回顾：三个内容

Svn：checkout 下载代码 ，add,commit 提交

Update 更新

解决冲突:先update,再手动解决代码中的冲突，然后重新提交！

adminLTE：前端组件，拷贝，修改

ssm表和模块：产品表，订单表

## 今日重点：

1. SSM环境搭建
2. **产品查询，添加**
3. **订单查询与分页**

## 数据库与表结构介绍

后台管理系统

创建用户：

Create user xxx

Identified by a123

Default tablespace xxxx;

Grant dba to xxx;

## SSM环境搭建1

思路:ssm整合

1. 要把mybatis的对象交给spring容器创建和管理
2. Spring配置文件加载和容器的初始化在web.xml加载的时候完成

配置监听器

1. 创建工程

Dao 数据库交互 依赖 domain

Service 业务层工程 依赖dao,domain

Web 与前端交互 依赖service,domain

Domain 实体类工程

Utils 工具类的工程

1. 写代码或者拷贝配置文件
2. 测试（单元测试）

## SSM环境搭建2

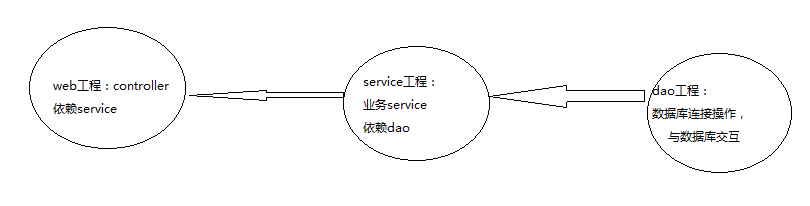
ApplicationContext.xml 将mybatis的配置整合

Spring-mvc.xml

Web.xml 监听器加载spring的配置

要求：独立完成！！！

## 产品操作-查询全部产品1



## 产品操作-查询全部产品2

Webapp：

Css,js,plugin,pages

## 产品操作-查询全部产品3

Jsp页面：拷贝老师的

转换：用户看到的和数据库存储之间的！！！

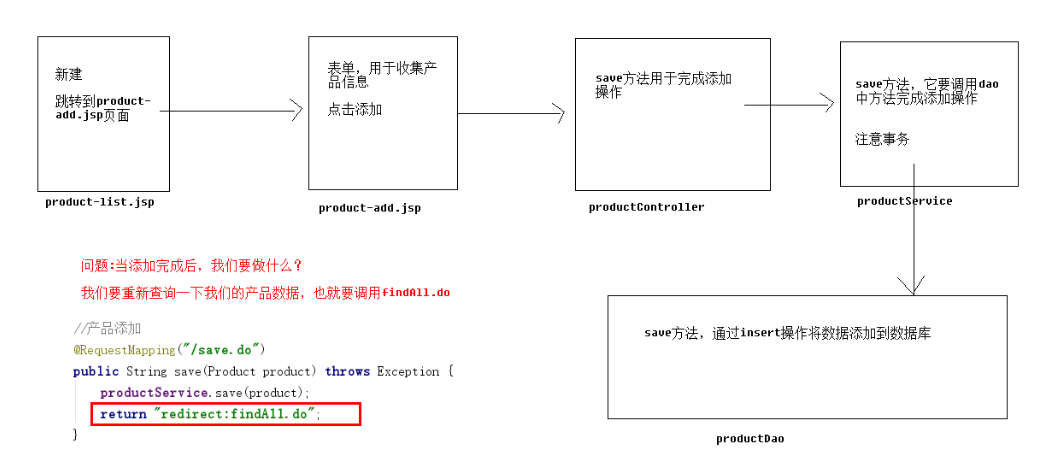
## product-list.jsp页面制作

请求同步，跳转！！！

异步！！！！

## main.jsp页面制作

## 产品操作-添加产品流程描述



思路！！！

1. 点击新建跳转到add.jsp
2. 表单，submit提交----------》action=”controller地址”
3. Controller类接收表单请求 save(Product product ) date
4. Service
5. Dao

## 产品操作-产品添加操作1

## 产品操作-产品添加操作2（类型转换）

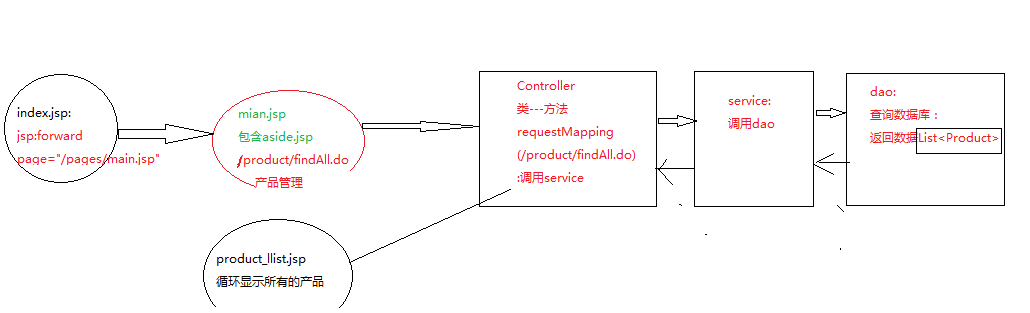
属性添加注解！

@DateTimeFormat(pattern="yyyy-MM-dd HH:mm")

LocalDate.now() yyyy-MM-dd

LocalDateTime.now()

产品：查询所有，添加



查询所有产品：

明天晚自习8点找

前端<url:今天晚自习8点> -----》controller(ModelAndView)----->service------>dao(查询数据库，返回List<Product>)

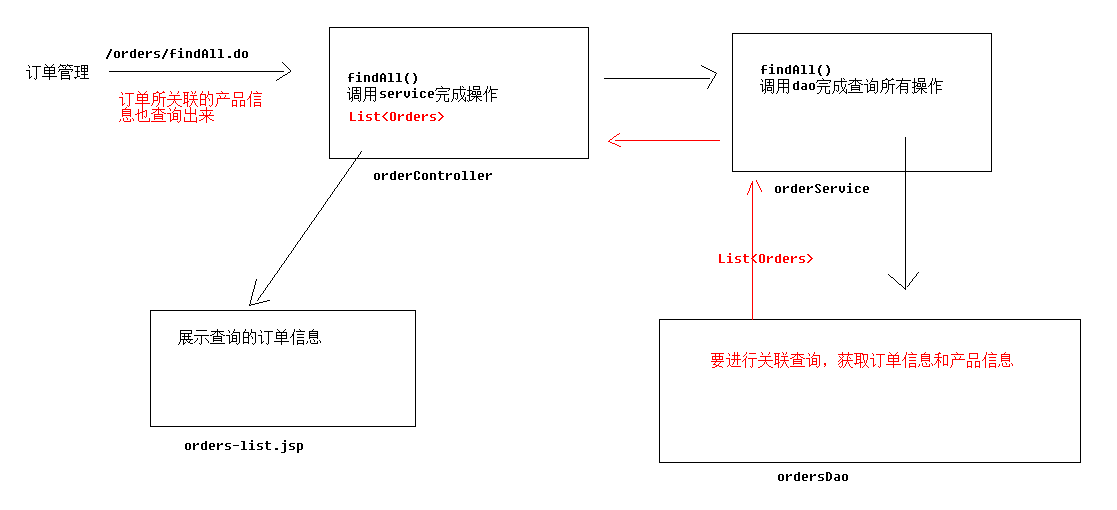
## 订单操作-表结构分析与表创建

订单表-------产品表

会员表

注意：orders表他包含了外键关系，先创建产品和会员表

## 订单操作-查询所有订单流程分析



## 订单操作-查询所有订单代码实现1

认为：订单和产品一对一的！！！

实体类：三个都写完，dao,service,controller

## 订单操作-查询所有订单代码实现2

一对一的关联查询：

@Select("select \* from orders")

@Results({

@Result(column = "productId",property = "product",one = @One(select =

"com.itheima.ssm.dao.IProductDao.findById"))

})

List<Orders> findAll() throws Exception;

查询订单信息同时查询订单中包含的产品信息！！！

## 订单操作-查询所有订单代码实现3

查询所有：分页！

## PageHelper概述与基本使用步骤介绍

实体类 PageBean(pagesize 每页记录数,total 总记录数,totalPage 总页数，curpage 当前页码，list<Product>每页数据)

Service组成！！！

1. 导入依赖
2. 配置插件
3. 使用：

Service 查询前：PageHelper.start(page,每页条数)

注意：一定要在dao访问方法前写参数设置代码！！！

## 订单操作-分页查询所有订单1(服务器端代码)

Controller中：

PageInfo pageInfo = new PageInfo(**查询集合**);

PageInfo 类似于pageBean

将pageInfo 存储在ModelAndView中跳转！

## 19.订单操作-分页查询所有订单2(页面代码处理)

PageInfo类：

Total 总记录数

Pages 总页数

Size 每页记录数

PageNum 当前页码

List 存储每页的数据

总结：

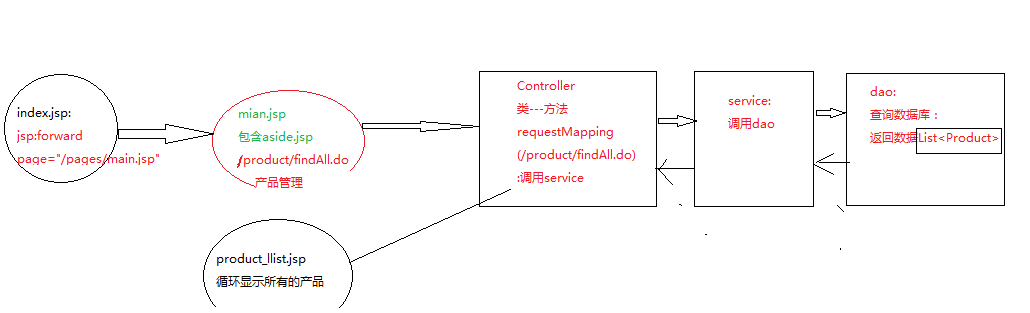
完成了三个功能：产品的查询所有，产品的添加，订单的查询所有并分页！

技能储备：

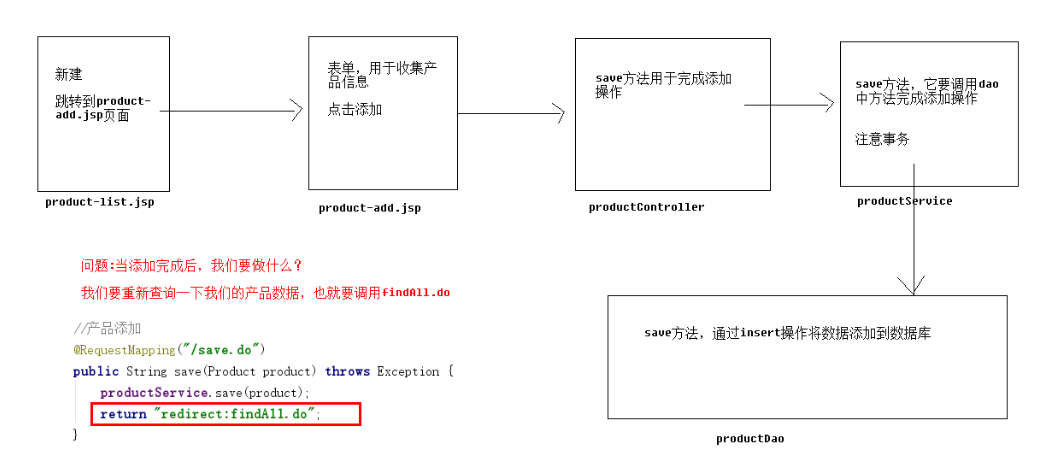
SSM整合，pageHelper分页，前端拷贝！

流程：

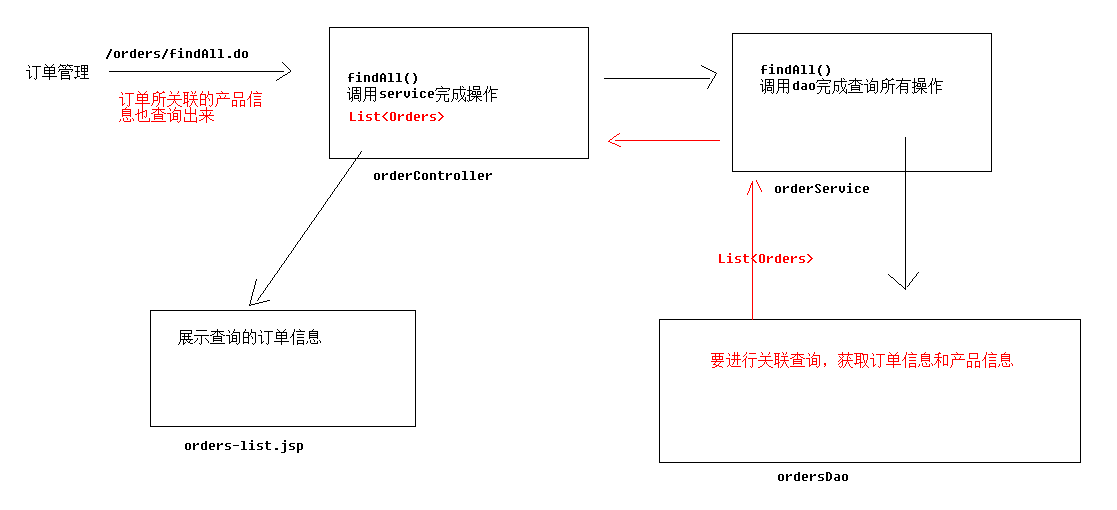
产品的查询所有：



添加产品：



订单的查询所有：



分页：

1. 添加pageHelper依赖
2. Dao的配置文件中添加pageHelper插件
3. Service查询前 PageHelper.start(page,size) ,再查询
4. Controller中：

PageInfo pageInfo = new PageInfo(查询集合list);

PageInfo 类似于pageBean

将pageInfo 存储在ModelAndView中跳转！

1. PageInfo源码看看属性！！