Javaweb-项目

# 项目介绍

后台管理 p2p\_management

实现登录，退出 产品管理（产品添加，产品查询，产品修改）

前台管理 p2p\_home

客户注册，登录，邮箱认证

帐户中心 产品查询 帐户信息查询 产品购买

技术架构：

采用javaweb开发中的经典三层架构：web层 service 层 dao层

技术点:

Jsp servlet javabean listener filter jquery ajax bootstrap

jdbc dbutils c3p0

mysql redis

tomcat nginx

新的知识点:

1. 数据库操作的事务管理 ThreadLocal
2. Javamail
3. Beanutils工具它可以将请求参数直接封装成实体对象
4. Md5加密

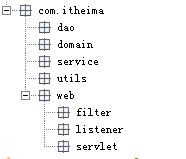
本次综合练习，主要是以ajax操作为主。

# 项目环境搭建

1. eclipse中项目环境搭建



搭建package结构



1. 关于数据库

五张表

* + - 1. user表，它是后台用户表
      2. product表，产品表
      3. account表 客户帐户表
      4. product\_account客户购买产品表
      5. customer 客户表

# 功能实现

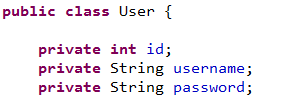
## p2p\_management功能实现

### 简单登录与退出

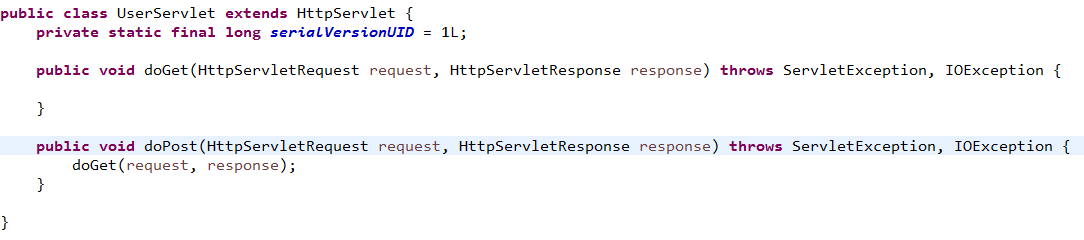
#### 登录操作

登录页面login.html

步骤1：创建一个User类它与表user对应



步骤2：创建UserServlet



步骤3：将login.html页面转换为login.jsp



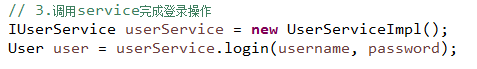
步骤4:在servlet中完成用户登录操作

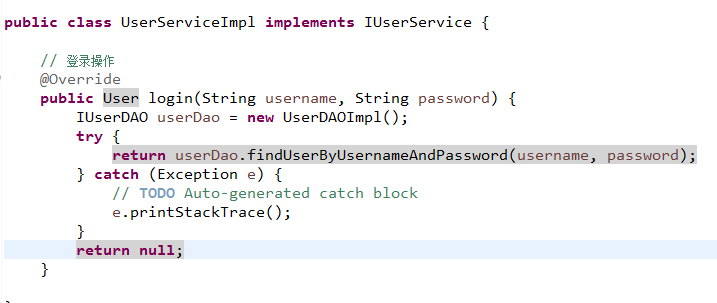
1.先获取username,password



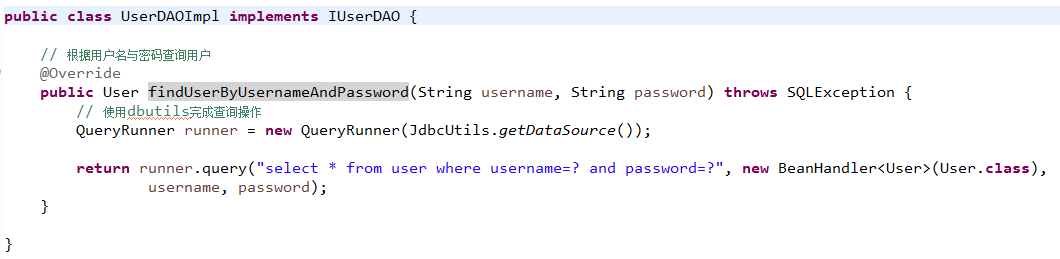
2.完成验证操作

3.调用service完成操作

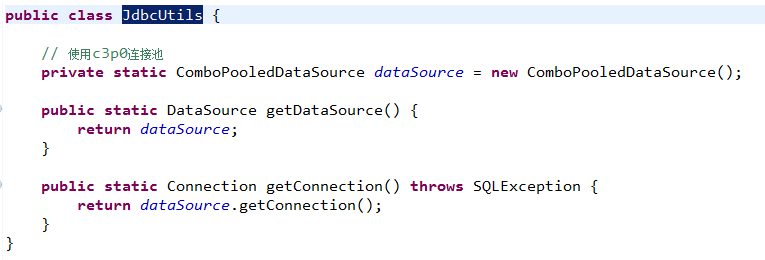




4.在service中调用dao完成操作



操作数据库使用的是c3p0连接池与dbutils工具，提供了一个JdbcUtils



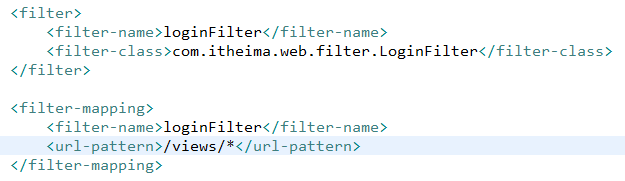
步骤5：从数据库中查询出user,判断是否登录成功，进行页面跳转



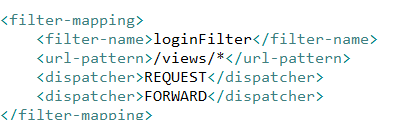
#### 用户未登录不能访问/views/hom.jsp

可以使用filter来完成操作





注意：上面的filter配置它只能对REQUEST请求拦截，我们也需要对FORWARD拦截



#### 退出操作

1.在home.jsp页面显示登录用户的名称



2.退出操作

我们退出时，一定要将session销毁，在将页面关闭。

我使用ajax操作来完成。

步骤1：创建home.js文件，在home.jsp页面导入



步骤2：修改home.jsp页面



步骤3:在home.js文件中创建一个函数shutdown()来完成关闭操作



### 产品管理-产品查询

关于产品信息的查询，我们使用ajax来完成操作，并且没有分页。

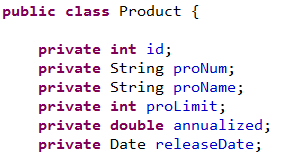
思路:

1.当home.jsp页面加载时就会向服务器发送请求来获取产品信息

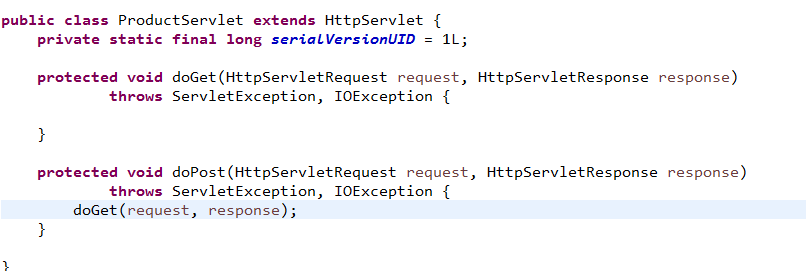
2.服务器端在servlet中处理请求，调用service,dao完成产品信息查询，并将信息转换成json响应到浏览器端

3.在浏览器中得到服务器响应的数据(json)，对数据进行处理，展示在home.jsp页面上。

#### 步骤1:创建Product模型



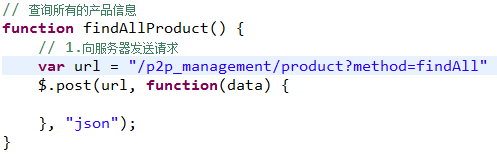
#### 步骤2创建服务器端的servlet ,service,dao



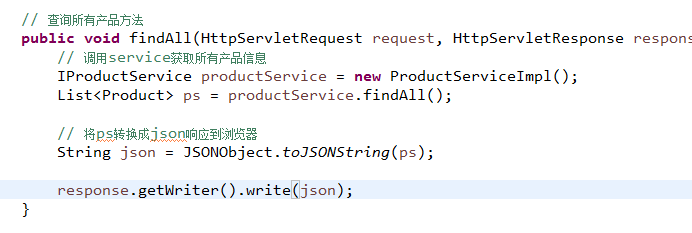
在service包中创建IProductService 与其实现ProductServiceImpl

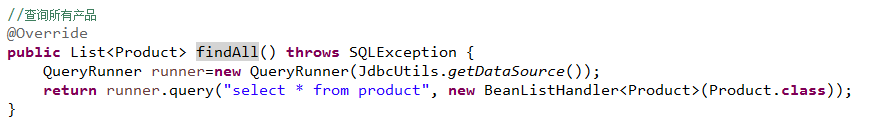
在dao包中创建IProductDAO 与其实现ProductDAOImpl

#### 步骤3 在home.js中完成页面加载时向服务器发送请求，获取产品信息



#### 步骤4.在服务器端查询所有产品信息，并转换成json响应到浏览器



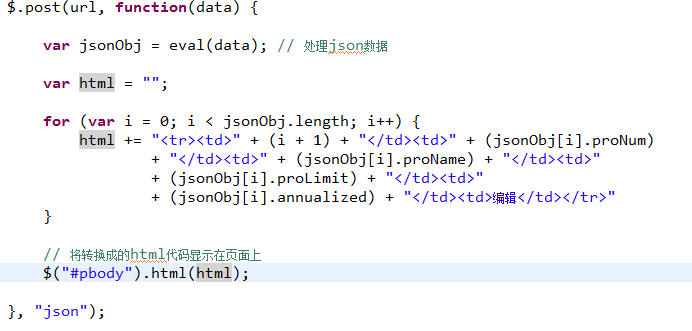


#### 步骤5:在浏览器端处理json数据，展示在页面上

在home.jsp页面上添加一个



将json数据处理后转换成的html代码显示在pbody中就可以。



### 产品管理-产品添加

产品添加思路:

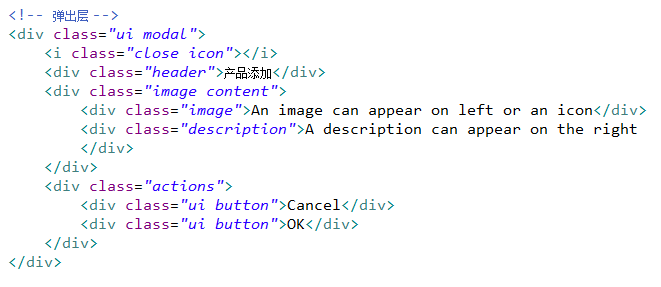
在home.jsp页面上点击添加按钮，弹出一个窗口，在窗口中录入相关产品信息，将信息发送到服务器端，在servlet中调用service,dao,完成将数据插入到product表中，最后我们要在调用查询所有产品操作，将添加的产品显示出来。

#### 步骤1：在home.jsp页面上点击添加按钮弹出窗口

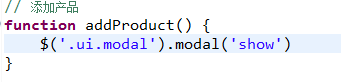
1.修改home.jsp页面上的按钮



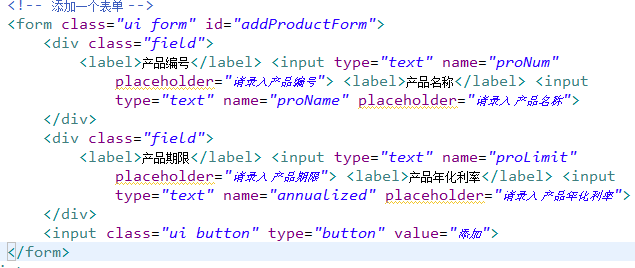
2.在home.jsp中添加弹出层



3.在home.js中添加js代码



4.在弹出的窗口中创建一个表单

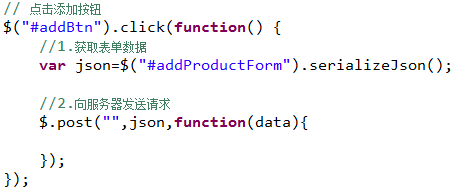


#### 步骤2：添加产品信息，将信息发送到服务器端完成添加操作

1.在home.js中完成点击添加按钮，获取添加的表单的数据

我们可以使用前面讲过的关于jquery插件。





#### 步骤3 在服务器端完成添加操作

使用beanutils来完成请求参数封装

1.导入jar包



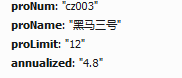
2.使用beanutils将请求参数封装到bean对象中



参数1 是bean对象

参数2 是我们的请求参数的Map集合，一般我们使用时都是使用request.getParameterMap()方法获取。

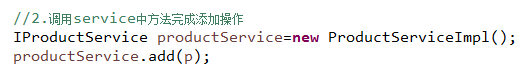
注意:要想将请求参数封装到bean对象中，必须让bean属性与请求参数的名称对应。



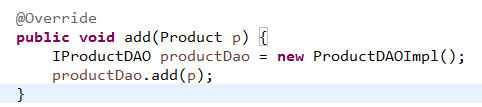
这些参数与Product类中的bean属性名称相同

完成服务器端添加操作

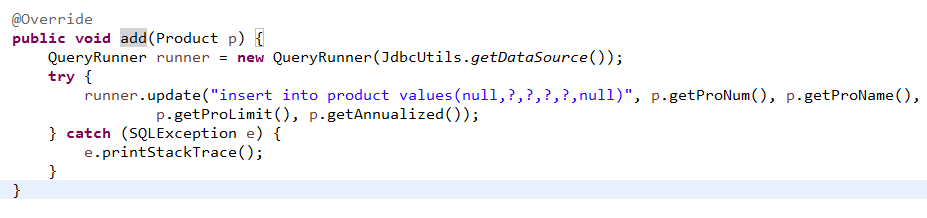
Servlet



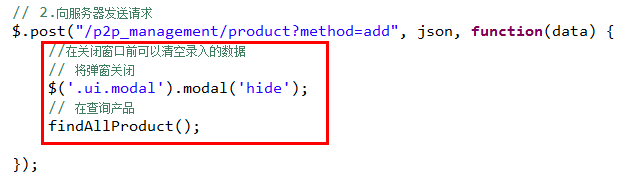
Service



Dao



#### 步骤4 添加完成后，在查询产品在页面上显示



### 产品管理-产品修改

对于修改操作就是进行查询，回显示数据后在修改。

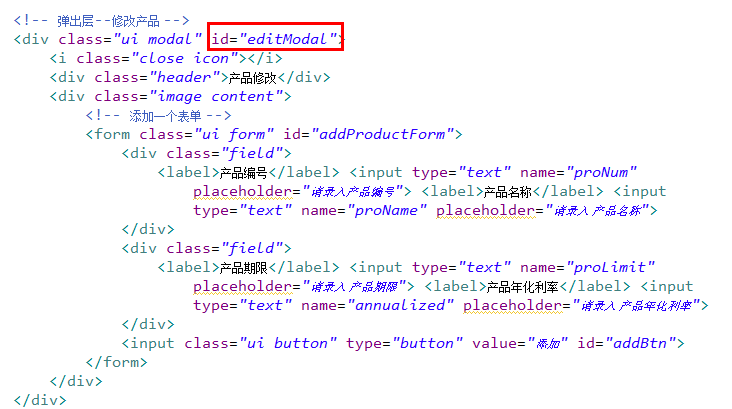
#### 步骤1:给home.jsp页面上的编辑添加连接，点击进行产品的查询

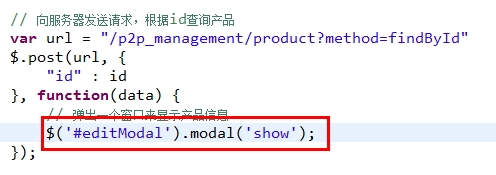
给编辑添加连接



创建了一个findById函数来完成根据产品id查询产品操作

注意:点击编辑后要弹出一个窗口

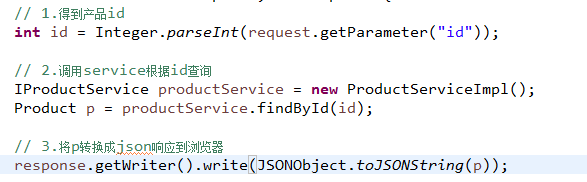




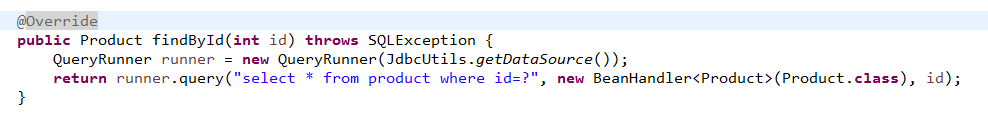
注意:要对添加操作中的弹窗进行修改，添加一个id进行操作

#### 步骤2 在服务器端完成根据id查询产品操作

Servlet

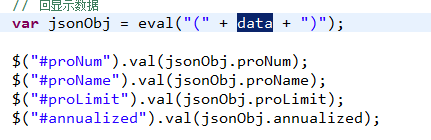


Dao



#### 步骤3 在home.jsp页面上显示查询到的产品信息

对每一个文本框添加id



#### 步骤4 修改操作

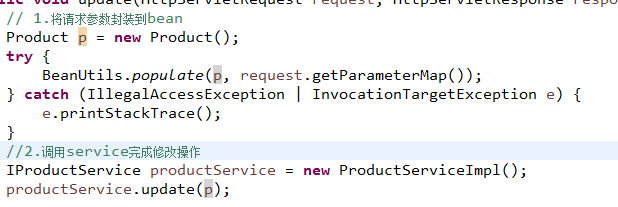
浏览器端操作



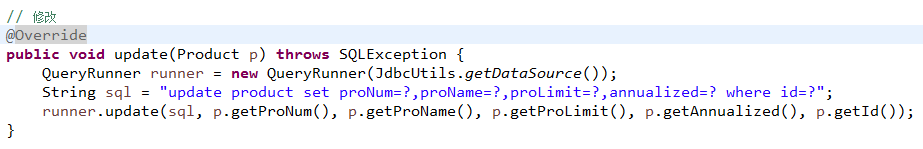
注意:参数中一定要携带id

服务器端操作

Servlet



dao



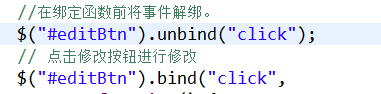
## p2p\_management中bug修复

### 退出中bug



### 产品添加与产品修改中bug

修改中存在的bug修复



产品添加存在的bug修复

