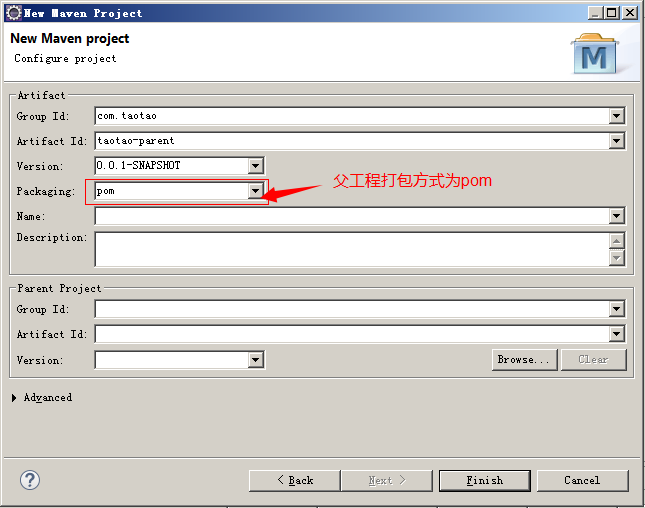


把参考资料中的.m2.7z解压缩覆盖本地仓库即可。

## 创建工程

### 创建taotao-parent工程



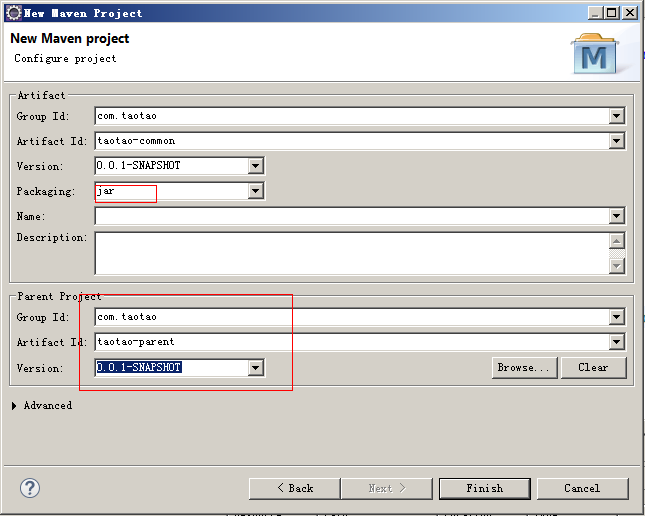


#### Pom文件

|  |
| --- |
| <project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>  <modelVersion>4.0.0</modelVersion>  <groupId>com.taotao</groupId>  <artifactId>taotao-parent</artifactId>  <version>0.0.1-SNAPSHOT</version>  <packaging>pom</packaging>  <!-- 集中定义依赖版本号 -->  <properties>  <junit.version>4.12</junit.version>  <spring.version>4.1.3.RELEASE</spring.version>  <mybatis.version>3.2.8</mybatis.version>  <mybatis.spring.version>1.2.2</mybatis.spring.version>  <mybatis.paginator.version>1.2.15</mybatis.paginator.version>  <mysql.version>5.1.32</mysql.version>  <slf4j.version>1.6.4</slf4j.version>  <jackson.version>2.4.2</jackson.version>  <druid.version>1.0.9</druid.version>  <httpclient.version>4.3.5</httpclient.version>  <jstl.version>1.2</jstl.version>  <servlet-api.version>2.5</servlet-api.version>  <jsp-api.version>2.0</jsp-api.version>  <joda-time.version>2.5</joda-time.version>  <commons-lang3.version>3.3.2</commons-lang3.version>  <commons-io.version>1.3.2</commons-io.version>  <commons-net.version>3.3</commons-net.version>  <pagehelper.version>3.4.2-fix</pagehelper.version>  <jsqlparser.version>0.9.1</jsqlparser.version>  <commons-fileupload.version>1.3.1</commons-fileupload.version>  <jedis.version>2.7.2</jedis.version>  <solrj.version>4.10.3</solrj.version>  <freemarker.version>2.3.23</freemarker.version>  <quartz.version>2.2.2</quartz.version>  </properties>  <dependencyManagement>  <dependencies>  <!-- 时间操作组件 -->  <dependency>  <groupId>joda-time</groupId>  <artifactId>joda-time</artifactId>  <version>${joda-time.version}</version>  </dependency>  <!-- Apache工具组件 -->  <dependency>  <groupId>org.apache.commons</groupId>  <artifactId>commons-lang3</artifactId>  <version>${commons-lang3.version}</version>  </dependency>  <dependency>  <groupId>org.apache.commons</groupId>  <artifactId>commons-io</artifactId>  <version>${commons-io.version}</version>  </dependency>  <dependency>  <groupId>commons-net</groupId>  <artifactId>commons-net</artifactId>  <version>${commons-net.version}</version>  </dependency>  <!-- Jackson Json处理工具包 -->  <dependency>  <groupId>com.fasterxml.jackson.core</groupId>  <artifactId>jackson-databind</artifactId>  <version>${jackson.version}</version>  </dependency>  <!-- httpclient -->  <dependency>  <groupId>org.apache.httpcomponents</groupId>  <artifactId>httpclient</artifactId>  <version>${httpclient.version}</version>  </dependency>  <!-- 单元测试 -->  <dependency>  <groupId>junit</groupId>  <artifactId>junit</artifactId>  <version>${junit.version}</version>  <scope>test</scope>  </dependency>  <!-- 日志处理 -->  <dependency>  <groupId>org.slf4j</groupId>  <artifactId>slf4j-log4j12</artifactId>  <version>${slf4j.version}</version>  </dependency>  <!-- Mybatis -->  <dependency>  <groupId>org.mybatis</groupId>  <artifactId>mybatis</artifactId>  <version>${mybatis.version}</version>  </dependency>  <dependency>  <groupId>org.mybatis</groupId>  <artifactId>mybatis-spring</artifactId>  <version>${mybatis.spring.version}</version>  </dependency>  <dependency>  <groupId>com.github.miemiedev</groupId>  <artifactId>mybatis-paginator</artifactId>  <version>${mybatis.paginator.version}</version>  </dependency>  <dependency>  <groupId>com.github.pagehelper</groupId>  <artifactId>pagehelper</artifactId>  <version>${pagehelper.version}</version>  </dependency>  <!-- MySql -->  <dependency>  <groupId>mysql</groupId>  <artifactId>mysql-connector-java</artifactId>  <version>${mysql.version}</version>  </dependency>  <!-- 连接池 -->  <dependency>  <groupId>com.alibaba</groupId>  <artifactId>druid</artifactId>  <version>${druid.version}</version>  </dependency>  <!-- Spring -->  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-context</artifactId>  <version>${spring.version}</version>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-beans</artifactId>  <version>${spring.version}</version>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-webmvc</artifactId>  <version>${spring.version}</version>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-jdbc</artifactId>  <version>${spring.version}</version>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-aspects</artifactId>  <version>${spring.version}</version>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-context-support</artifactId>  <version>${spring.version}</version>  </dependency>  <!-- JSP相关 -->  <dependency>  <groupId>jstl</groupId>  <artifactId>jstl</artifactId>  <version>${jstl.version}</version>  </dependency>  <dependency>  <groupId>javax.servlet</groupId>  <artifactId>servlet-api</artifactId>  <version>${servlet-api.version}</version>  <scope>provided</scope>  </dependency>  <dependency>  <groupId>javax.servlet</groupId>  <artifactId>jsp-api</artifactId>  <version>${jsp-api.version}</version>  <scope>provided</scope>  </dependency>  <!-- 文件上传组件 -->  <dependency>  <groupId>commons-fileupload</groupId>  <artifactId>commons-fileupload</artifactId>  <version>${commons-fileupload.version}</version>  </dependency>  <!-- Redis客户端 -->  <dependency>  <groupId>redis.clients</groupId>  <artifactId>jedis</artifactId>  <version>${jedis.version}</version>  </dependency>  <!-- solr客户端 -->  <dependency>  <groupId>org.apache.solr</groupId>  <artifactId>solr-solrj</artifactId>  <version>${solrj.version}</version>  </dependency>  <dependency>  <groupId>org.freemarker</groupId>  <artifactId>freemarker</artifactId>  <version>${freemarker.version}</version>  </dependency>  <dependency>  <groupId>org.quartz-scheduler</groupId>  <artifactId>quartz</artifactId>  <version>${quartz.version}</version>  </dependency>  </dependencies>  </dependencyManagement>  <build>  <finalName>${project.artifactId}</finalName>  <plugins>  <!-- 资源文件拷贝插件 -->  <plugin>  <groupId>org.apache.maven.plugins</groupId>  <artifactId>maven-resources-plugin</artifactId>  <version>2.7</version>  <configuration>  <encoding>UTF-8</encoding>  </configuration>  </plugin>  <!-- java编译插件 -->  <plugin>  <groupId>org.apache.maven.plugins</groupId>  <artifactId>maven-compiler-plugin</artifactId>  <version>3.2</version>  <configuration>  <source>1.7</source>  <target>1.7</target>  <encoding>UTF-8</encoding>  </configuration>  </plugin>  </plugins>  <pluginManagement>  <plugins>  <!-- 配置Tomcat插件 -->  <plugin>  <groupId>org.apache.tomcat.maven</groupId>  <artifactId>tomcat7-maven-plugin</artifactId>  <version>2.2</version>  </plugin>  </plugins>  </pluginManagement>  </build>  </project> |

### 创建taotao-common

保存是项目中用到的通用的工具类以及通用的pojo。打包方式jar包。

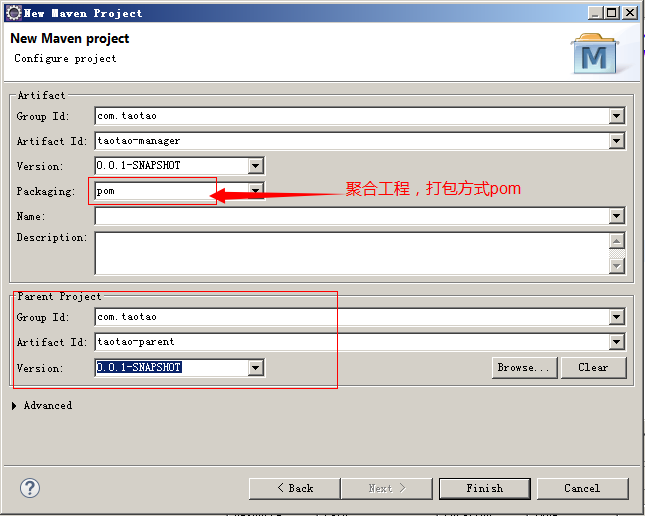


#### Pom文件

|  |
| --- |
| <project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>  <modelVersion>4.0.0</modelVersion>  <parent>  <groupId>com.taotao</groupId>  <artifactId>taotao-parent</artifactId>  <version>0.0.1-SNAPSHOT</version>  </parent>  <groupId>com.taotao</groupId>  <artifactId>taotao-common</artifactId>  <version>0.0.1-SNAPSHOT</version>  <!-- 添加依赖 -->  <dependencies>  <!-- 时间操作组件 -->  <dependency>  <groupId>joda-time</groupId>  <artifactId>joda-time</artifactId>  </dependency>  <!-- Apache工具组件 -->  <dependency>  <groupId>org.apache.commons</groupId>  <artifactId>commons-lang3</artifactId>  </dependency>  <dependency>  <groupId>org.apache.commons</groupId>  <artifactId>commons-io</artifactId>  </dependency>  <dependency>  <groupId>commons-net</groupId>  <artifactId>commons-net</artifactId>  </dependency>  <!-- Jackson Json处理工具包 -->  <dependency>  <groupId>com.fasterxml.jackson.core</groupId>  <artifactId>jackson-databind</artifactId>  </dependency>  <!-- httpclient -->  <dependency>  <groupId>org.apache.httpcomponents</groupId>  <artifactId>httpclient</artifactId>  </dependency>  <!-- 单元测试 -->  <dependency>  <groupId>junit</groupId>  <artifactId>junit</artifactId>  </dependency>  <!-- 日志处理 -->  <dependency>  <groupId>org.slf4j</groupId>  <artifactId>slf4j-log4j12</artifactId>  </dependency>  </dependencies>  </project> |

### 创建taotao-manager

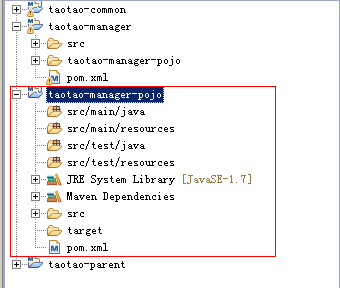
为聚合工程。打包方式pom



#### Pom文件

|  |
| --- |
| <project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>  <modelVersion>4.0.0</modelVersion>  <parent>  <groupId>com.taotao</groupId>  <artifactId>taotao-parent</artifactId>  <version>0.0.1-SNAPSHOT</version>  </parent>  <groupId>com.taotao</groupId>  <artifactId>taotao-manager</artifactId>  <version>0.0.1-SNAPSHOT</version>  <packaging>pom</packaging>  <!-- 添加依赖 -->  <dependencies>  <dependency>  <groupId>com.taotao</groupId>  <artifactId>taotao-common</artifactId>  <version>0.0.1-SNAPSHOT</version>  </dependency>  </dependencies>  </project> |

#### 添加模块taotao-manager-pojo



#### 添加模块taotao-manager-dao

Pom文件：

|  |
| --- |
| <project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>  <modelVersion>4.0.0</modelVersion>  <parent>  <groupId>com.taotao</groupId>  <artifactId>taotao-manager</artifactId>  <version>0.0.1-SNAPSHOT</version>  </parent>  <artifactId>taotao-manager-dao</artifactId>  <!-- 添加依赖 -->  <dependencies>  <dependency>  <groupId>com.taotao</groupId>  <artifactId>taotao-manager-pojo</artifactId>  <version>0.0.1-SNAPSHOT</version>  </dependency>  <!-- Mybatis -->  <dependency>  <groupId>org.mybatis</groupId>  <artifactId>mybatis</artifactId>  </dependency>  <dependency>  <groupId>org.mybatis</groupId>  <artifactId>mybatis-spring</artifactId>  </dependency>  <dependency>  <groupId>com.github.miemiedev</groupId>  <artifactId>mybatis-paginator</artifactId>  </dependency>  <dependency>  <groupId>com.github.pagehelper</groupId>  <artifactId>pagehelper</artifactId>  </dependency>  <!-- MySql -->  <dependency>  <groupId>mysql</groupId>  <artifactId>mysql-connector-java</artifactId>  </dependency>  <!-- 连接池 -->  <dependency>  <groupId>com.alibaba</groupId>  <artifactId>druid</artifactId>  </dependency>  </dependencies>  </project> |

#### 添加模块taotao-manager-Service

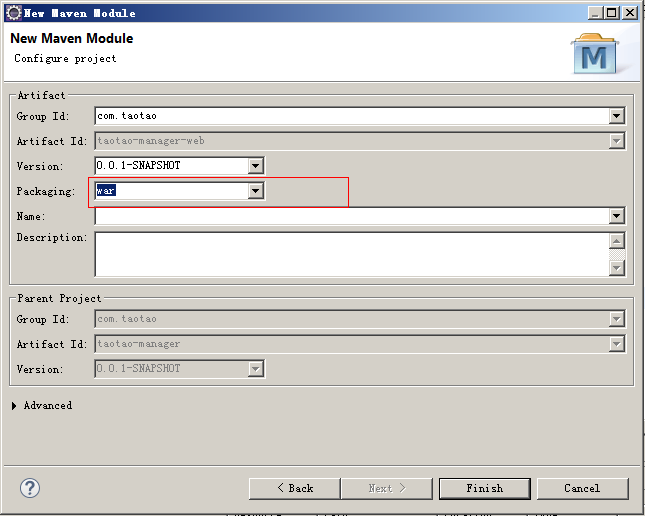
打包方式jar包。

Pom文件：

|  |
| --- |
| <project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>  <modelVersion>4.0.0</modelVersion>  <parent>  <groupId>com.taotao</groupId>  <artifactId>taotao-manager</artifactId>  <version>0.0.1-SNAPSHOT</version>  </parent>  <artifactId>taotao-manager-service</artifactId>  <dependencies>  <dependency>  <groupId>com.taotao</groupId>  <artifactId>taotao-manager-dao</artifactId>  <version>0.0.1-SNAPSHOT</version>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-context</artifactId>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-beans</artifactId>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-webmvc</artifactId>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-jdbc</artifactId>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-aspects</artifactId>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-context-support</artifactId>  </dependency>  </dependencies>  </project> |

#### 添加模块taotao-manager-web

打包方式war包。



##### Pom文件：

|  |
| --- |
| <project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>  <modelVersion>4.0.0</modelVersion>  <parent>  <groupId>com.taotao</groupId>  <artifactId>taotao-manager</artifactId>  <version>0.0.1-SNAPSHOT</version>  </parent>  <artifactId>taotao-manager-web</artifactId>  <packaging>war</packaging>  <!-- 添加依赖 -->  <dependencies>  <dependency>  <groupId>com.taotao</groupId>  <artifactId>taotao-manager-service</artifactId>  <version>0.0.1-SNAPSHOT</version>  </dependency>  <!-- JSP相关 -->  <dependency>  <groupId>jstl</groupId>  <artifactId>jstl</artifactId>  </dependency>  <dependency>  <groupId>javax.servlet</groupId>  <artifactId>servlet-api</artifactId>  <scope>provided</scope>  </dependency>  <dependency>  <groupId>javax.servlet</groupId>  <artifactId>jsp-api</artifactId>  <scope>provided</scope>  </dependency>  <!-- 文件上传组件 -->  <dependency>  <groupId>commons-fileupload</groupId>  <artifactId>commons-fileupload</artifactId>  </dependency>  </dependencies>  </project> |

##### Web.xml

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xmlns=*"http://java.sun.com/xml/ns/javaee"* xmlns:web=*"http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"*  xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"*  id=*"taotao"* version=*"2.5"*>  <display-name>taotao-manager</display-name>  <welcome-file-list>  <welcome-file>index.html</welcome-file>  <welcome-file>index.htm</welcome-file>  <welcome-file>index.jsp</welcome-file>  <welcome-file>default.html</welcome-file>  <welcome-file>default.htm</welcome-file>  <welcome-file>default.jsp</welcome-file>  </welcome-file-list>  </web-app> |

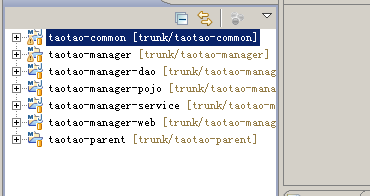
#### 在聚合工程中添加tomcat插件

|  |
| --- |
| <!-- 添加tomcat插件 -->  <build>  <plugins>  <plugin>  <groupId>org.apache.tomcat.maven</groupId>  <artifactId>tomcat7-maven-plugin</artifactId>  <configuration>  <port>8080</port>  <path>/</path>  </configuration>  </plugin>  </plugins>  </build> |

运行工程使用：tomcat7:run

需要把taotao-parent、taotao-common安装到本地仓库。

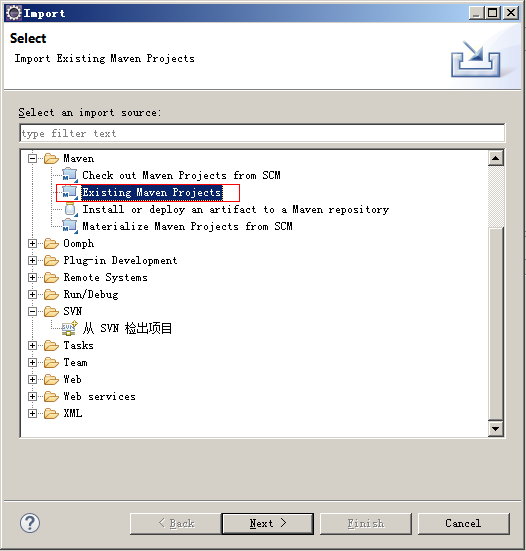
# Svn的使用

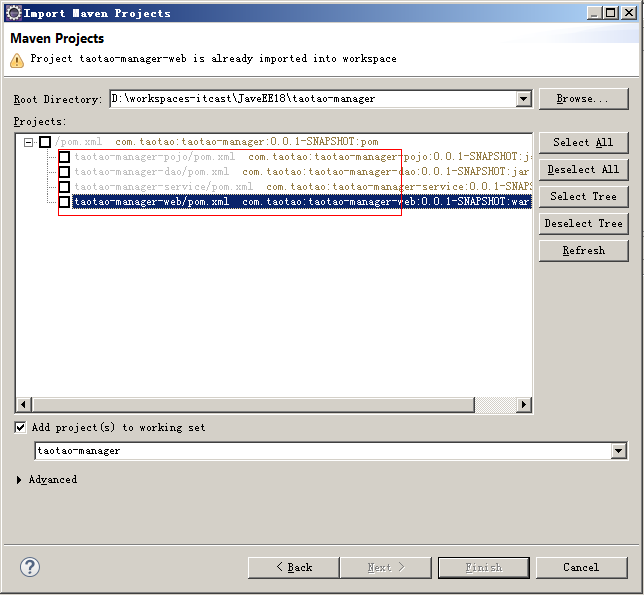


先把代码上传到svn，需要忽略到setting文件夹，和.classpath、.project文件。

下载代码后，需要转换成maven项目。

聚合工程使用导入存在maven工程导入到Eclipse中。





# 整合ssm框架

## 整合的思路

一、Dao层：整合mybatis和spring

需要的jar包：

1. mybatis的jar包
2. Mysql数据库驱动
3. 数据库连接池
4. Mybatis和spring的整合包。
5. Spring的jar包

配置文件：

1. mybatis的配置文件：SqlMapConfig.xml
2. Spring的配置文件：applicationContext-dao.xml
3. 数据源
4. 数据库连接池
5. 配置SqlSessionFactory（mybatis和spring整合包中的）
6. 配置mapper文件的扫描器。

二、Service层：

使用的jar包：spring的jar包。

配置文件：applicationContext-service.xml

配置一个包扫描器，扫描所有带@Service注解的类。

事务配置：

配置文件：applicationContext-trans.xml

1. 配置一个事务管理器
2. 配置tx
3. 配置切面
4. 表现层

使用springmvc，需要使用springmvc和spring的jar包。

配置文件：springmvc.xml

1. 配置注解驱动
2. 配置一个视图解析器。
3. 包扫描器，@Controller注解

Web.xml

1. 配置springmvc的前端控制器
2. Spring容器初始化的listener。

## 框架整合dao层

### SqlMapConfig.xml

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"* ?>  <!DOCTYPE configuration  PUBLIC "-//mybatis.org//DTD Config 3.0//EN"  "http://mybatis.org/dtd/mybatis-3-config.dtd">  <configuration>  </configuration> |

### applicationContext-dao.xml

db.properties

|  |
| --- |
| jdbc.driver=com.mysql.jdbc.Driver  jdbc.url=jdbc:mysql://localhost:3306/taotao?characterEncoding=utf-8  jdbc.username=root  jdbc.password=root |

applicationContext-dao.xml

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <beans xmlns=*"http://www.springframework.org/schema/beans"*  xmlns:context=*"http://www.springframework.org/schema/context"* xmlns:p=*"http://www.springframework.org/schema/p"*  xmlns:aop=*"http://www.springframework.org/schema/aop"* xmlns:tx=*"http://www.springframework.org/schema/tx"*  xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xsi:schemaLocation=*"http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.0.xsd*  *http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.0.xsd*  *http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-4.0.xsd http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-4.0.xsd*  *http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util-4.0.xsd"*>  <!-- 数据库连接池 -->  <!-- 加载配置文件 -->  <context:property-placeholder location=*"classpath:properties/\*.properties"* />  <!-- 数据库连接池 -->  <bean id=*"dataSource"* class=*"com.alibaba.druid.pool.DruidDataSource"*  destroy-method=*"close"*>  <property name=*"url"* value=*"${jdbc.url}"* />  <property name=*"username"* value=*"${jdbc.username}"* />  <property name=*"password"* value=*"${jdbc.password}"* />  <property name=*"driverClassName"* value=*"${jdbc.driver}"* />  <property name=*"maxActive"* value=*"10"* />  <property name=*"minIdle"* value=*"5"* />  </bean>  <!-- 让spring管理sqlsessionfactory 使用mybatis和spring整合包中的 -->  <bean id=*"sqlSessionFactory"* class=*"org.mybatis.spring.SqlSessionFactoryBean"*>  <!-- 数据库连接池 -->  <property name=*"dataSource"* ref=*"dataSource"* />  <!-- 加载mybatis的全局配置文件 -->  <property name=*"configLocation"* value=*"classpath:mybatis/SqlMapConfig.xml"* />  </bean>  <bean class=*"org.mybatis.spring.mapper.MapperScannerConfigurer"*>  <property name=*"basePackage"* value=*"com.taotao.mapper"* />  </bean>  </beans> |

## Service层

### applicationContext-service.xml

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <beans xmlns=*"http://www.springframework.org/schema/beans"*  xmlns:context=*"http://www.springframework.org/schema/context"* xmlns:p=*"http://www.springframework.org/schema/p"*  xmlns:aop=*"http://www.springframework.org/schema/aop"* xmlns:tx=*"http://www.springframework.org/schema/tx"*  xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xsi:schemaLocation=*"http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.0.xsd*  *http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.0.xsd*  *http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-4.0.xsd http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-4.0.xsd*  *http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util-4.0.xsd"*>  <!-- 包扫描器，扫描带@Service注解的类 -->  <context:component-scan base-package=*"com.taotao.service"*></context:component-scan>    </beans> |

### applicationContext-trans.xml

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <beans xmlns=*"http://www.springframework.org/schema/beans"*  xmlns:context=*"http://www.springframework.org/schema/context"* xmlns:p=*"http://www.springframework.org/schema/p"*  xmlns:aop=*"http://www.springframework.org/schema/aop"* xmlns:tx=*"http://www.springframework.org/schema/tx"*  xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xsi:schemaLocation=*"http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.0.xsd*  *http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.0.xsd*  *http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-4.0.xsd http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-4.0.xsd*  *http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util-4.0.xsd"*>  <!-- 事务管理器 -->  <bean id=*"transactionManager"*  class=*"org.springframework.jdbc.datasource.DataSourceTransactionManager"*>  <!-- 数据源 -->  <property name=*"dataSource"* ref=*"dataSource"* />  </bean>  <!-- 通知 -->  <tx:advice id=*"txAdvice"* transaction-manager=*"transactionManager"*>  <tx:attributes>  <!-- 传播行为 -->  <tx:method name=*"save\*"* propagation=*"REQUIRED"* />  <tx:method name=*"insert\*"* propagation=*"REQUIRED"* />  <tx:method name=*"add\*"* propagation=*"REQUIRED"* />  <tx:method name=*"create\*"* propagation=*"REQUIRED"* />  <tx:method name=*"delete\*"* propagation=*"REQUIRED"* />  <tx:method name=*"update\*"* propagation=*"REQUIRED"* />  <tx:method name=*"find\*"* propagation=*"SUPPORTS"* read-only=*"true"* />  <tx:method name=*"select\*"* propagation=*"SUPPORTS"* read-only=*"true"* />  <tx:method name=*"get\*"* propagation=*"SUPPORTS"* read-only=*"true"* />  </tx:attributes>  </tx:advice>  <!-- 切面 -->  <aop:config>  <aop:advisor advice-ref=*"txAdvice"*  pointcut=*"execution(\* com.taotao.service.\*.\*(..))"* />  </aop:config>  </beans> |

## 表现层

### 配置springmvc。

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <beans xmlns=*"http://www.springframework.org/schema/beans"*  xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns:p=*"http://www.springframework.org/schema/p"*  xmlns:context=*"http://www.springframework.org/schema/context"*  xmlns:mvc=*"http://www.springframework.org/schema/mvc"*  xsi:schemaLocation=*"http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd*  *http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-4.0.xsd*  *http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd"*>    <!-- 配置包扫描器 -->  <context:component-scan base-package=*"com.taotao.controller"*></context:component-scan>  <!-- 配置注解驱动 -->  <mvc:annotation-driven/>  <!-- 视图解析器 -->  <bean  class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>  <property name=*"prefix"* value=*"/WEB-INF/jsp/"* />  <property name=*"suffix"* value=*".jsp"* />  </bean>  </beans> |

### Web.xml

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xmlns=*"http://java.sun.com/xml/ns/javaee"* xmlns:web=*"http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"*  xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"*  id=*"taotao"* version=*"2.5"*>  <display-name>taotao-manager</display-name>  <welcome-file-list>  <welcome-file>index.html</welcome-file>  <welcome-file>index.htm</welcome-file>  <welcome-file>index.jsp</welcome-file>  <welcome-file>default.html</welcome-file>  <welcome-file>default.htm</welcome-file>  <welcome-file>default.jsp</welcome-file>  </welcome-file-list>  <!-- 初始化spring容器 -->  <context-param>  <param-name>contextConfigLocation</param-name>  <param-value>classpath:spring/applicationContext-\*.xml</param-value>  </context-param>  <listener>  <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>  </listener>  <!-- 解决post乱码 -->  <filter>  <filter-name>CharacterEncodingFilter</filter-name>  <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>  <init-param>  <param-name>encoding</param-name>  <param-value>utf-8</param-value>  </init-param>  </filter>  <filter-mapping>  <filter-name>CharacterEncodingFilter</filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping>  <!-- springmvc的前端控制器 -->  <servlet>  <servlet-name>taotao-manager</servlet-name>  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>  <!-- contextConfigLocation不是必须的， 如果不配置contextConfigLocation， springmvc的配置文件默认在：WEB-INF/servlet的name+"-servlet.xml" -->  <init-param>  <param-name>contextConfigLocation</param-name>  <param-value>classpath:spring/springmvc.xml</param-value>  </init-param>  <load-on-startup>1</load-on-startup>  </servlet>  <servlet-mapping>  <servlet-name>taotao-manager</servlet-name>  <url-pattern>/</url-pattern>  </servlet-mapping>  </web-app> |

## 测试整合框架

### 需求

根据商品id查询商品信息，返回json数据。

### Dao层

查询的表tb\_item，根据商品id查询。可以使用逆向工程生成的代码。

### Service层

接收商品id，调用mapper查询商品信息。返回商品的pojo。

参数：Long itemId

返回值：TbItem

|  |
| --- |
| /\*\*  \* 商品查询Service  \* <p>Title: ItemServiceImpl</p>  \* <p>Description: </p>  \* <p>Company: www.itcast.com</p>  \* **@author** 入云龙  \* **@date** 2015年11月11日下午4:28:55  \* **@version** 1.0  \*/  @Service  **public** **class** ItemServiceImpl **implements** ItemService {  @Autowired  **private** TbItemMapper itemMapper;  @Override  **public** TbItem getItemById(Long itemId) {  //TbItem item = itemMapper.selectByPrimaryKey(itemId);  TbItemExample example = **new** TbItemExample();  //创建查询条件  Criteria criteria = example.createCriteria();  criteria.andIdEqualTo(itemId);  List<TbItem> list = itemMapper.selectByExample(example);  //判断list中是否为空  TbItem item = **null**;  **if** (list != **null** && list.size() > 0) {  item = list.get(0);  }  **return** item;  }  } |

### Controller层

接收商品id，调用Service返回一个商品的pojo，直接响应pojo。需要返回json数据，使用@ResponseBody

**注意：使用@ResponseBody时候一定要把Jackson包添加到工程中。**

Url：/item/{itemId}

响应：TbItem

|  |
| --- |
| @Controller  **public** **class** ItemController {  @Autowired  **private** ItemService itemService;    @RequestMapping("/item/{itemId}")  @ResponseBody  **private** TbItem getItemById(@PathVariable Long itemId) {  TbItem item = itemService.getItemById(itemId);  **return** item;  }  } |

### 解决mapper映射文件不拷贝的问题

需要修改taotao-manager-dao工程的pom文件。添加如下内容：

|  |
| --- |
| <!-- 如果不添加此节点mybatis的mapper.xml文件都会被漏掉。 -->  <build>  <resources>  <resource>  <directory>src/main/java</directory>  <includes>  <include>\*\*/\*.properties</include>  <include>\*\*/\*.xml</include>  </includes>  <filtering>false</filtering>  </resource>  </resources>  </build> |