3、Ecshop 实战

项目：

商品管理功能

技术：

Struts2 + Spring + Spring Data JPA + Hibernate + EasyUI

项目源码：

https://github.com/zxjin001/JPA-Project

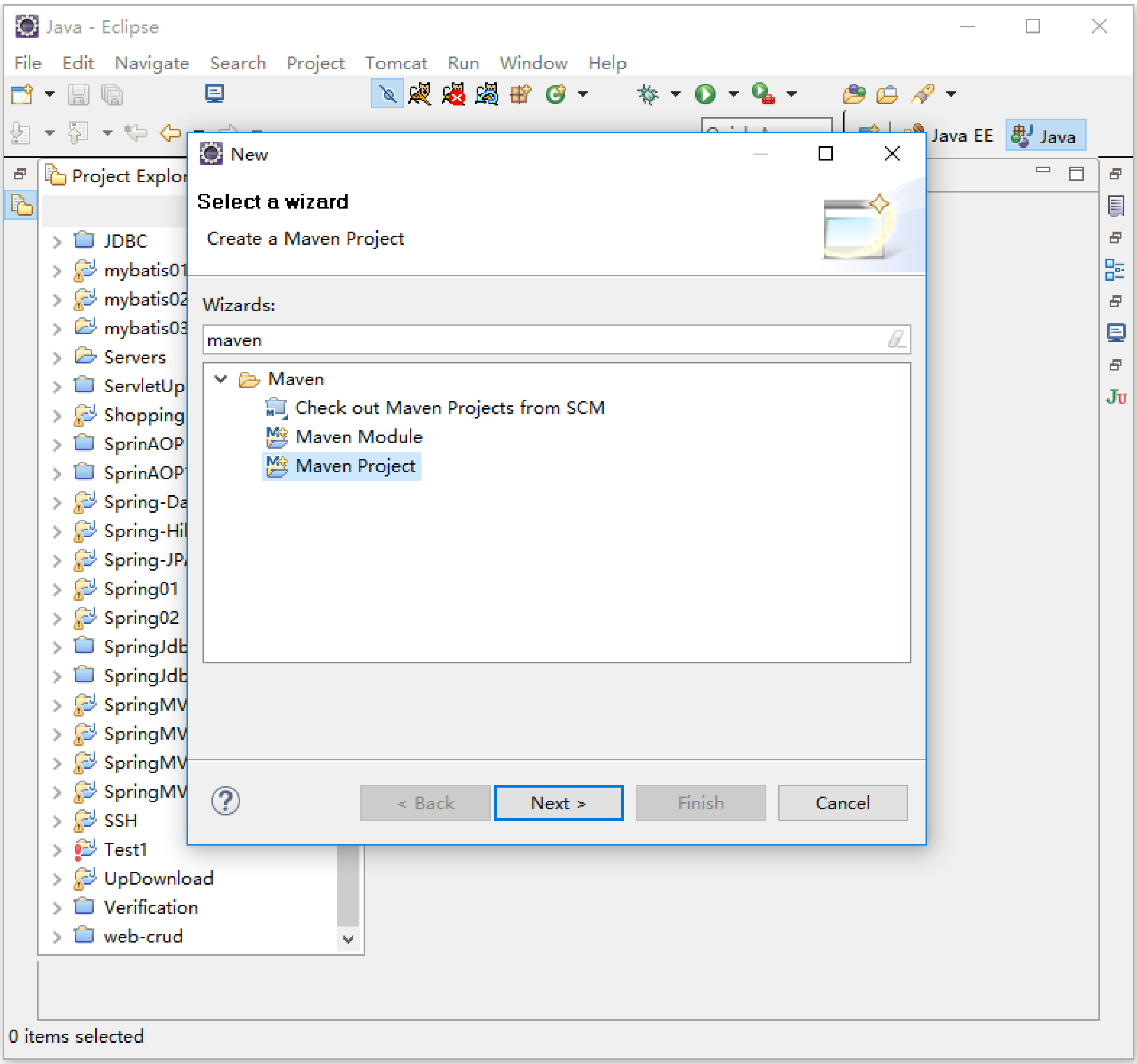
功能：

1、查询所有商品
2、分页查询商品
3、组合条件查询商品
4、商品添加
5、商品编辑
6、商品删除

开发步骤：

1、搭建 Maven 项目环境
2、Spring Data JPA-实体映射
3、Spring Data JPA-查询所有商品
4、Spring Data JPA-分页查询商品
5、Spring Data JPA-组合条件查询商品
6、Spring Data JPA-商品添加
7、Spring Data JPA-商品编辑
8、Spring Data JPA-商品批量删除

1、Maven 项目搭建：



New

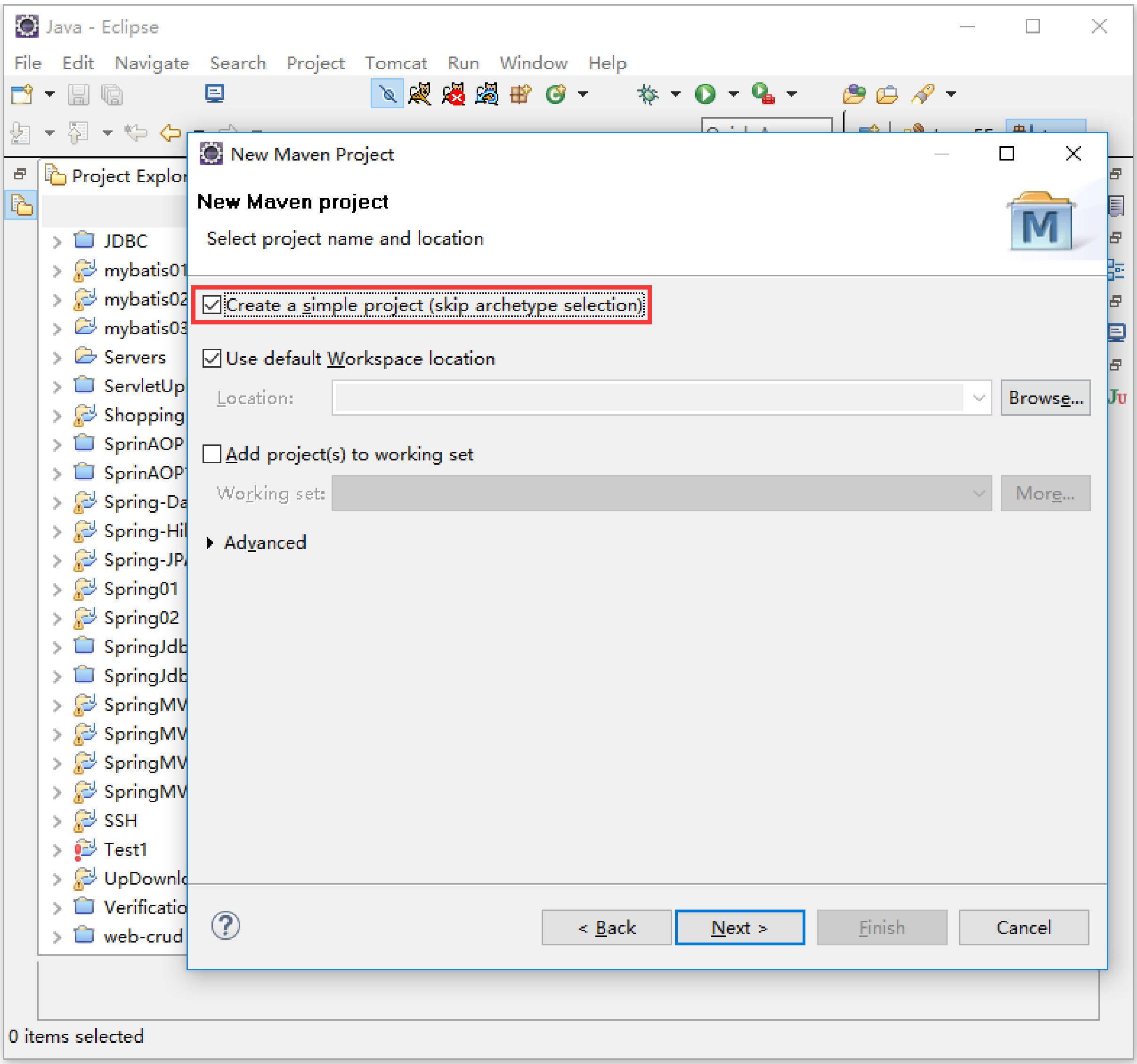
Select a wizard

Create a Maven Project

Wizards: maven

- Maven
  - Check out Maven Projects from SCM
  - Maven Module
  - Maven Project

0 items selected



Java - Eclipse

New Maven Project

New Maven project

Select project name and location

☑ Create a simple project (skip archetype selection)

☑ Use default Workspace location

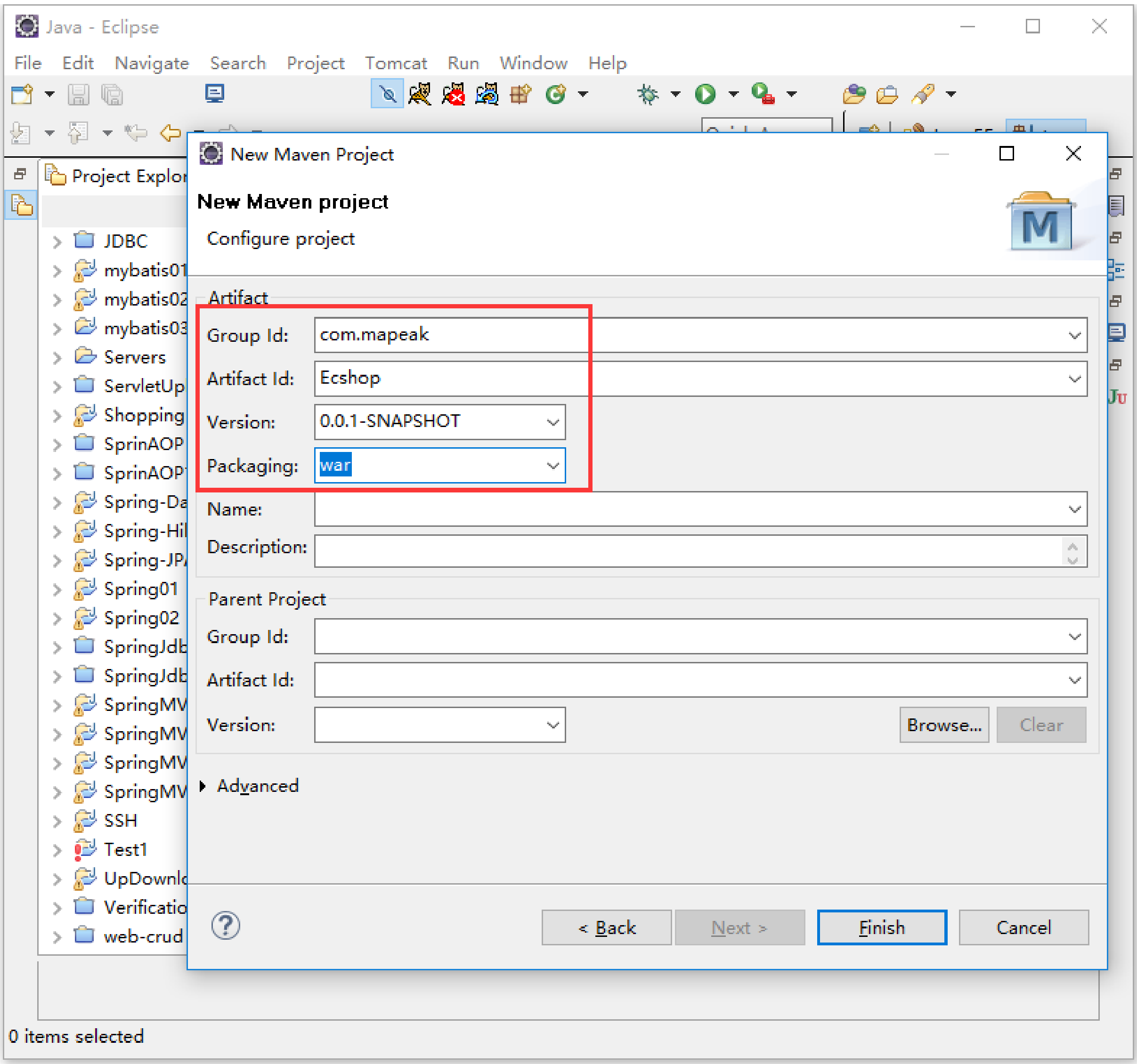
Location:

□ Add project(s) to working set

Working set:

Advanced

0 items selected



New Maven Project

New Maven project

Configure project

Artifact

Group Id: com.mapeak

Artifact Id: Ecshop

Version: 0.0.1-SNAPSHOT

Packaging: war

Name:

Description:

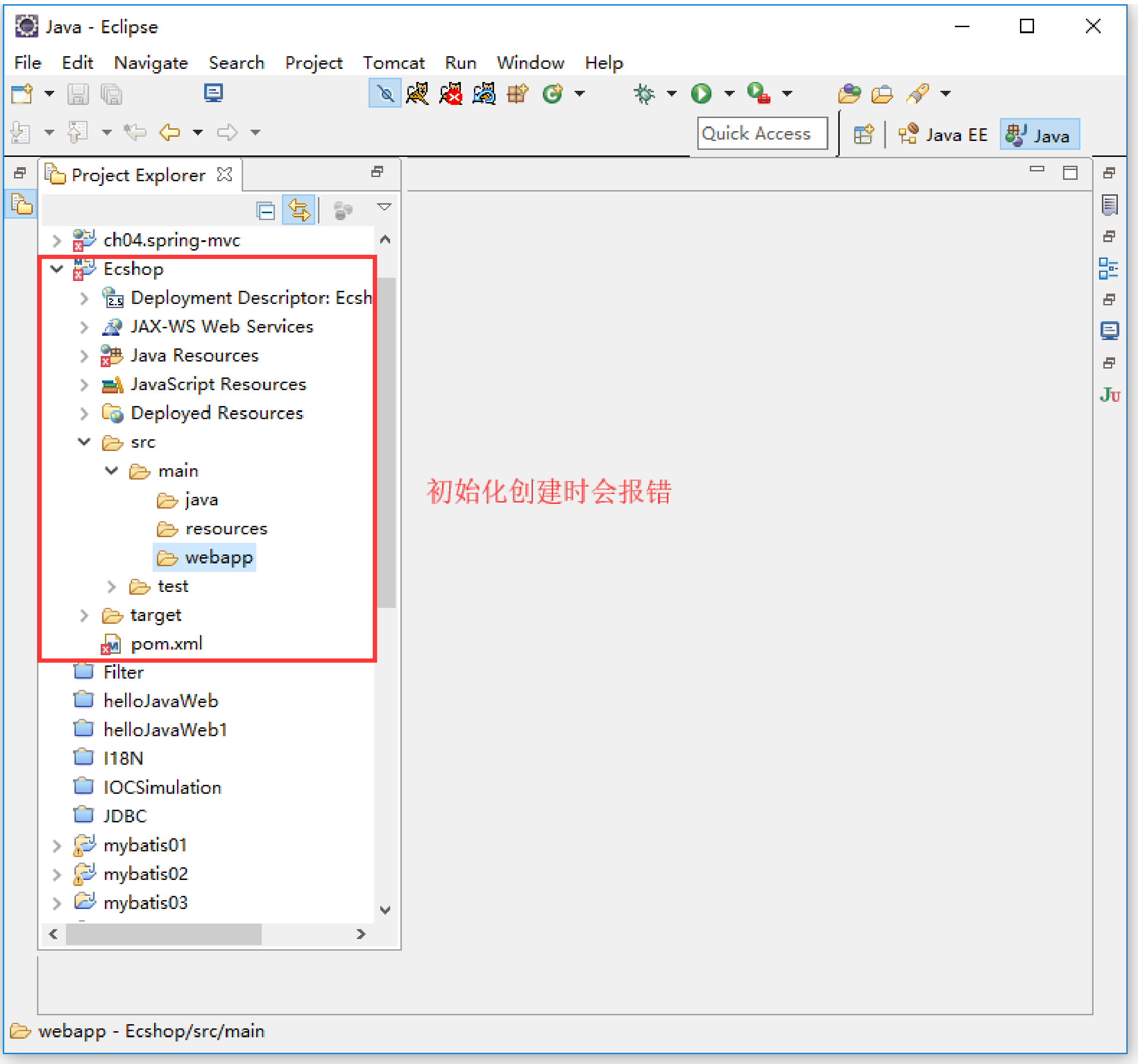
Parent Project

Group Id:

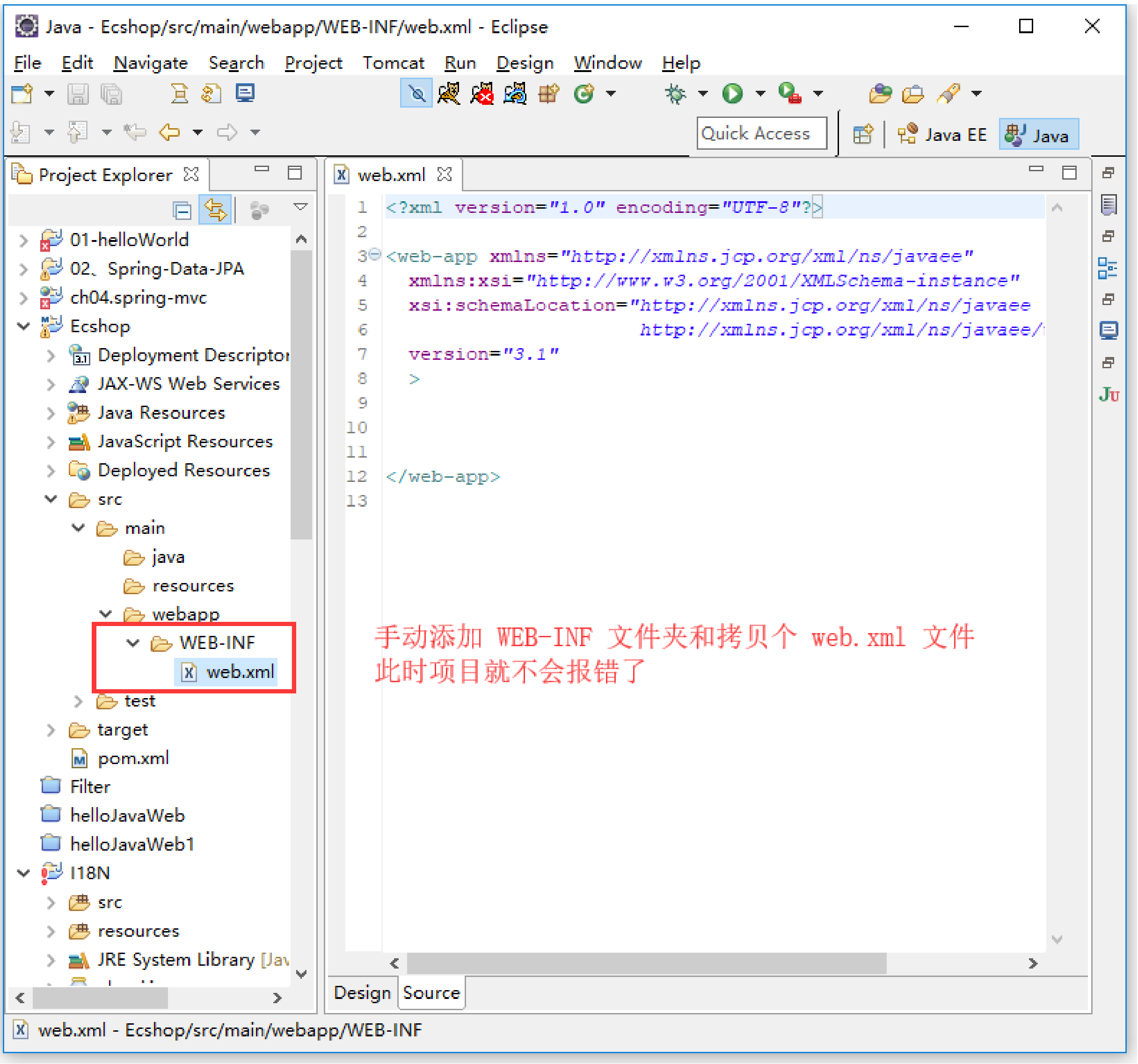
Artifact Id:

Version:

0 items selected

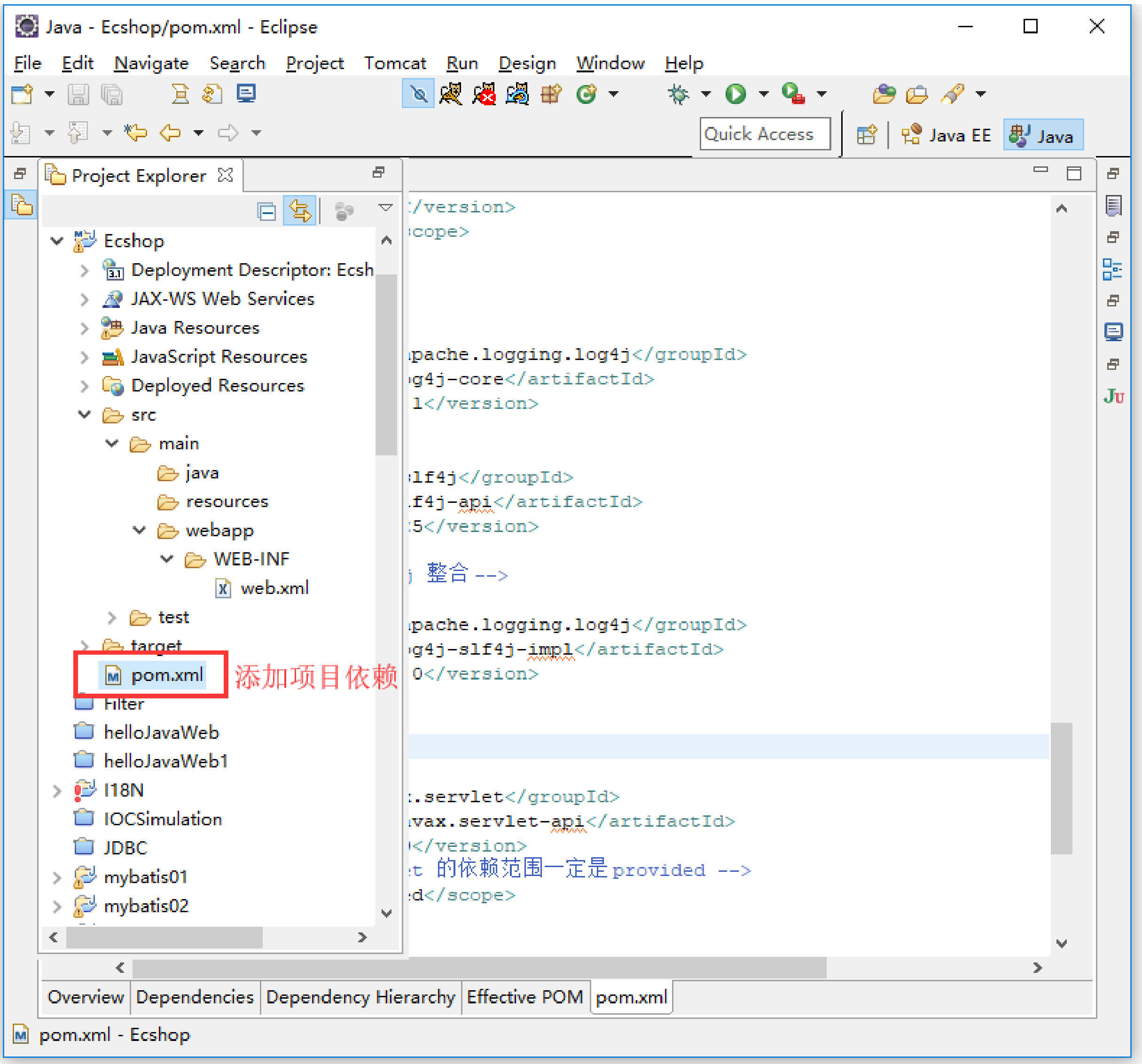

Java - Eclipse
File Edit Navigate Search Project Tomcat Run Window Help
Quick Access
Java EE
Java
Project Explorer
ch04.spring-mvc
Ecshop
Deployment Descriptor: Ecsh
JAX-WS Web Services
Java Resources
JavaScript Resources
Deployed Resources
src
main
java
resources
webapp
test
target
pom.xml
Filter
helloJavaWeb
helloJavaWeb1
I18N
IOCSimulation
JDBC
mybatis01
mybatis02
mybatis03
webapp - Ecshop/src/main

初始化创建时会报错



```xml
<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
                      http://xmlns.jcp.org/xml/ns/javaee/
  version="3.1"
  >



</web-app>

```

手动添加 WEB-INF 文件夹和拷贝个 web.xml 文件
此时项目就不会报错了



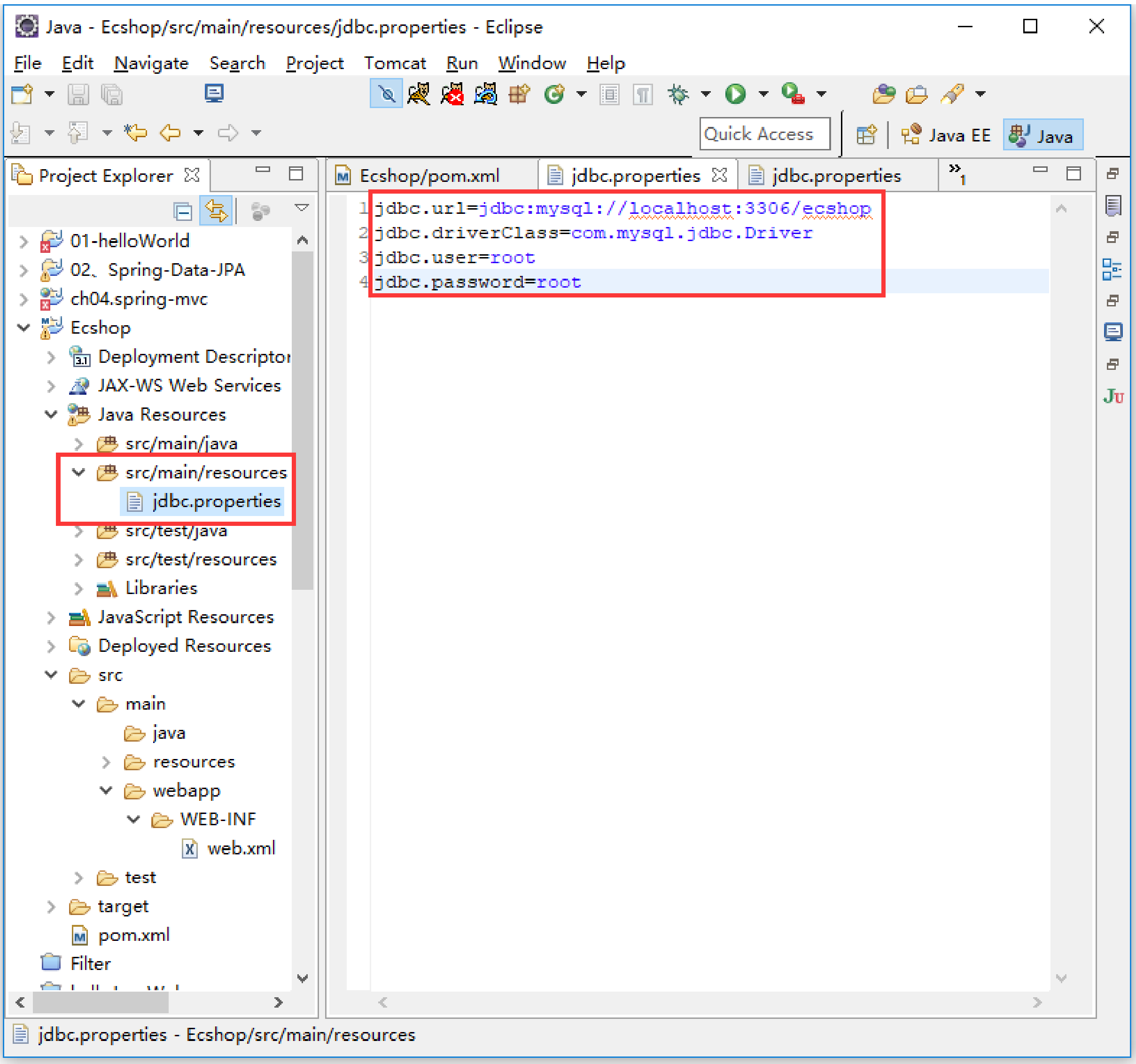
添加项目依赖

```
:/version>
:cope>


pache.logging.log4j</groupId>
g4j-core</artifactId>
1</version>


lf4j</groupId>
f4j-api</artifactId>
5</version>

 整合 -->

pache.logging.log4j</groupId>
g4j-slf4j-impl</artifactId>
0</version>




.servlet</groupId>
vax.servlet-api</artifactId>
</version>
t 的依赖范围一定是provided -->
d</scope>
```

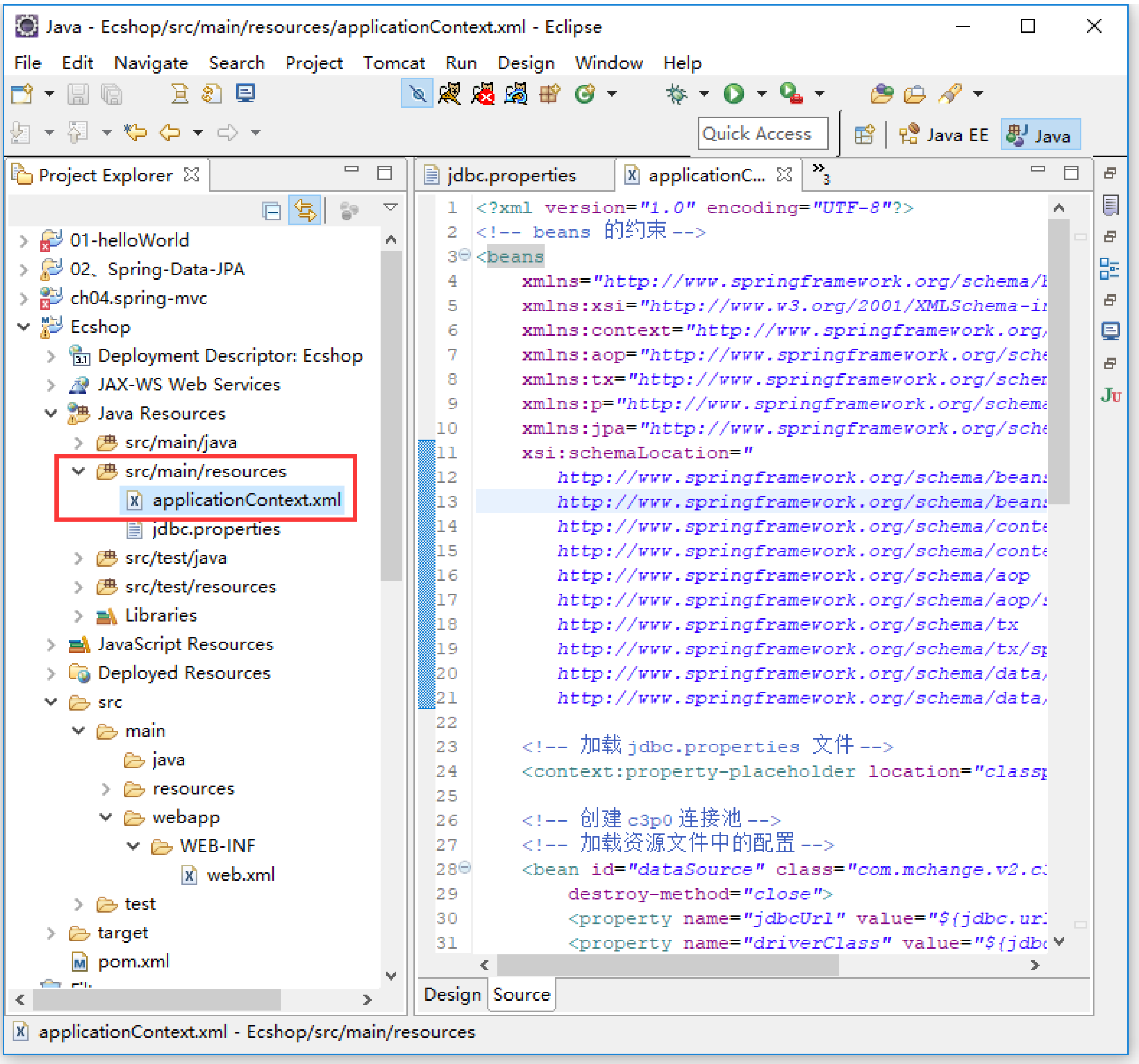
```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.mapeak</groupId>
  <artifactId>Ecshop</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>war</packaging>

  <dependencies>
    <!-- mysql 驱动 -->
    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
      <version>5.1.6</version>
    </dependency>

    <!-- c3p0 连接池 -->
    <dependency>
      <groupId>com.mchange</groupId>
      <artifactId>c3p0</artifactId>
      <version>0.9.5.2</version>
    </dependency>

    <!-- hibernate core -->
    <dependency>
      <groupId>org.hibernate</groupId>
      <artifactId>hibernate-core</artifactId>
      <version>5.2.17.Final</version>
    </dependency>

    <!-- herinate entityManager -->
```

```
<!-- Spring Data JPA 整合 Hibernate 使用 -->
<!-- PS：和 hibernate 版本要保持一致 -->
<dependency>
   <groupId>org.hibernate</groupId>
   <artifactId>hibernate-entitymanager</artifactId>
   <version>5.2.17.Final</version>
</dependency>

<!-- Spring -->
<dependency>
   <groupId>org.springframework</groupId>
   <artifactId>spring-core</artifactId>
   <version>4.3.18.RELEASE</version>
</dependency>
<dependency>
   <groupId>org.springframework</groupId>
   <artifactId>spring-beans</artifactId>
   <version>4.3.18.RELEASE</version>
</dependency>
<dependency>
   <groupId>org.springframework</groupId>
   <artifactId>spring-context</artifactId>
   <version>4.3.18.RELEASE</version>
</dependency>
<dependency>
   <groupId>org.springframework</groupId>
   <artifactId>spring-context-support</artifactId>
   <version>4.3.18.RELEASE</version>
</dependency>
<dependency>
   <groupId>org.springframework</groupId>
   <artifactId>spring-orm</artifactId>
   <version>4.3.18.RELEASE</version>
</dependency>
<dependency>
   <groupId>org.springframework</groupId>
   <artifactId>spring-tx</artifactId>
   <version>4.3.18.RELEASE</version>
</dependency>
<dependency>
   <groupId>org.springframework</groupId>
   <artifactId>spring-aop</artifactId>
   <version>4.3.18.RELEASE</version>
</dependency>
<dependency>
   <groupId>org.springframework</groupId>
   <artifactId>spring-aspects</artifactId>
   <version>4.3.18.RELEASE</version>
</dependency>
<dependency>
   <groupId>org.springframework</groupId>
   <artifactId>spring-web</artifactId>
   <version>4.3.18.RELEASE</version>
</dependency>
<dependency>
   <groupId>org.springframework</groupId>
   <artifactId>spring-test</artifactId>
   <version>4.3.18.RELEASE</version>
</dependency>

<!-- struts2 -->
<dependency>
   <groupId>org.apache.struts</groupId>
   <artifactId>struts2-core</artifactId>
   <version>2.5</version>
</dependency>
<dependency>
   <groupId>org.apache.struts</groupId>
```

```
            <artifactId>struts2-spring-plugin</artifactId>
            <version>2.5</version>
        </dependency>
        <dependency>
            <groupId>org.apache.struts</groupId>
            <artifactId>struts2-convention-plugin</artifactId>
            <version>2.5</version>
        </dependency>

        <!-- spring data jpa -->
        <dependency>
            <groupId>org.springframework.data</groupId>
            <artifactId>spring-data-jpa</artifactId>
            <version>1.11.9.RELEASE</version>
        </dependency>

        <!-- junit -->
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>4.12</version>
            <scope>test</scope>
        </dependency>

        <!-- log4j -->
        <dependency>
            <groupId>org.apache.logging.log4j</groupId>
            <artifactId>log4j-core</artifactId>
            <version>2.8.2</version>
        </dependency>
        <dependency>
            <groupId>org.slf4j</groupId>
            <artifactId>slf4j-api</artifactId>
            <version>1.7.25</version>
        </dependency>
        <!-- log4j 和 slf4j 整合 -->
        <dependency>
            <groupId>org.apache.logging.log4j</groupId>
            <artifactId>log4j-slf4j-impl</artifactId>
            <version>2.8.2</version>
            <!-- 必须添加依赖范围：test -->
            <scope>test</scope>
        </dependency>

        <!-- servlet -->
        <dependency>
            <groupId>javax.servlet</groupId>
            <artifactId>javax.servlet-api</artifactId>
            <version>3.1.0</version>
            <!-- PS:servlet 的依赖范围一定是 provided -->
            <scope>provided</scope>
        </dependency>
    </dependencies>
</project>
```



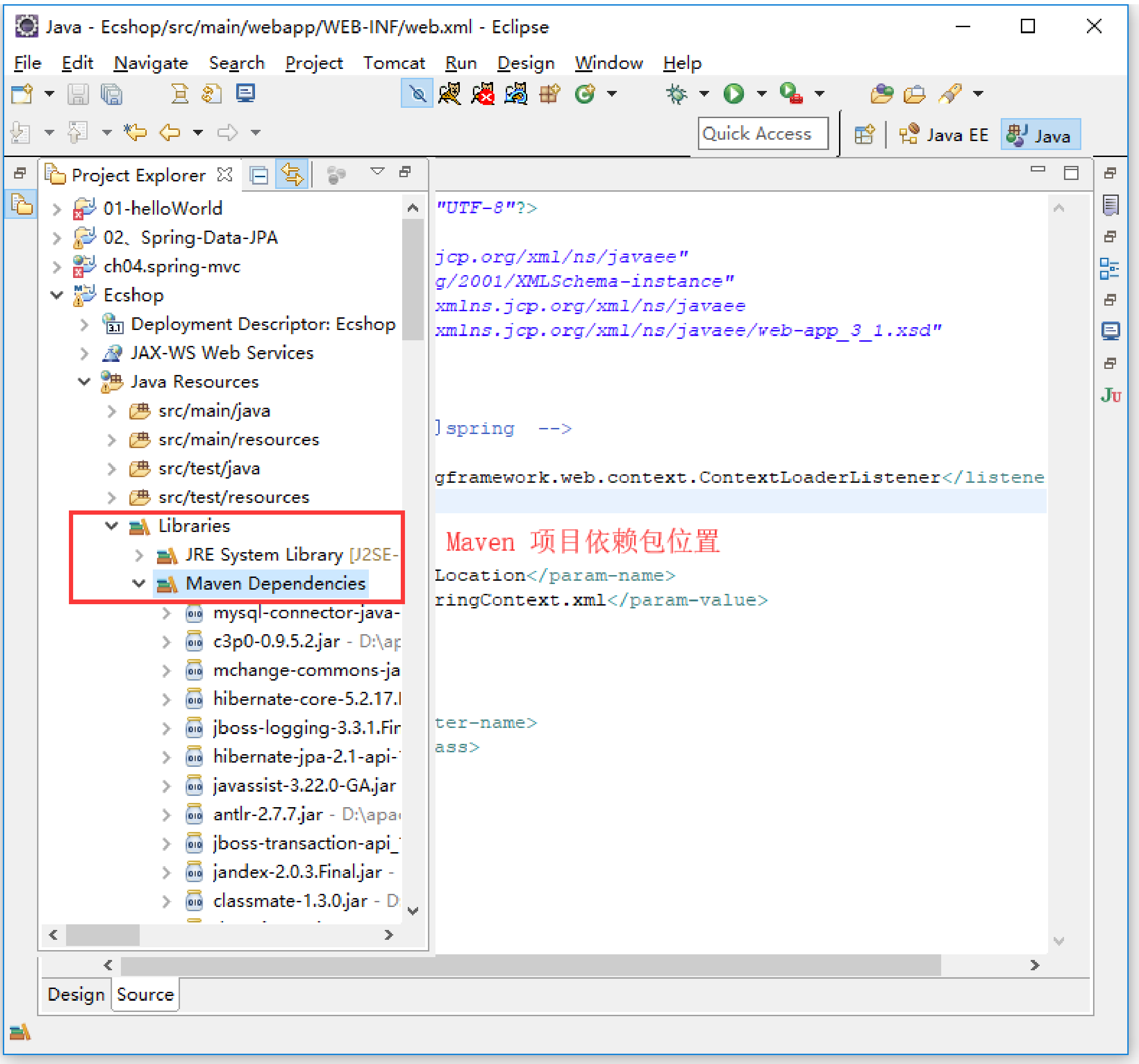
```
jdbc.url=jdbc:mysql://localhost:3306/ecshop
jdbc.driverClass=com.mysql.jdbc.Driver
jdbc.user=root
jdbc.password=root
```



```
<?xml version="1.0" encoding="UTF-8"?>
<!-- beans 的约束 -->
<beans
    xmlns="http://www.springframework.org/schema/b
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-in
    xmlns:context="http://www.springframework.org/
    xmlns:aop="http://www.springframework.org/sche
    xmlns:tx="http://www.springframework.org/schem
    xmlns:p="http://www.springframework.org/schema
    xmlns:jpa="http://www.springframework.org/sche
    xsi:schemaLocation="
        http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/conte
        http://www.springframework.org/schema/conte
        http://www.springframework.org/schema/aop
        http://www.springframework.org/schema/aop/s
        http://www.springframework.org/schema/tx
        http://www.springframework.org/schema/tx/sp
        http://www.springframework.org/schema/data/
        http://www.springframework.org/schema/data/

    <!-- 加载 jdbc.properties 文件 -->
    <context:property-placeholder location="classp

    <!-- 创建 c3p0 连接池 -->
    <!-- 加载资源文件中的配置 -->
    <bean id="dataSource" class="com.mchange.v2.c3
        destroy-method="close">
        <property name="jdbcUrl" value="${jdbc.url
        <property name="driverClass" value="${jdbc
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- beans 的约束 -->
<beans
  xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:context="http://www.springframework.org/schema/context"
  xmlns:aop="http://www.springframework.org/schema/aop"
  xmlns:tx="http://www.springframework.org/schema/tx"
  xmlns:p="http://www.springframework.org/schema/p"
  xmlns:jpa="http://www.springframework.org/schema/data/jpa"
  xsi:schemaLocation="
    http://www.springframework.org/schema/beans
      http://www.springframework.org/schema/beans/spring-beans.xsd
     http://www.springframework.org/schema/context
     http://www.springframework.org/schema/context/spring-context.xsd
     http://www.springframework.org/schema/aop
     http://www.springframework.org/schema/aop/spring-aop.xsd
     http://www.springframework.org/schema/tx
    http://www.springframework.org/schema/tx/spring-tx.xsd
    http://www.springframework.org/schema/data/jpa
    http://www.springframework.org/schema/data/jpa/spring-jpa.xsd">

  <!-- 加载 jdbc.properties 文件 -->
  <context:property-placeholder location="classpath:jdbc.properties" />

  <!-- 创建 c3p0 连接池 -->
  <!-- 加载资源文件中的配置 -->
  <bean id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource"
     destroy-method="close">
     <property name="jdbcUrl" value="${jdbc.url}" />
     <property name="driverClass" value="${jdbc.driverClass}" />
```

```
      <property name="user" value="${jdbc.user}" />
      <property name="password" value="${jdbc.password}" />
  </bean>

  <!-- Spring 整合  JPA 配置 -->
  <!-- 配置  EntityManagerFactory -->
  <bean id="entityManagerFactory"
      class="org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean">
      <property name="dataSource" ref="dataSource" />
      <!-- JPA 底层封装了  Hibernate，依赖  Hibernate -->
      <property name="jpaVendorAdapter">
      <bean class="org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter">
          <!-- hibernate 属性 -->
          <!-- 数据库 -->
          <property name="database" value="MYSQL" />
          <!-- 自动建表 -->
          <property name="generateDdl" value="true" />
          <!-- 显示 SQL -->
          <property name="showSql" value="true" />
        </bean>
      </property>
      <property name="packagesToScan">
        <!-- 扫描实体所在的包 -->
        <list>
          <value>com.mapeak.ecshop.domain</value>
        </list>
      </property>
  </bean>

  <!-- 事务管理器 -->
  <!-- 使用注解的方式使用事务 -->
  <tx:annotation-driven transaction-manager="transactionManager" />
  <bean id="transactionManager"
      class="org.springframework.orm.jpa.JpaTransactionManager">
      <property name="entityManagerFactory" ref="entityManagerFactory" />
  </bean>

  <!-- 开启事务注解 -->
  <tx:annotation-driven transaction-manager="transactionManager" />

  <!-- 开启 Spring IOC注解功能 -->
  <!-- base-package：IOC 注解的类所在包，即包的范围，Spring 也会扫描该包的子包 -->
  <context:component-scan base-package="com.mapeak.ecshop" />

  <!-- spring data jpa 扫描包 -->
  <jpa:repositories base-package="com.mapeak.ecshop.dao" />
</beans>
```

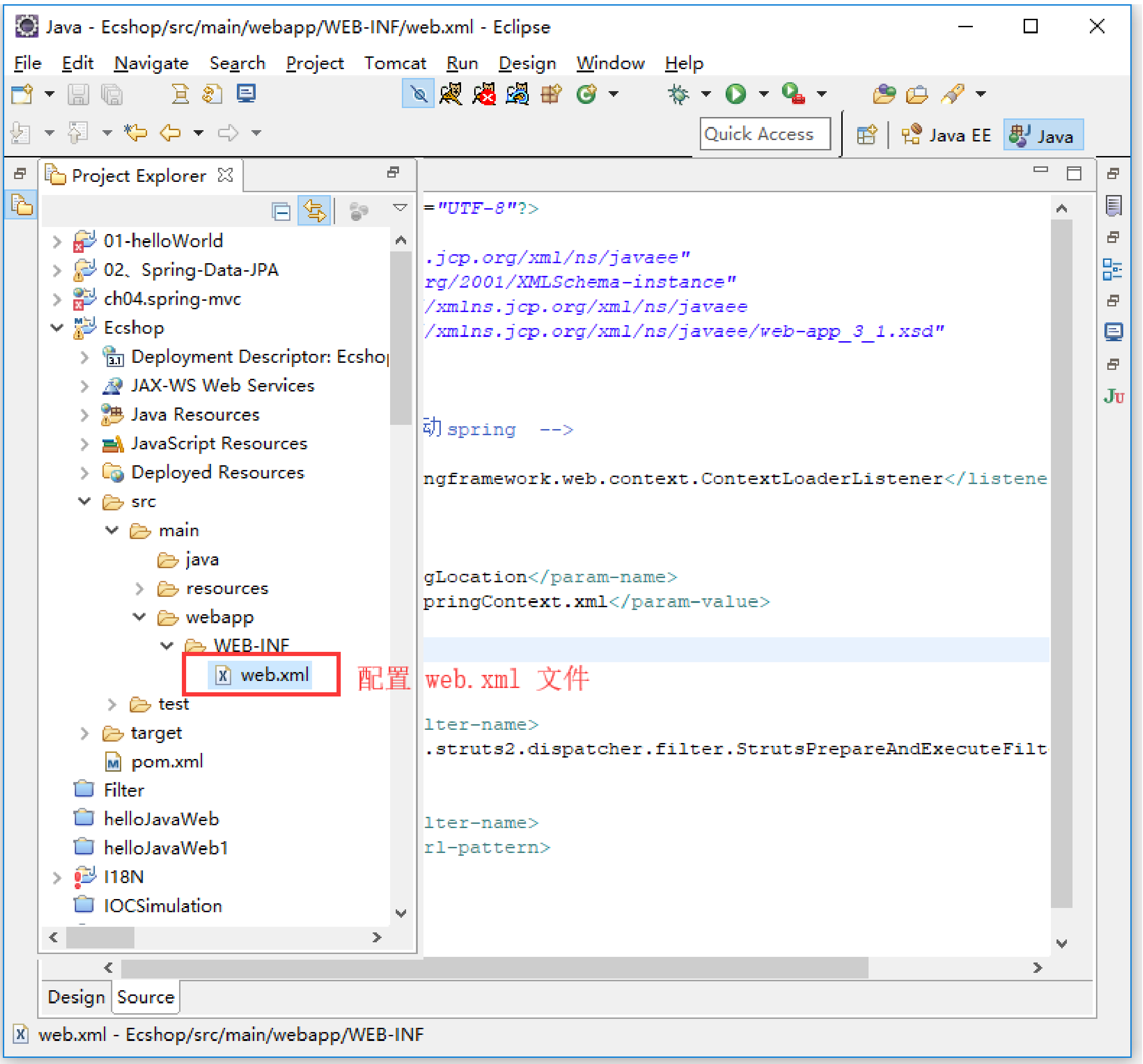


Java - Ecshop/src/main/webapp/WEB-INF/web.xml - Eclipse
File Edit Navigate Search Project Tomcat Run Design Window Help
Quick Access
Java EE
Java
Project Explorer
Design
Source

```
01-helloWorld
02、Spring-Data-JPA
ch04.spring-mvc
Ecshop
  Deployment Descriptor: Ecshop
  JAX-WS Web Services
  Java Resources
    src/main/java
    src/main/resources
    src/test/java
    src/test/resources
    Libraries
      JRE System Library [J2SE-
      Maven Dependencies
        mysql-connector-java-
        c3p0-0.9.5.2.jar - D:\ap
        mchange-commons-ja
        hibernate-core-5.2.17.I
        jboss-logging-3.3.1.Fir
        hibernate-jpa-2.1-api-
        javassist-3.22.0-GA.jar
        antlr-2.7.7.jar - D:\apa
        jboss-transaction-api_
        jandex-2.0.3.Final.jar -
        classmate-1.3.0.jar - D
```

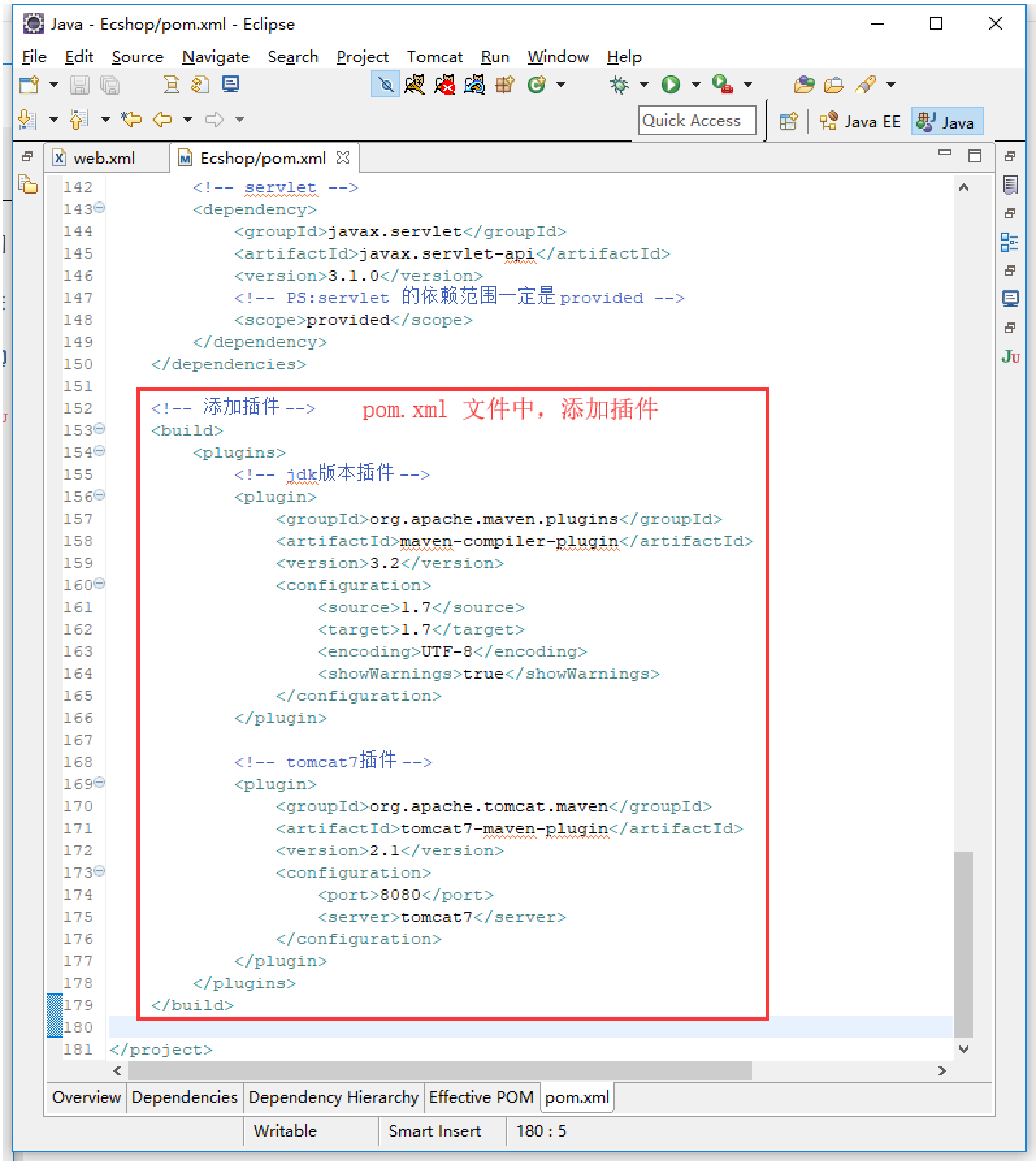
```
"UTF-8"?>

jcp.org/xml/ns/javaee"
g/2001/XMLSchema-instance"
xmlns.jcp.org/xml/ns/javaee
xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"


]spring  -->

gframework.web.context.ContextLoaderListener</listene

Location</param-name>
ringContext.xml</param-value>



ter-name>
ass>
```

Maven 项目依赖包位置



配置 web. xml 文件

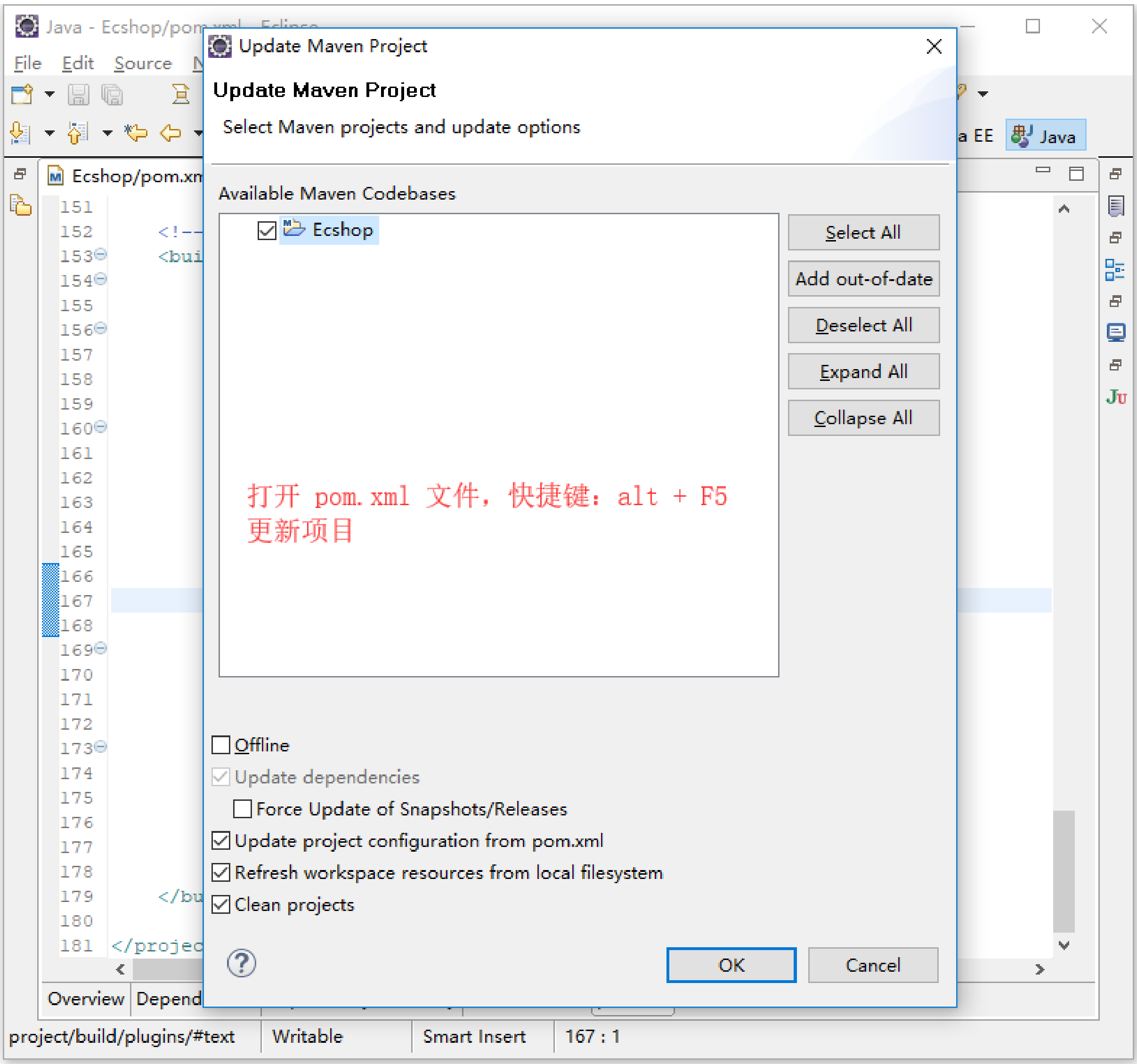
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
            http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
 version="3.1"
 >

  <!-- spring 的 web 监听器启动 spring  -->
 <listener>
    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
  </listener>
  <!-- 修改加载路径 -->
  <context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>classpath:applicationContext.xml</param-value>
  </context-param>

  <!-- 启动 Struts2 -->
  <filter>
    <filter-name>struts2</filter-name>
    <filter-class>org.apache.struts2.dispatcher.filter.StrutsPrepareAndExecuteFilter</filter-class>
  </filter>
  <filter-mapping>
    <filter-name>struts2</filter-name>
    <url-pattern>*.action</url-pattern>
  </filter-mapping>
</web-app>
```



Java - Ecshop/pom.xml - Eclipse
File Edit Source Navigate Search Project Tomcat Run Window Help
Quick Access
Java EE
Java
web.xml
Ecshop/pom.xml

```
        <!-- servlet -->
        <dependency>
            <groupId>javax.servlet</groupId>
            <artifactId>javax.servlet-api</artifactId>
            <version>3.1.0</version>
            <!-- PS:servlet 的依赖范围一定是provided -->
            <scope>provided</scope>
        </dependency>
    </dependencies>

    <!-- 添加插件 -->
    <build>
        <plugins>
            <!-- jdk版本插件 -->
            <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>
                    <showWarnings>true</showWarnings>
                </configuration>
            </plugin>

            <!-- tomcat7插件 -->
            <plugin>
                <groupId>org.apache.tomcat.maven</groupId>
                <artifactId>tomcat7-maven-plugin</artifactId>
                <version>2.1</version>
                <configuration>
                    <port>8080</port>
                    <server>tomcat7</server>
                </configuration>
            </plugin>
        </plugins>
    </build>

</project>
```

pom. xml 文件中，添加插件



Update Maven Project

Update Maven Project

Select Maven projects and update options

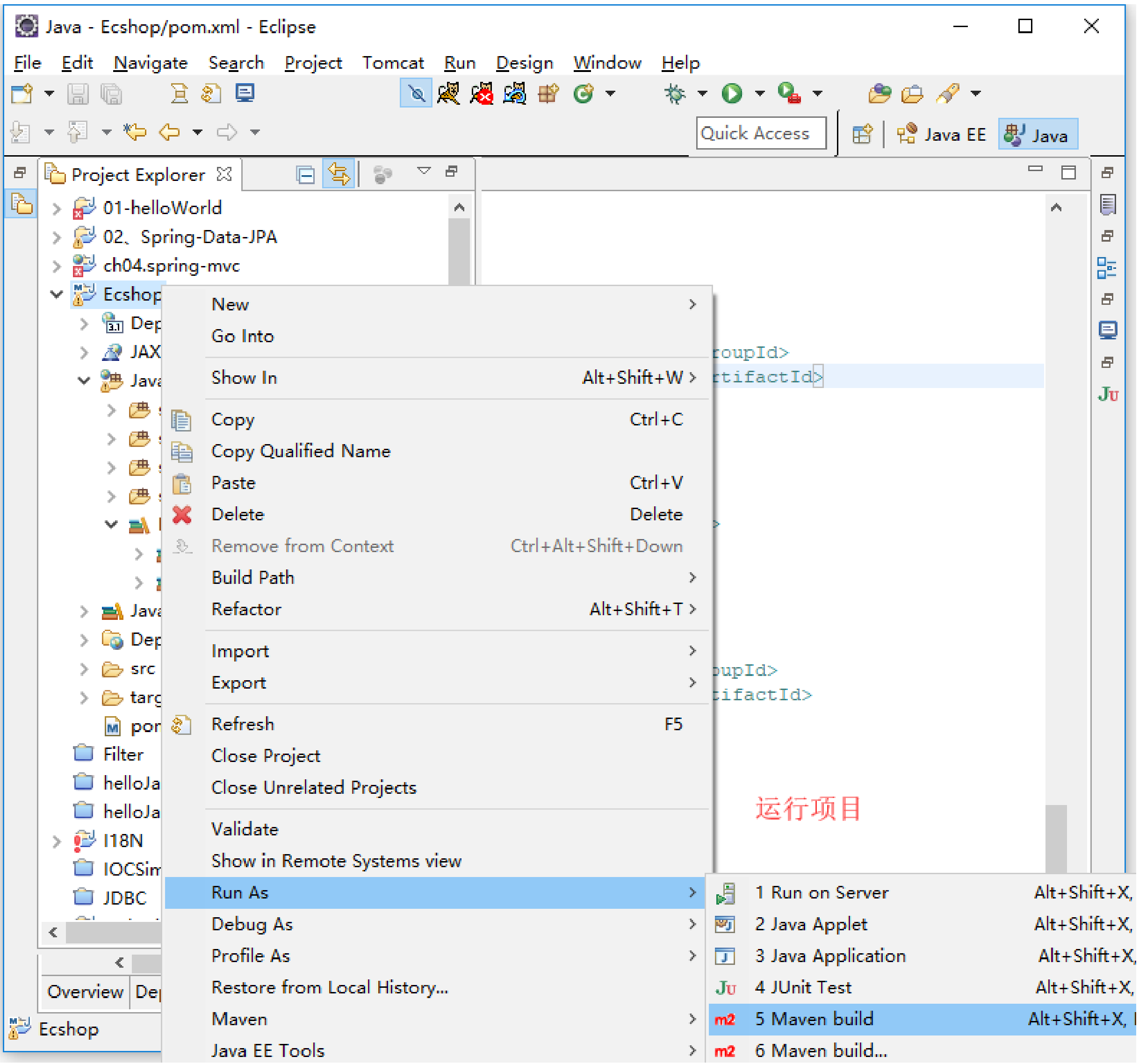
Available Maven Codebases

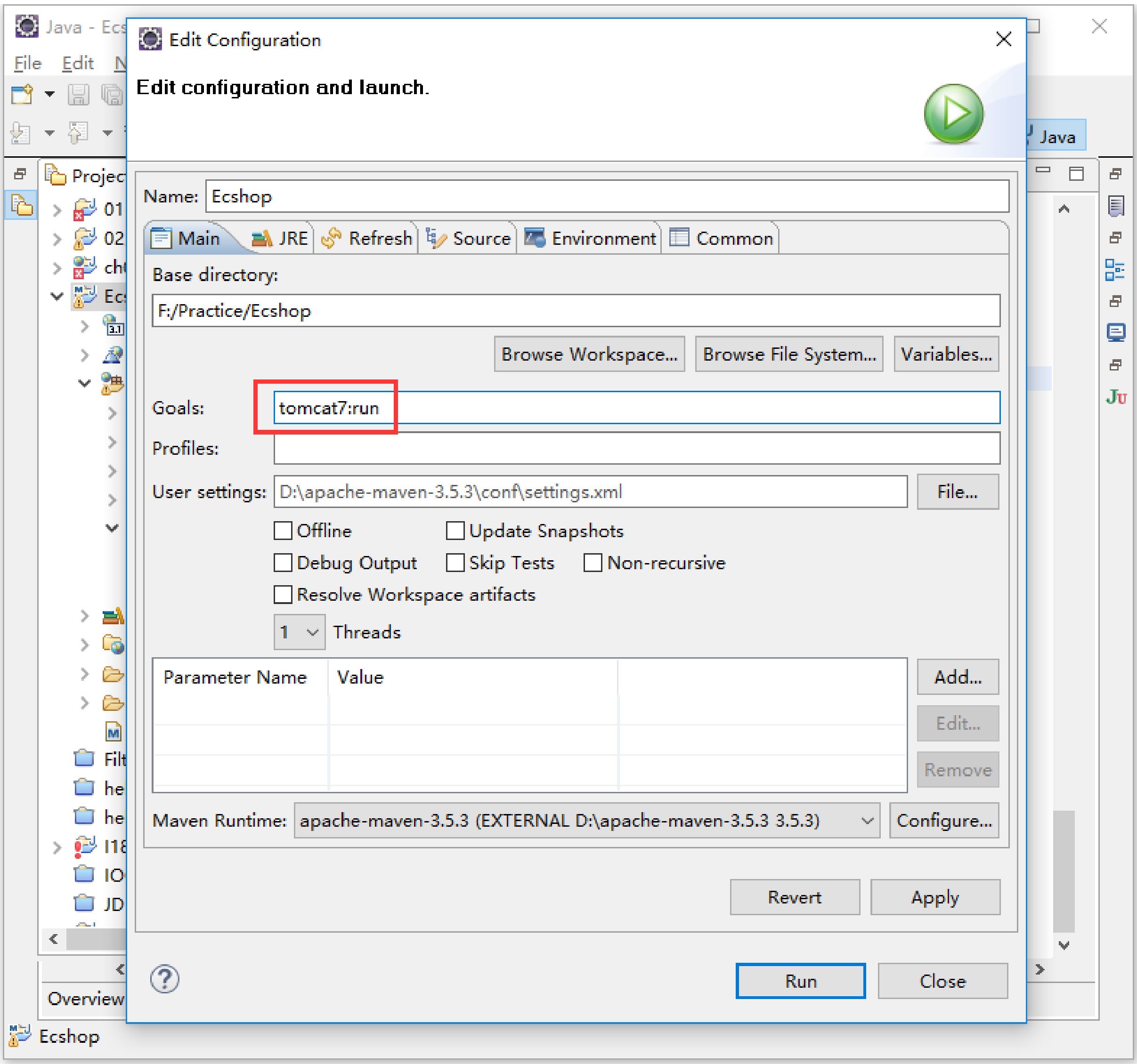
☑ Ecshop

打开 pom.xml 文件，快捷键：alt + F5 更新项目

□ Offline
☑ Update dependencies
□ Force Update of Snapshots/Releases
☑ Update project configuration from pom.xml
☑ Refresh workspace resources from local filesystem
☑ Clean projects

project/build/plugins/#text | Writable | Smart Insert | 167 : 1


Java - Ecshop/pom.xml - Eclipse
File Edit Navigate Search Project Tomcat Run Design Window Help
Quick Access
Java EE
Java
Project Explorer
01-helloWorld
02、Spring-Data-JPA
ch04.spring-mvc
Ecshop
src
Filter
I18N
JDBC
Overview
New
Go Into
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Build Path
Refactor Alt+Shift+T
Import
Export
Refresh F5
Close Project
Close Unrelated Projects
Validate
Show in Remote Systems view
Run As
Debug As
Profile As
Restore from Local History...
Maven
Java EE Tools
1 Run on Server Alt+Shift+X,
2 Java Applet Alt+Shift+X,
3 Java Application Alt+Shift+X,
4 JUnit Test Alt+Shift+X,
5 Maven build Alt+Shift+X,
6 Maven build...
运行项目



Edit Configuration

Edit configuration and launch.

Name: Ecshop

Main | JRE | Refresh | Source | Environment | Common

Base directory:
F:/Practice/Ecshop

Browse Workspace... | Browse File System... | Variables...

Goals: tomcat7:run

Profiles:

User settings: D:\apache-maven-3.5.3\conf\settings.xml | File...

□ Offline □ Update Snapshots
□ Debug Output □ Skip Tests □ Non-recursive
□ Resolve Workspace artifacts
1 Threads

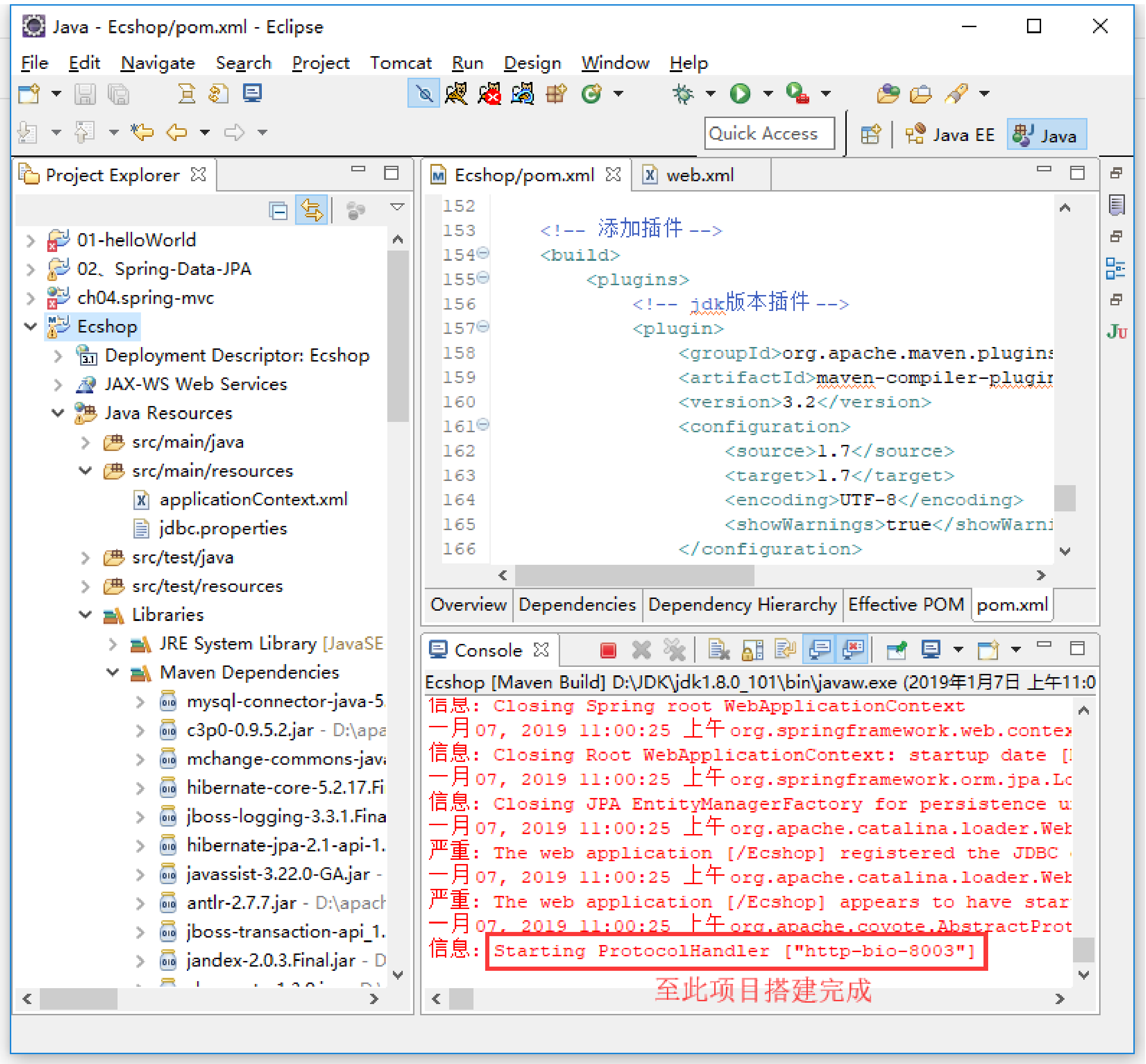
Parameter Name	Value

Add... | Edit... | Remove

Maven Runtime: apache-maven-3.5.3 (EXTERNAL D:\apache-maven-3.5.3 3.5.3) | Configure...

Revert | Apply

Run | Close



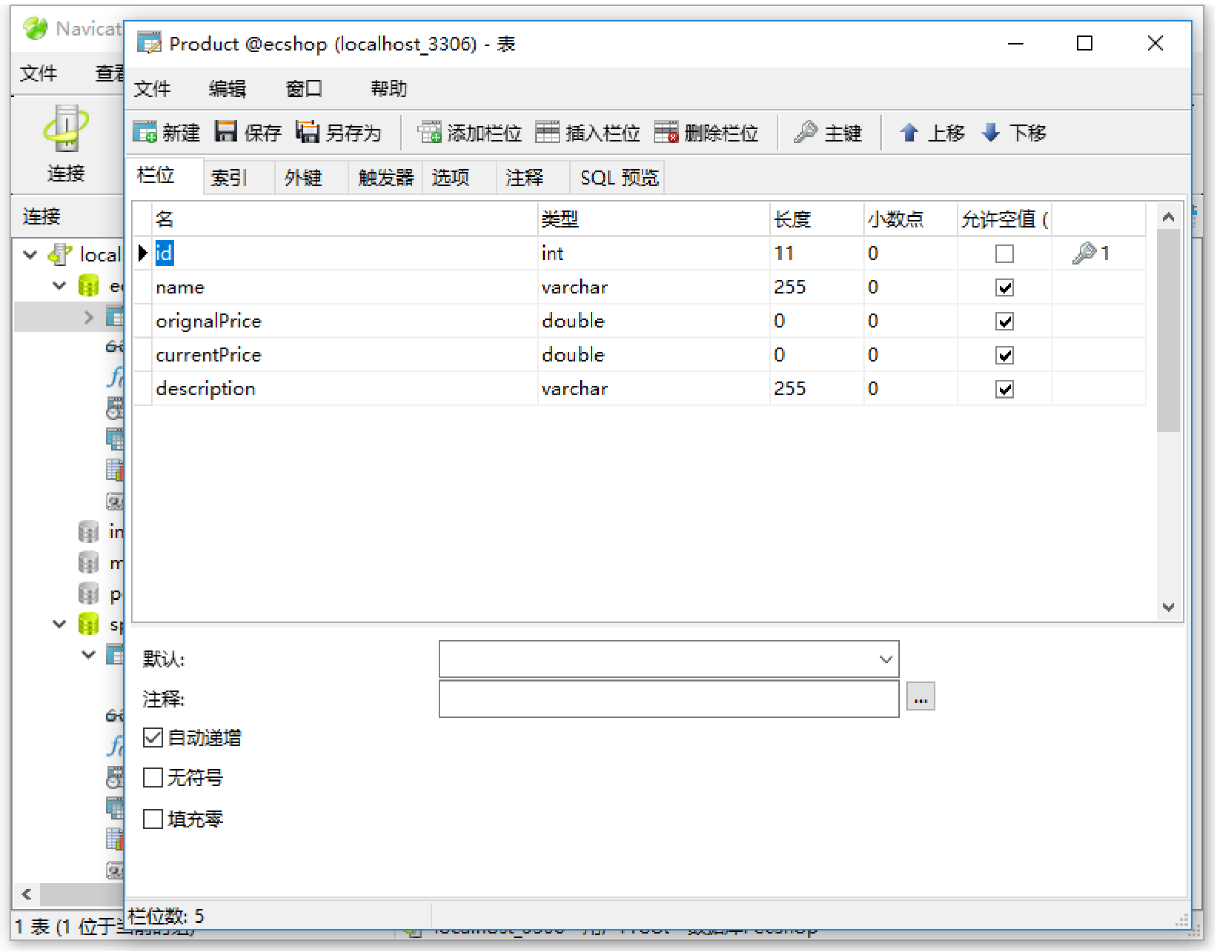
```
    <!-- 添加插件 -->
    <build>
        <plugins>
            <!-- jdk版本插件 -->
            <plugin>
                <groupId>org.apache.maven.plugins
                <artifactId>maven-compiler-plugin
                <version>3.2</version>
                <configuration>
                    <source>1.7</source>
                    <target>1.7</target>
                    <encoding>UTF-8</encoding>
                    <showWarnings>true</showWarni
                </configuration>
```

```
Ecshop [Maven Build] D:\JDK\jdk1.8.0_101\bin\javaw.exe (2019年1月7日 上午11:0
信息: Closing Spring root WebApplicationContext
一月 07, 2019 11:00:25 上午 org.springframework.web.contex
信息: Closing Root WebApplicationContext: startup date [
一月 07, 2019 11:00:25 上午 org.springframework.orm.jpa.Lo
信息: Closing JPA EntityManagerFactory for persistence u
一月 07, 2019 11:00:25 上午 org.apache.catalina.loader.Web
严重: The web application [/Ecshop] registered the JDBC
一月 07, 2019 11:00:25 上午 org.apache.catalina.loader.Web
严重: The web application [/Ecshop] appears to have star
一月 07, 2019 11:00:25 上午 org.apache.coyote.AbstractProt
信息: Starting ProtocolHandler ["http-bio-8003"]
```

至此项目搭建完成

2、Spring Data JPA-实体映射：

建表：



名	类型	长度	小数点	允许空值 (	
id	int	11	0	□	🔑1
name	varchar	255	0	☑	
orignalPrice	double	0	0	☑	
currentPrice	double	0	0	☑	
description	varchar	255	0	☑	

默认:

注释:

☑ 自动递增

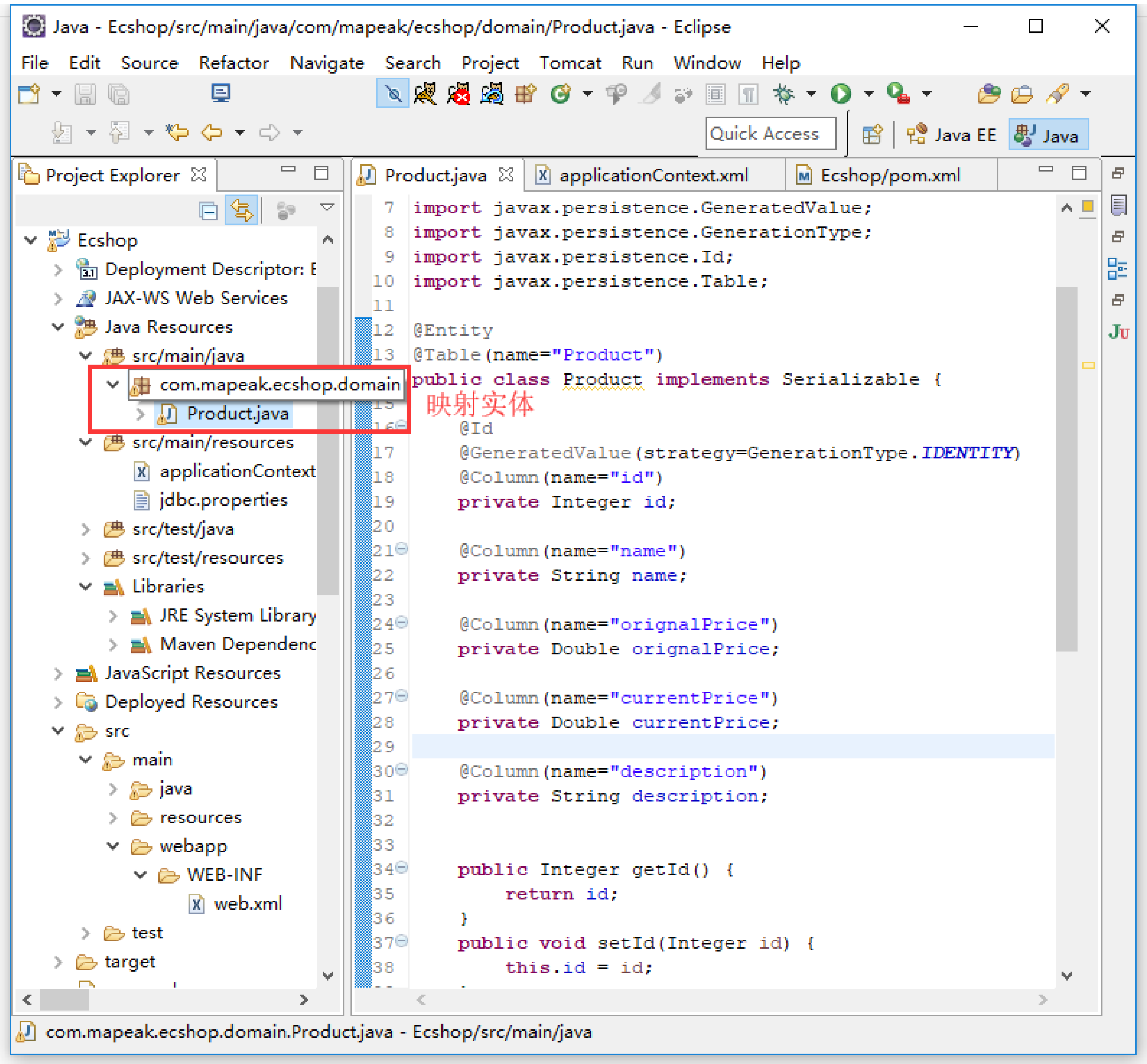
□ 无符号

□ 填充零

栏位数: 5

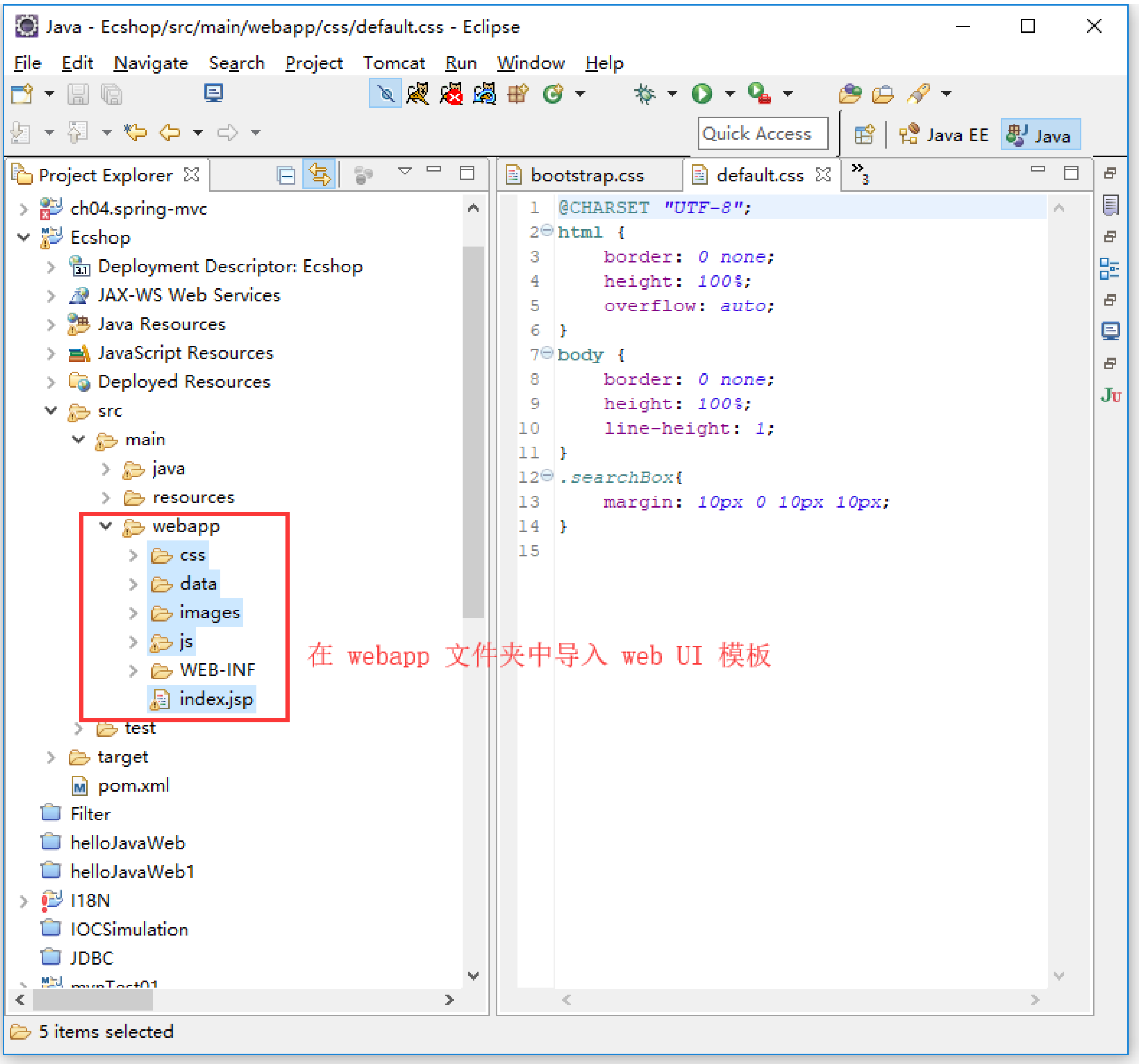
1 表 (1 位于当前的组)

映射实体:



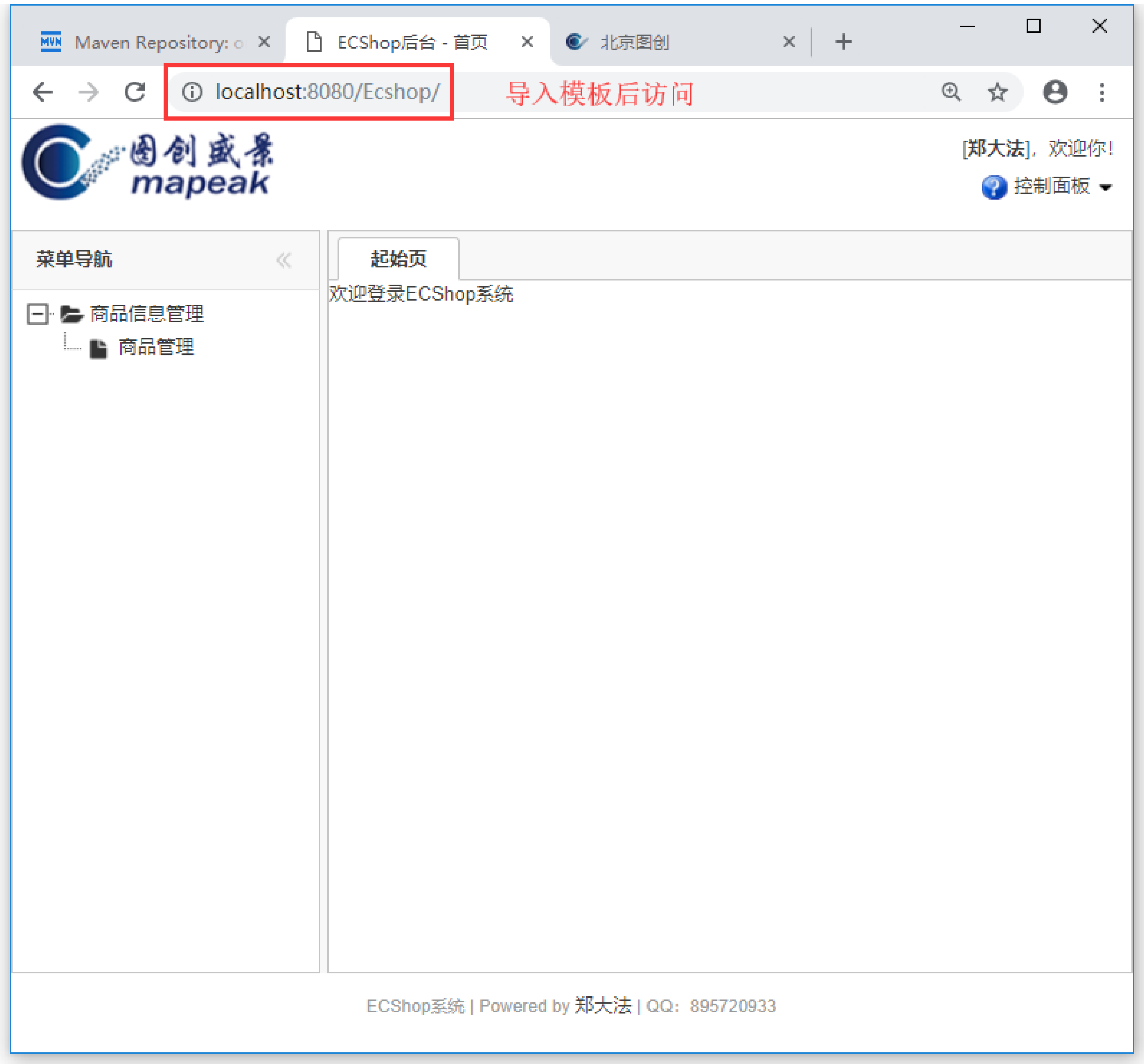
```java
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="Product")
public class Product implements Serializable {

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    @Column(name="id")
    private Integer id;

    @Column(name="name")
    private String name;

    @Column(name="orignalPrice")
    private Double orignalPrice;

    @Column(name="currentPrice")
    private Double currentPrice;

    @Column(name="description")
    private String description;


    public Integer getId() {
        return id;
    }
    public void setId(Integer id) {
        this.id = id;
```

3、Spring Data JPA-查询所有商品：



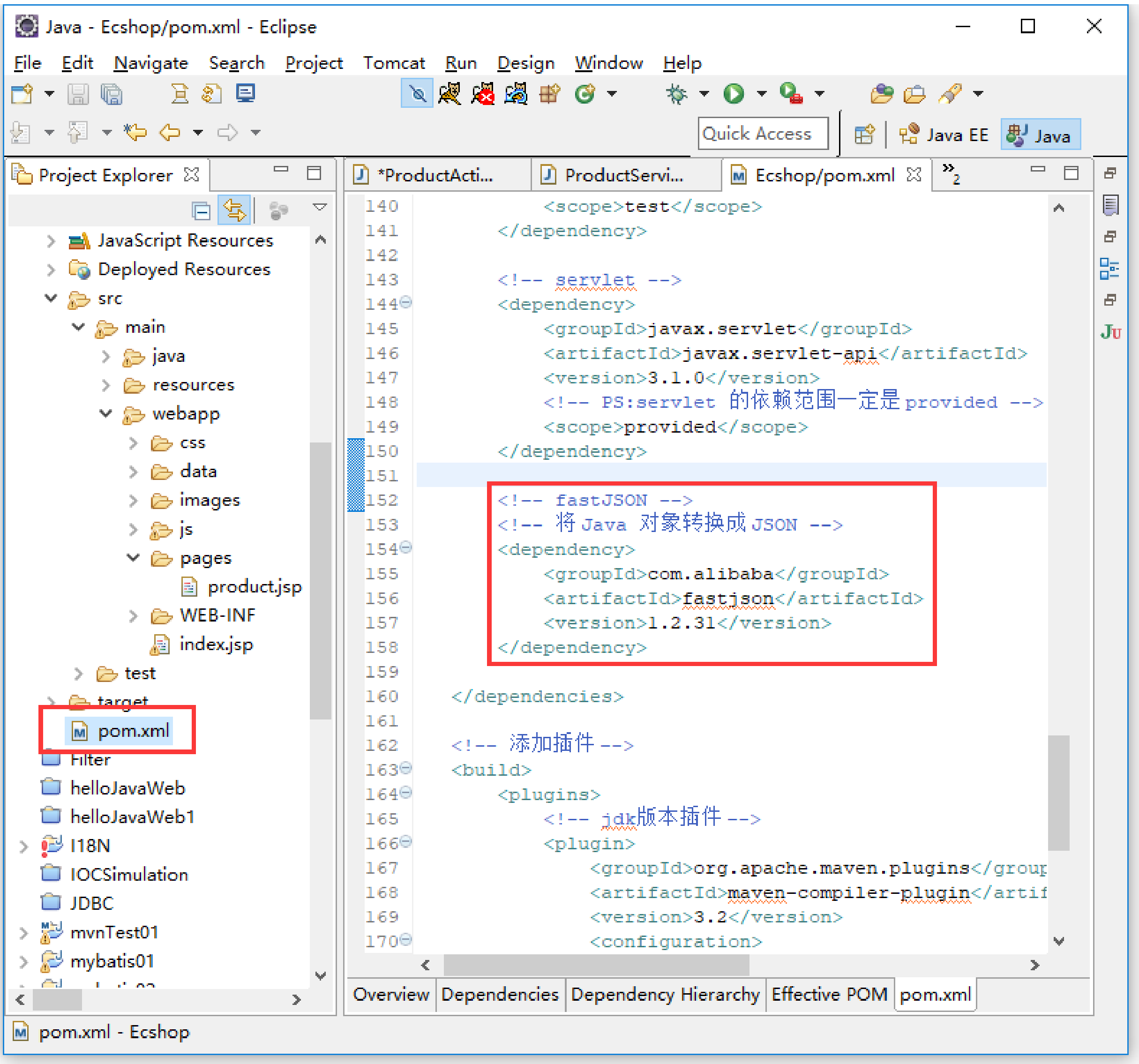
```
@CHARSET "UTF-8";
html {
    border: 0 none;
    height: 100%;
    overflow: auto;
}
body {
    border: 0 none;
    height: 100%;
    line-height: 1;
}
.searchBox{
    margin: 10px 0 10px 10px;
}

```

在 webapp 文件夹中导入 web UI 模板



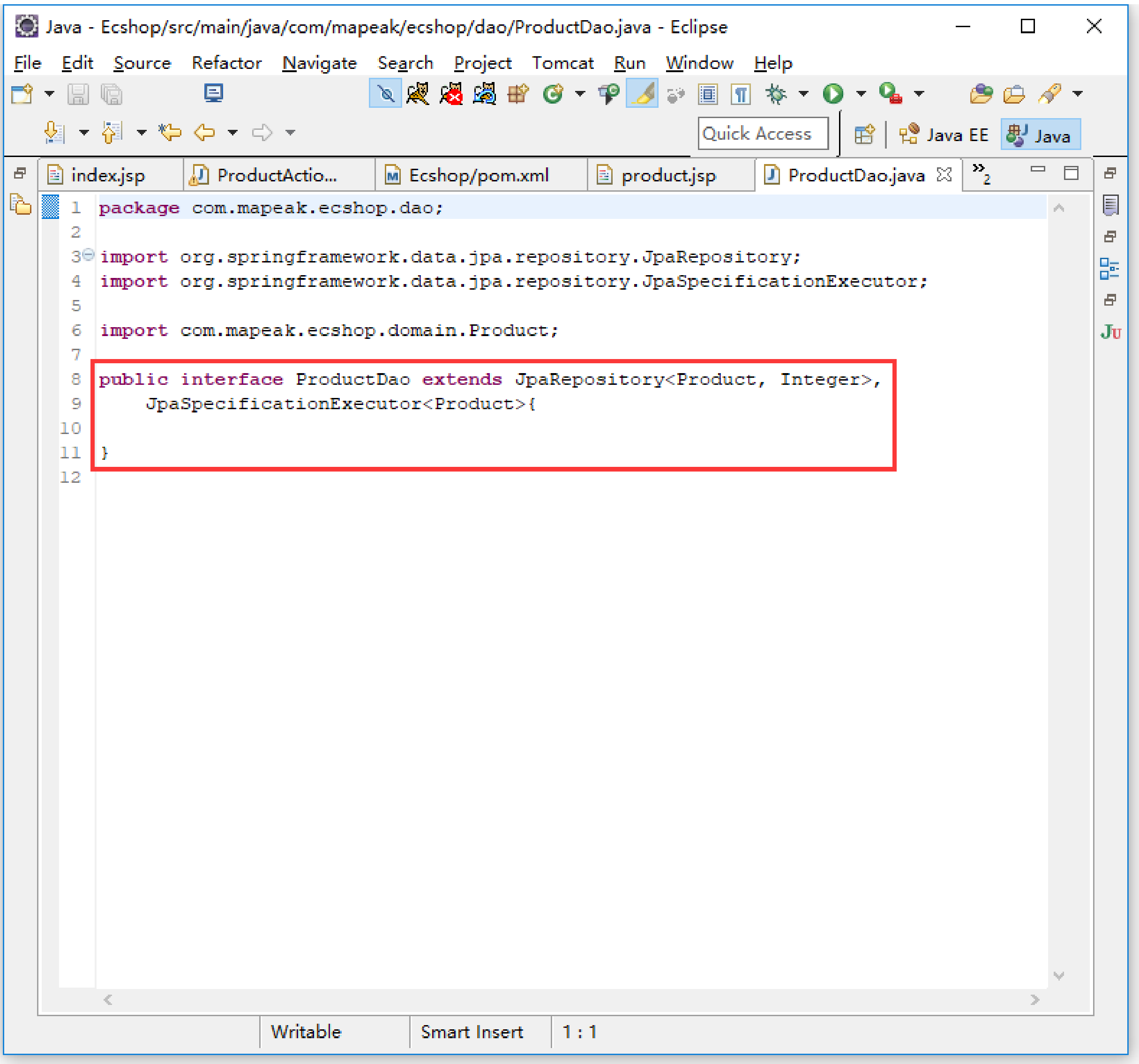
欢迎登录ECShop系统

ECShop系统 | Powered by 郑大法 | QQ：895720933



```
            <scope>test</scope>
        </dependency>

        <!-- servlet -->
        <dependency>
            <groupId>javax.servlet</groupId>
            <artifactId>javax.servlet-api</artifactId>
            <version>3.1.0</version>
            <!-- PS:servlet 的依赖范围一定是 provided -->
            <scope>provided</scope>
        </dependency>

        <!-- fastJSON -->
        <!-- 将 Java 对象转换成 JSON -->
        <dependency>
            <groupId>com.alibaba</groupId>
            <artifactId>fastjson</artifactId>
            <version>1.2.31</version>
        </dependency>

    </dependencies>

    <!-- 添加插件 -->
    <build>
        <plugins>
            <!-- jdk版本插件 -->
            <plugin>
                <groupId>org.apache.maven.plugins</group
                <artifactId>maven-compiler-plugin</artif
                <version>3.2</version>
                <configuration>
```

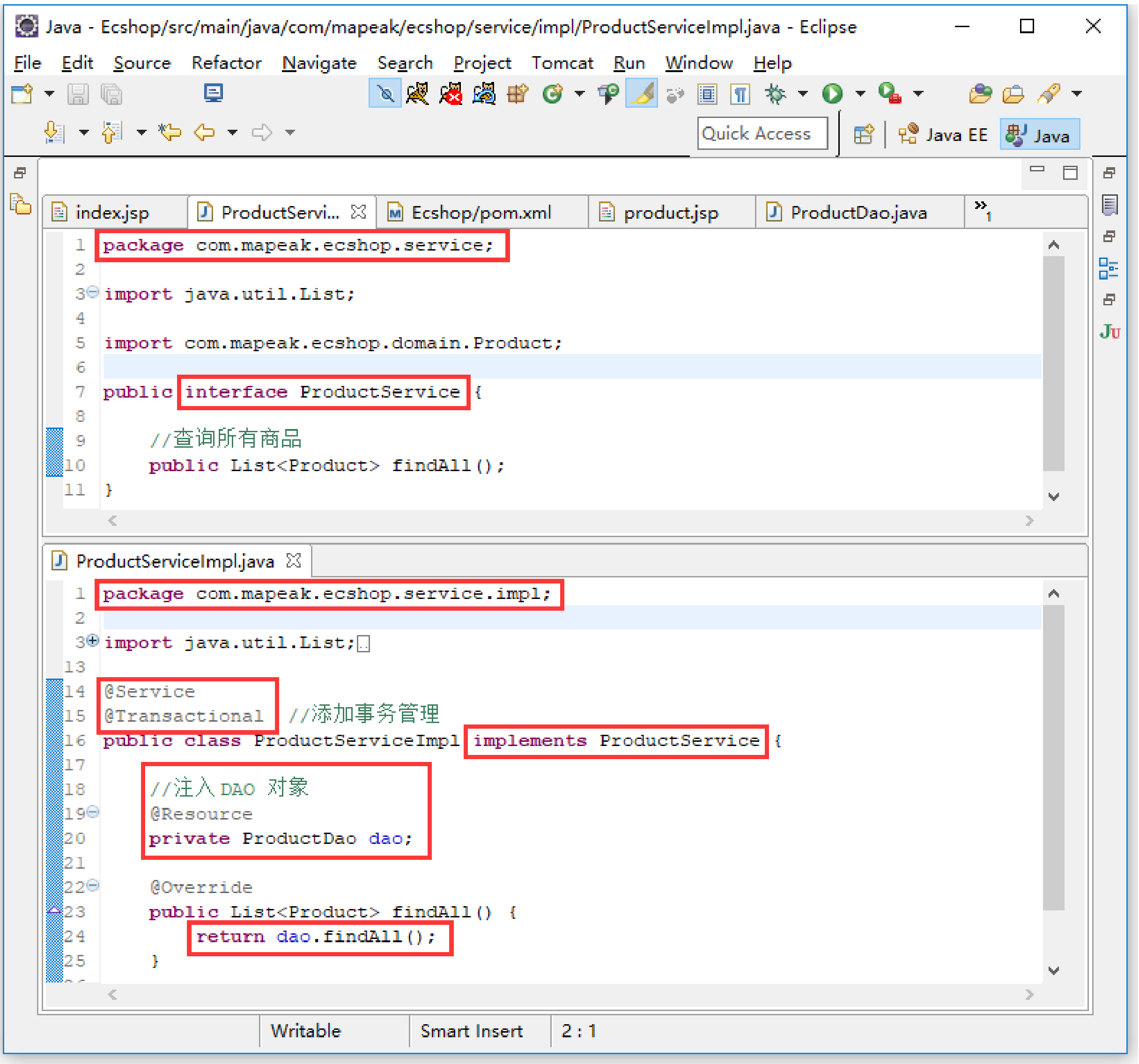
dao:



Java - Ecshop/src/main/java/com/mapeak/ecshop/dao/ProductDao.java - Eclipse
File Edit Source Refactor Navigate Search Project Tomcat Run Window Help
Quick Access
Java EE
Java
index.jsp
ProductActio...
Ecshop/pom.xml
product.jsp
ProductDao.java
Writable
Smart Insert
1 : 1

```java
package com.mapeak.ecshop.dao;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.JpaSpecificationExecutor;

import com.mapeak.ecshop.domain.Product;

public interface ProductDao extends JpaRepository<Product, Integer>,
    JpaSpecificationExecutor<Product>{

}
```

service:

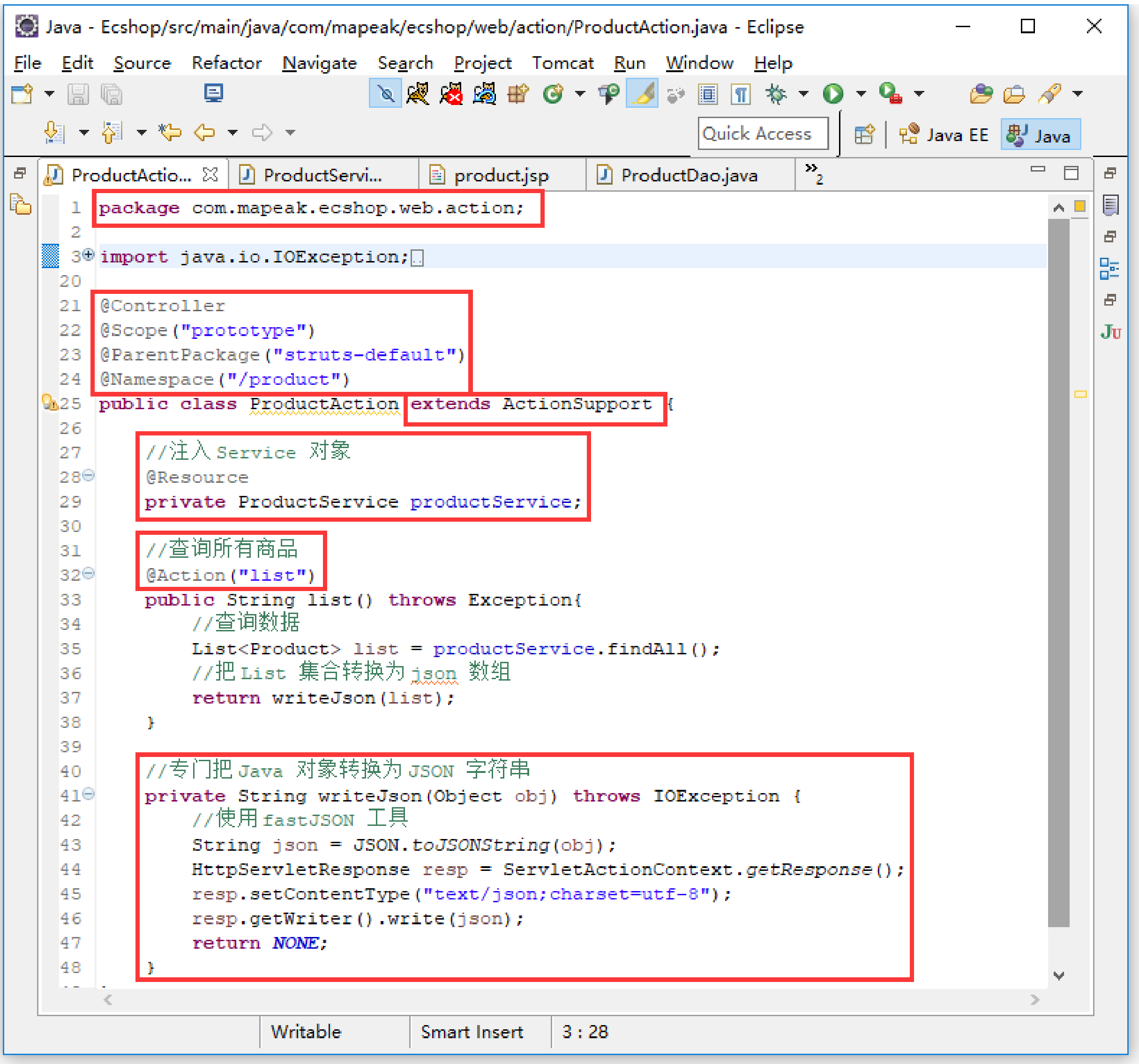


Java - Ecshop/src/main/java/com/mapeak/ecshop/service/impl/ProductServiceImpl.java - Eclipse
File Edit Source Refactor Navigate Search Project Tomcat Run Window Help
Quick Access
Java EE
Java
index.jsp
ProductServi...
Ecshop/pom.xml
product.jsp
ProductDao.java
ProductServiceImpl.java
Writable
Smart Insert
2 : 1

```java
package com.mapeak.ecshop.service;

import java.util.List;

import com.mapeak.ecshop.domain.Product;

public interface ProductService {

    //查询所有商品
    public List<Product> findAll();
}
```

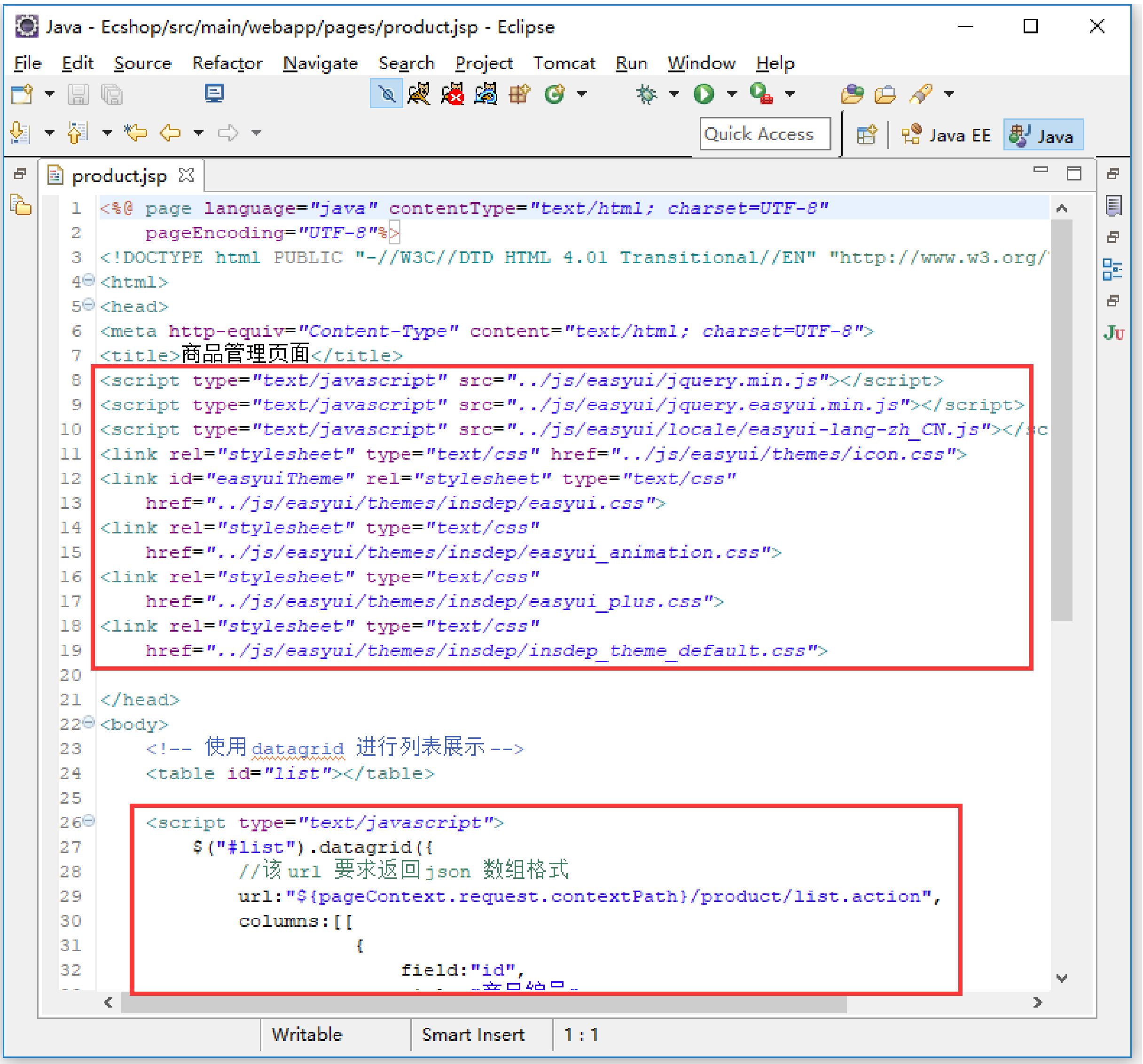
```java
package com.mapeak.ecshop.service.impl;

import java.util.List;

@Service
@Transactional //添加事务管理
public class ProductServiceImpl implements ProductService {

    //注入 DAO 对象
    @Resource
    private ProductDao dao;

    @Override
    public List<Product> findAll() {
        return dao.findAll();
    }
```

action:



```java
package com.mapeak.ecshop.web.action;

import java.io.IOException;

@Controller
@Scope("prototype")
@ParentPackage("struts-default")
@Namespace("/product")
public class ProductAction extends ActionSupport {

    //注入 Service 对象
    @Resource
    private ProductService productService;

    //查询所有商品
    @Action("list")
    public String list() throws Exception{
        //查询数据
        List<Product> list = productService.findAll();
        //把 List 集合转换为 json 数组
        return writeJson(list);
    }

    //专门把 Java 对象转换为 JSON 字符串
    private String writeJson(Object obj) throws IOException {
        //使用 fastJSON 工具
        String json = JSON.toJSONString(obj);
        HttpServletResponse resp = ServletActionContext.getResponse();
        resp.setContentType("text/json;charset=utf-8");
        resp.getWriter().write(json);
        return NONE;
    }
```

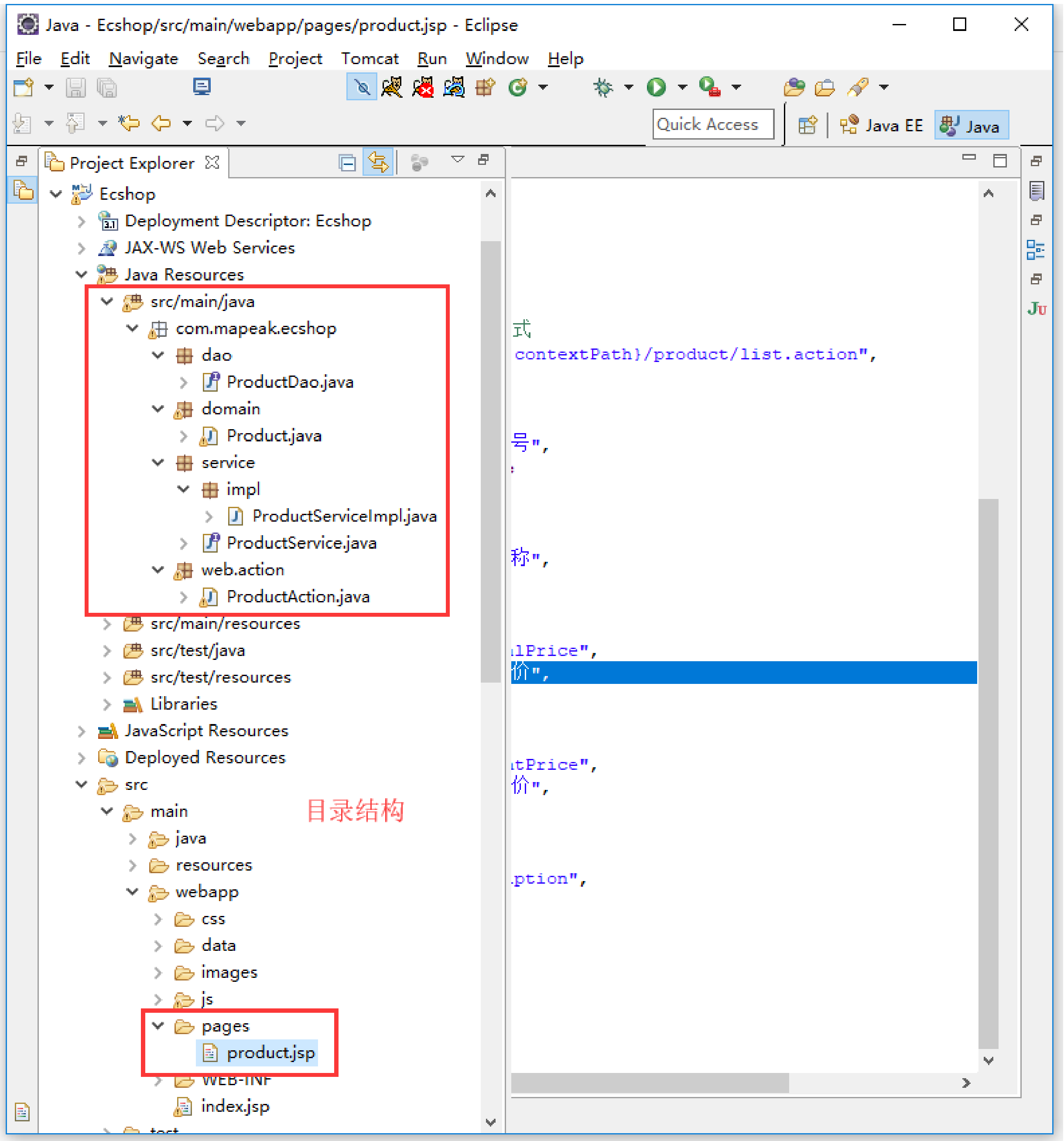
UI:


Java - Ecshop/src/main/webapp/pages/product.jsp - Eclipse
File Edit Source Refactor Navigate Search Project Tomcat Run Window Help
Quick Access
Java EE
Java
product.jsp
Writable
Smart Insert
1 : 1

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>商品管理页面</title>
<script type="text/javascript" src="../js/easyui/jquery.min.js"></script>
<script type="text/javascript" src="../js/easyui/jquery.easyui.min.js"></script>
<script type="text/javascript" src="../js/easyui/locale/easyui-lang-zh_CN.js"></sc
<link rel="stylesheet" type="text/css" href="../js/easyui/themes/icon.css">
<link id="easyuiTheme" rel="stylesheet" type="text/css"
    href="../js/easyui/themes/insdep/easyui.css">
<link rel="stylesheet" type="text/css"
    href="../js/easyui/themes/insdep/easyui_animation.css">
<link rel="stylesheet" type="text/css"
    href="../js/easyui/themes/insdep/easyui_plus.css">
<link rel="stylesheet" type="text/css"
    href="../js/easyui/themes/insdep/insdep_theme_default.css">

</head>
<body>
    <!-- 使用datagrid 进行列表展示 -->
    <table id="list"></table>

    <script type="text/javascript">
        $("#list").datagrid({
            //该 url 要求返回json 数组格式
            url:"${pageContext.request.contextPath}/product/list.action",
            columns:[[
                    {
                        field:"id",
```

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
   pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>商品管理页面</title>
<script type="text/javascript" src="../js/easyui/jquery.min.js"></script>
<script type="text/javascript" src="../js/easyui/jquery.easyui.min.js"></script>
<script type="text/javascript" src="../js/easyui/locale/easyui-lang-zh_CN.js"></script>
<link rel="stylesheet" type="text/css" href="../js/easyui/themes/icon.css">
<link id="easyuiTheme" rel="stylesheet" type="text/css"
   href="../js/easyui/themes/insdep/easyui.css">
<link rel="stylesheet" type="text/css"
   href="../js/easyui/themes/insdep/easyui_animation.css">
<link rel="stylesheet" type="text/css"
   href="../js/easyui/themes/insdep/easyui_plus.css">
<link rel="stylesheet" type="text/css"
   href="../js/easyui/themes/insdep/insdep_theme_default.css">

</head>
<body>
   <!-- 使用 datagrid 进行列表展示 -->
   <table id="list"></table>

   <script type="text/javascript">
      $("#list").datagrid({
         //该 url 要求返回 json 数组格式
         url:"${pageContext.request.contextPath}/product/list.action",
         columns:[[
               {
```

```
            field:"id",
            title:"商品编号",
            checkbox:true
          },
          {
            field:"name",
            title:"商品名称",
            width:200
          },
          {
            field:"orignalPrice",
            title:"商品原价",
            width:120
          },
          {
            field:"currentPrice",
            title:"商品现价",
            width:120
          },
          {
            field:"description",
            title:"描述",
            width:200
          }
        ]]
    });
  </script>
</body>
</html>
```

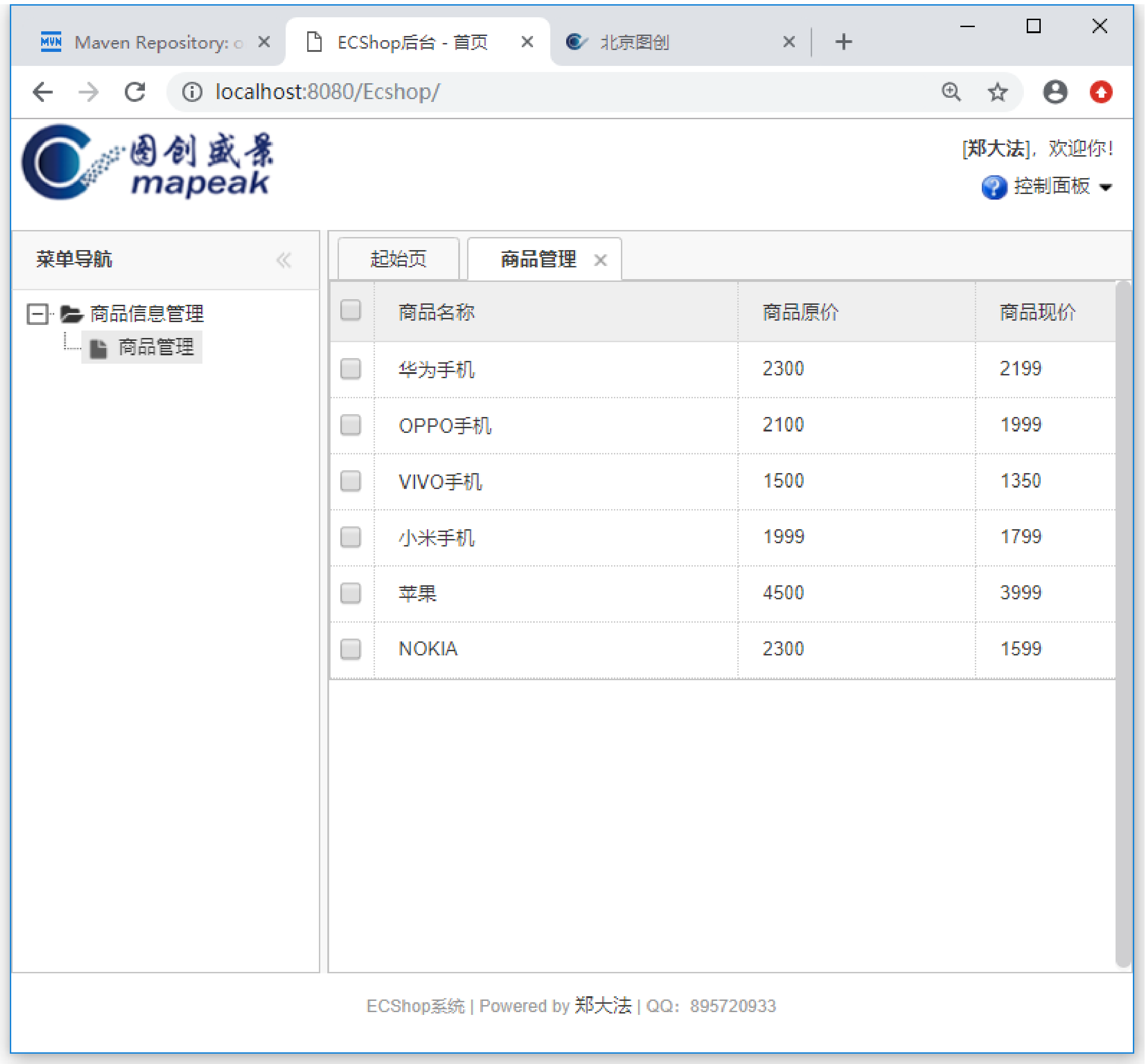

Java - Ecshop/src/main/webapp/pages/product.jsp - Eclipse

```
Ecshop
  Deployment Descriptor: Ecshop
  JAX-WS Web Services
  Java Resources
    src/main/java
      com.mapeak.ecshop
        dao
          ProductDao.java
        domain
          Product.java
        service
          impl
            ProductServiceImpl.java
          ProductService.java
        web.action
          ProductAction.java
    src/main/resources
    src/test/java
    src/test/resources
    Libraries
  JavaScript Resources
  Deployed Resources
  src
    main
      java
      resources
      webapp
        css
        data
        images
        js
        pages
          product.jsp
        WEB-INF
        index.jsp
    test
```

目录结构

```
式
contextPath}/product/list.action",
号",
:
称",
lPrice",
价",
tPrice",
价",
ption",
```

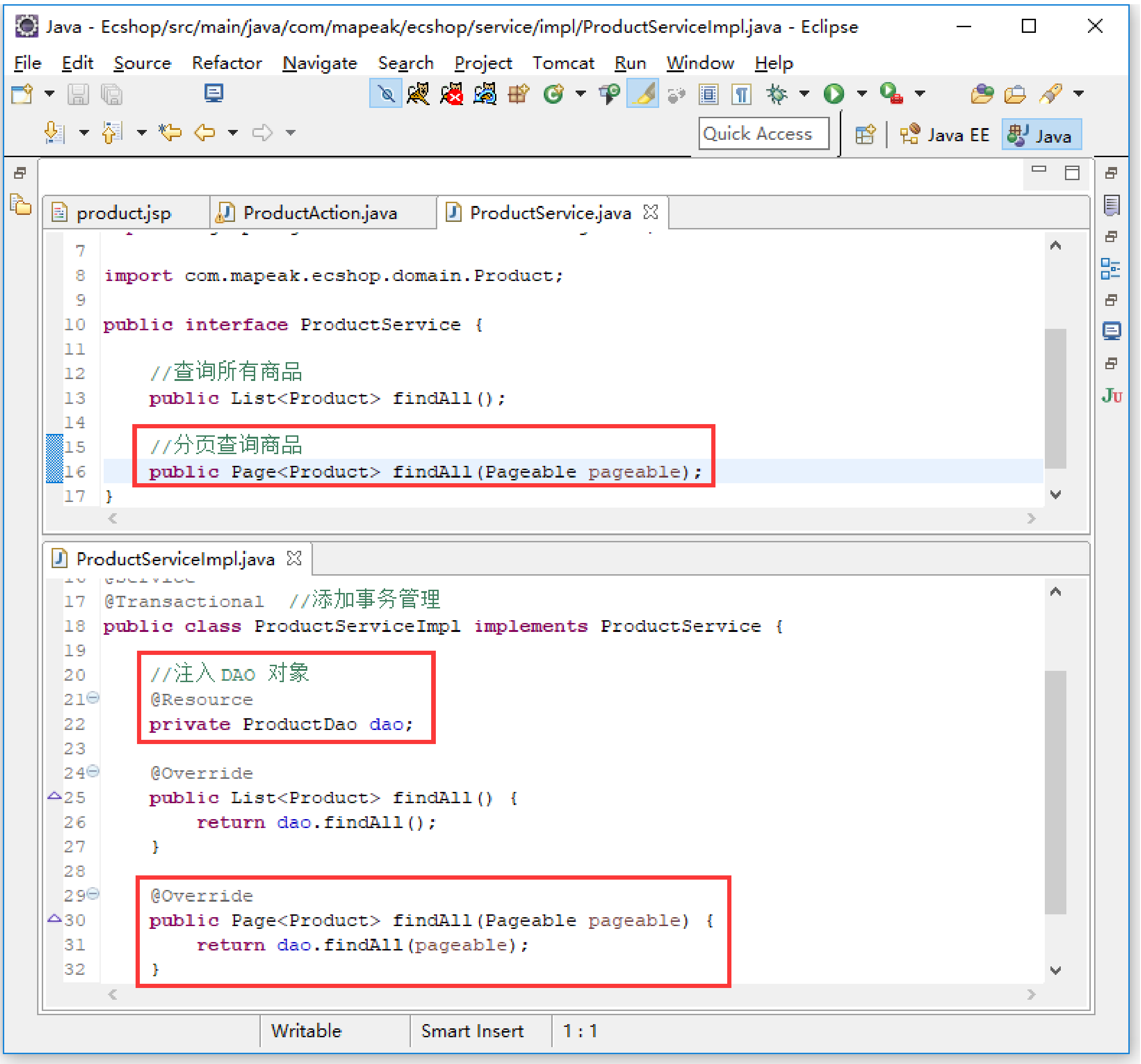
访问:



□	商品名称	商品原价	商品现价
□	华为手机	2300	2199
□	OPPO手机	2100	1999
□	VIVO手机	1500	1350
□	小米手机	1999	1799
□	苹果	4500	3999
□	NOKIA	2300	1599

ECShop系统 | Powered by 郑大法 | QQ：895720933

4、Spring Data JPA-分页查询商品：
service：


Java - Ecshop/src/main/java/com/mapeak/ecshop/service/impl/ProductServiceImpl.java - Eclipse
File Edit Source Refactor Navigate Search Project Tomcat Run Window Help
Quick Access
Java EE
Java
product.jsp
ProductAction.java
ProductService.java
ProductServiceImpl.java
Writable
Smart Insert
1 : 1

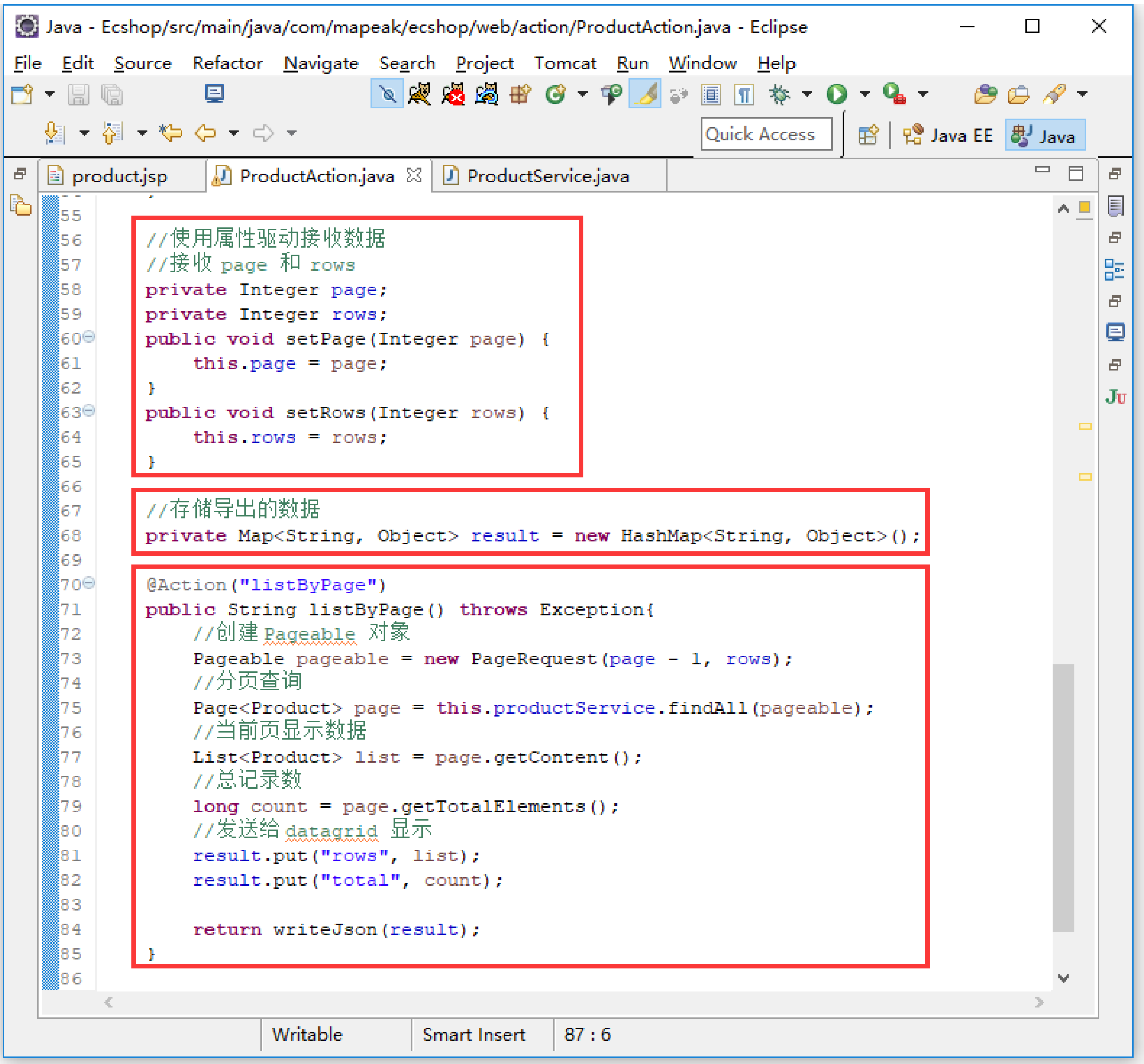
ProductService.java:

```java
import com.mapeak.ecshop.domain.Product;

public interface ProductService {

    //查询所有商品
    public List<Product> findAll();

    //分页查询商品
    public Page<Product> findAll(Pageable pageable);
}
```

ProductServiceImpl.java:

```java
@Service
@Transactional  //添加事务管理
public class ProductServiceImpl implements ProductService {

    //注入 DAO 对象
    @Resource
    private ProductDao dao;

    @Override
    public List<Product> findAll() {
        return dao.findAll();
    }

    @Override
    public Page<Product> findAll(Pageable pageable) {
        return dao.findAll(pageable);
    }
```

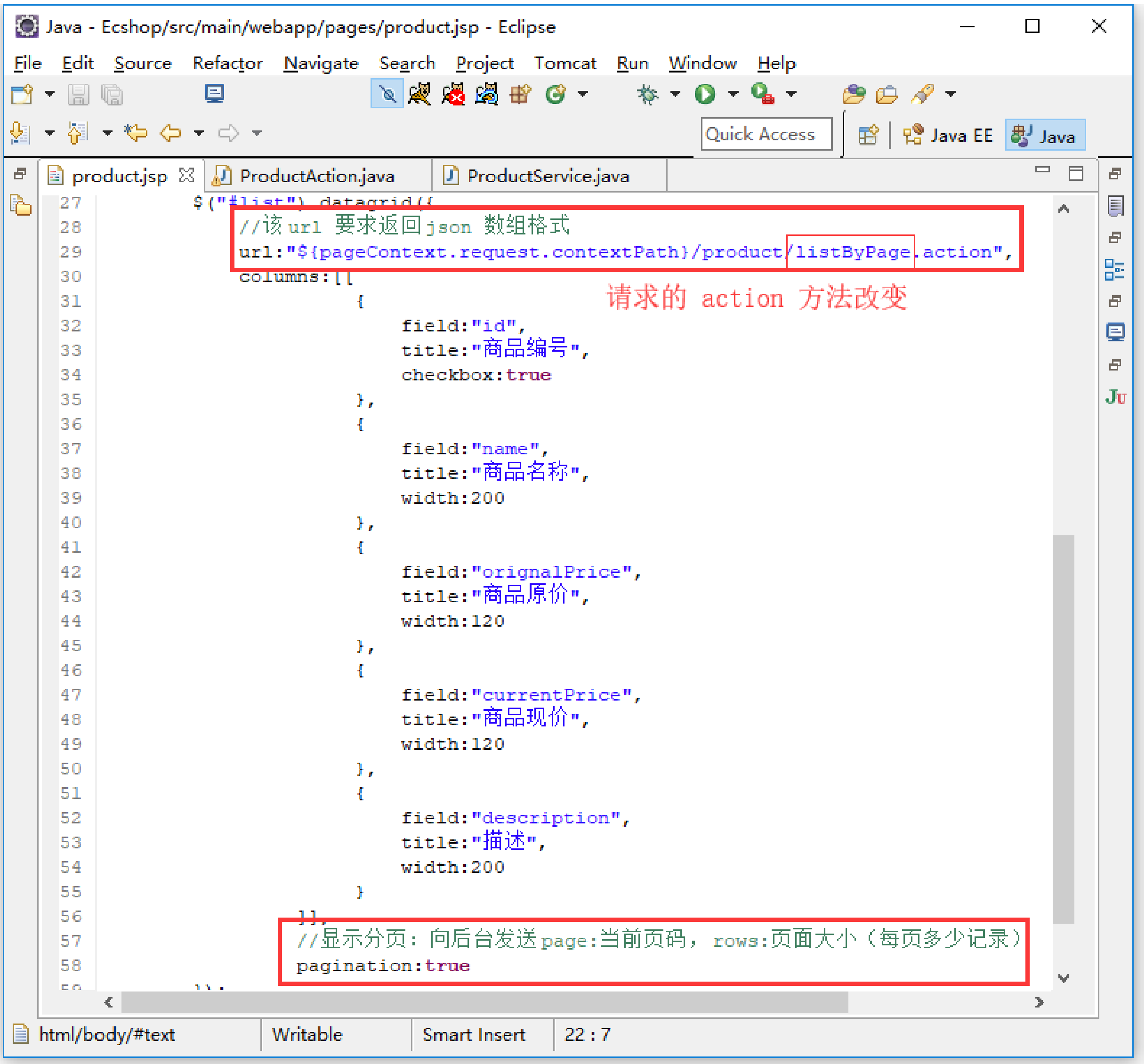
action:



Java - Ecshop/src/main/java/com/mapeak/ecshop/web/action/ProductAction.java - Eclipse
File Edit Source Refactor Navigate Search Project Tomcat Run Window Help
Quick Access
Java EE
Java
product.jsp
ProductAction.java
ProductService.java
Writable
Smart Insert
87 : 6

```java
//使用属性驱动接收数据
//接收 page 和 rows
private Integer page;
private Integer rows;
public void setPage(Integer page) {
    this.page = page;
}
public void setRows(Integer rows) {
    this.rows = rows;
}

//存储导出的数据
private Map<String, Object> result = new HashMap<String, Object>();

@Action("listByPage")
public String listByPage() throws Exception{
    //创建 Pageable 对象
    Pageable pageable = new PageRequest(page - 1, rows);
    //分页查询
    Page<Product> page = this.productService.findAll(pageable);
    //当前页显示数据
    List<Product> list = page.getContent();
    //总记录数
    long count = page.getTotalElements();
    //发送给 datagrid 显示
    result.put("rows", list);
    result.put("total", count);

    return writeJson(result);
}
```

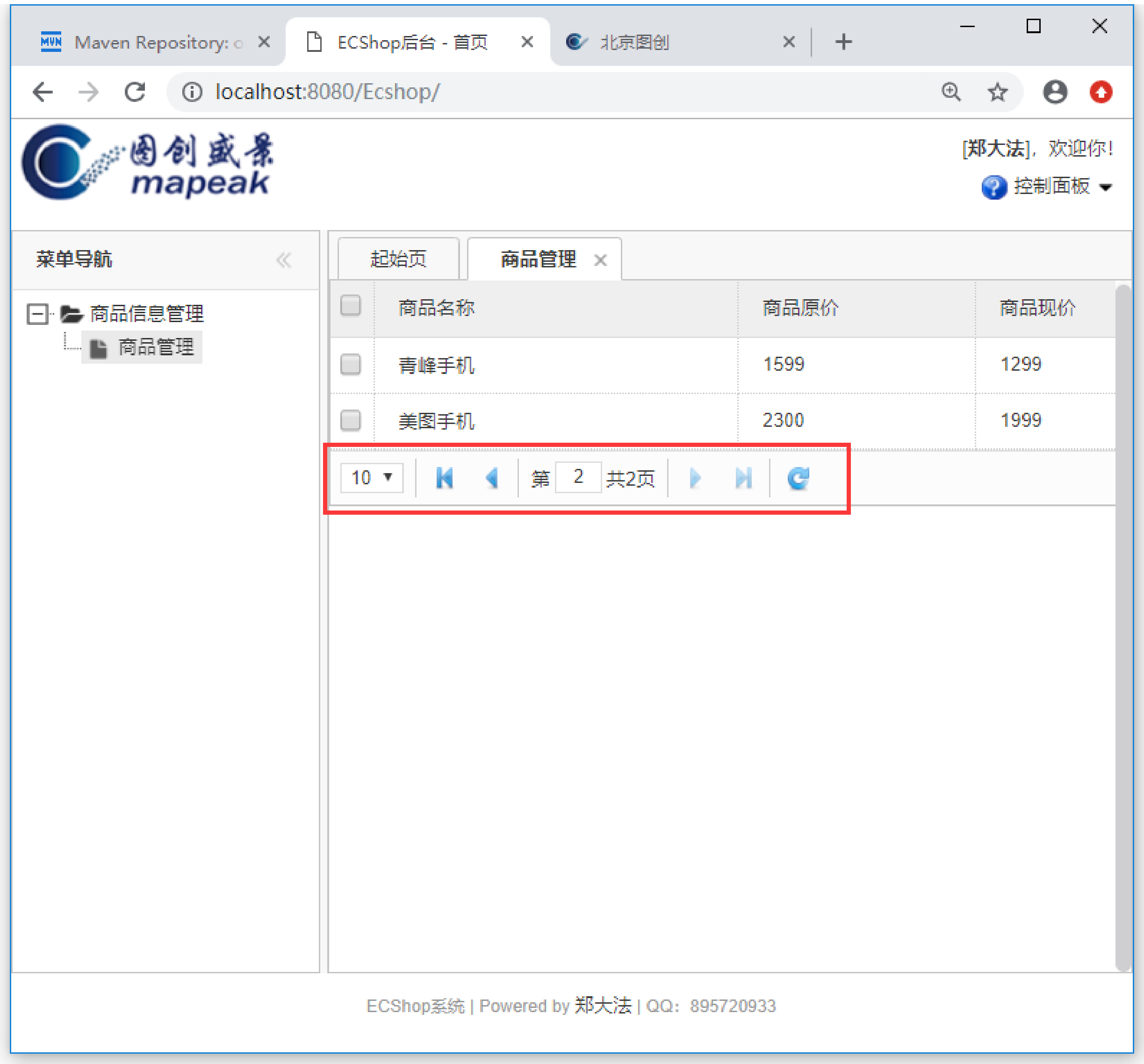
UI:


Java - Ecshop/src/main/webapp/pages/product.jsp - Eclipse
File Edit Source Refactor Navigate Search Project Tomcat Run Window Help
Quick Access
Java EE
Java
product.jsp
ProductAction.java
ProductService.java
html/body/#text
Writable
Smart Insert
22 : 7

```
$("#list").datagrid({
    //该 url 要求返回 json 数组格式
    url:"${pageContext.request.contextPath}/product/listByPage.action",
    columns:[[
              {
                  field:"id",
                  title:"商品编号",
                  checkbox:true
              },
              {
                  field:"name",
                  title:"商品名称",
                  width:200
              },
              {
                  field:"orignalPrice",
                  title:"商品原价",
                  width:120
              },
              {
                  field:"currentPrice",
                  title:"商品现价",
                  width:120
              },
              {
                  field:"description",
                  title:"描述",
                  width:200
              }
         ]],
         //显示分页：向后台发送page:当前页码，rows:页面大小（每页多少记录）
         pagination:true
});
```

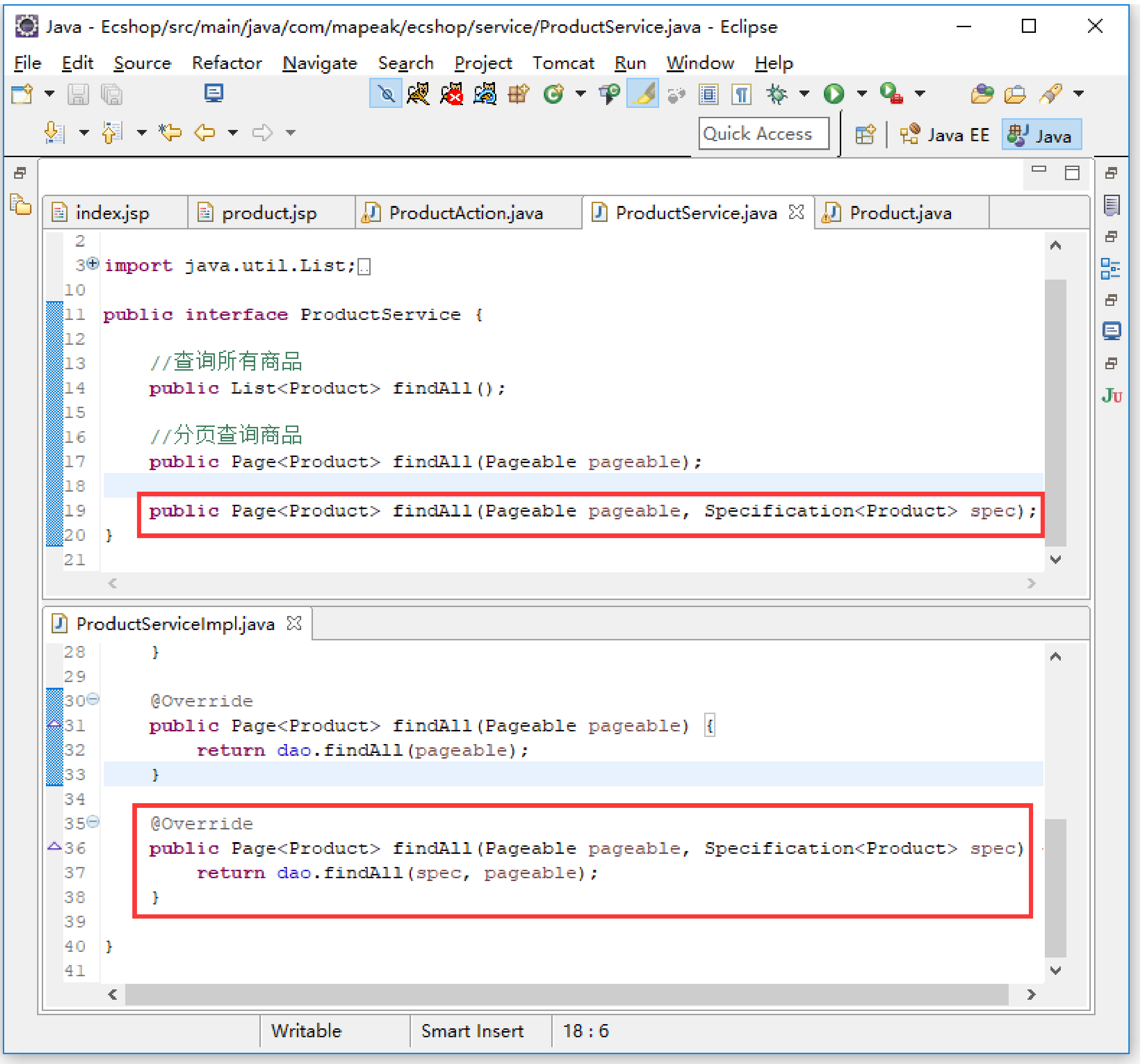
请求的 action 方法改变

访问：



□	商品名称	商品原价	商品现价
□	青峰手机	1599	1299
□	美图手机	2300	1999

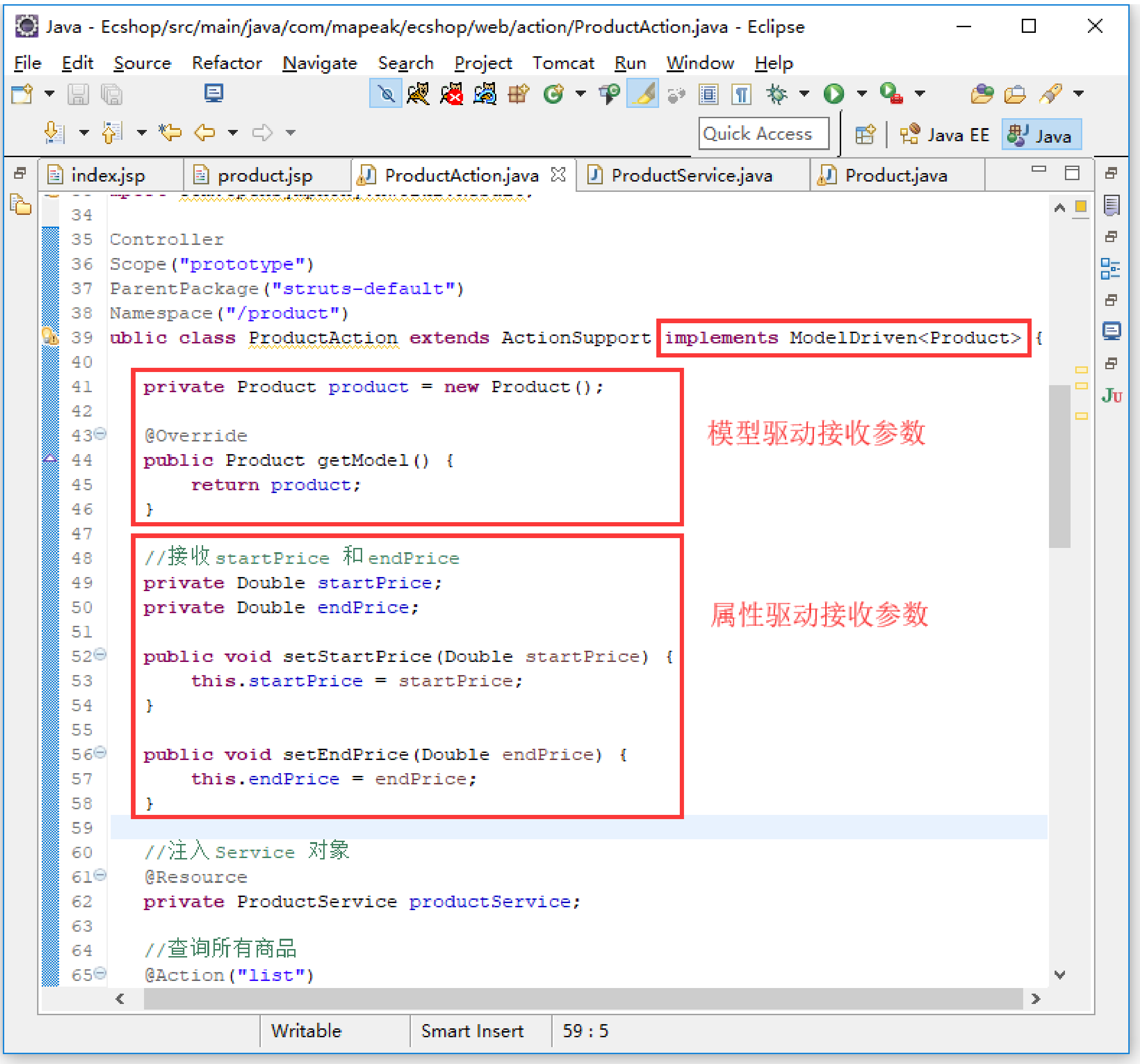
5、Spring Data JPA-组合条件查询商品：
service：


Java - Ecshop/src/main/java/com/mapeak/ecshop/service/ProductService.java - Eclipse
File Edit Source Refactor Navigate Search Project Tomcat Run Window Help
Quick Access
Java EE
Java
index.jsp
product.jsp
ProductAction.java
ProductService.java
Product.java
ProductServiceImpl.java
Writable
Smart Insert
18 : 6

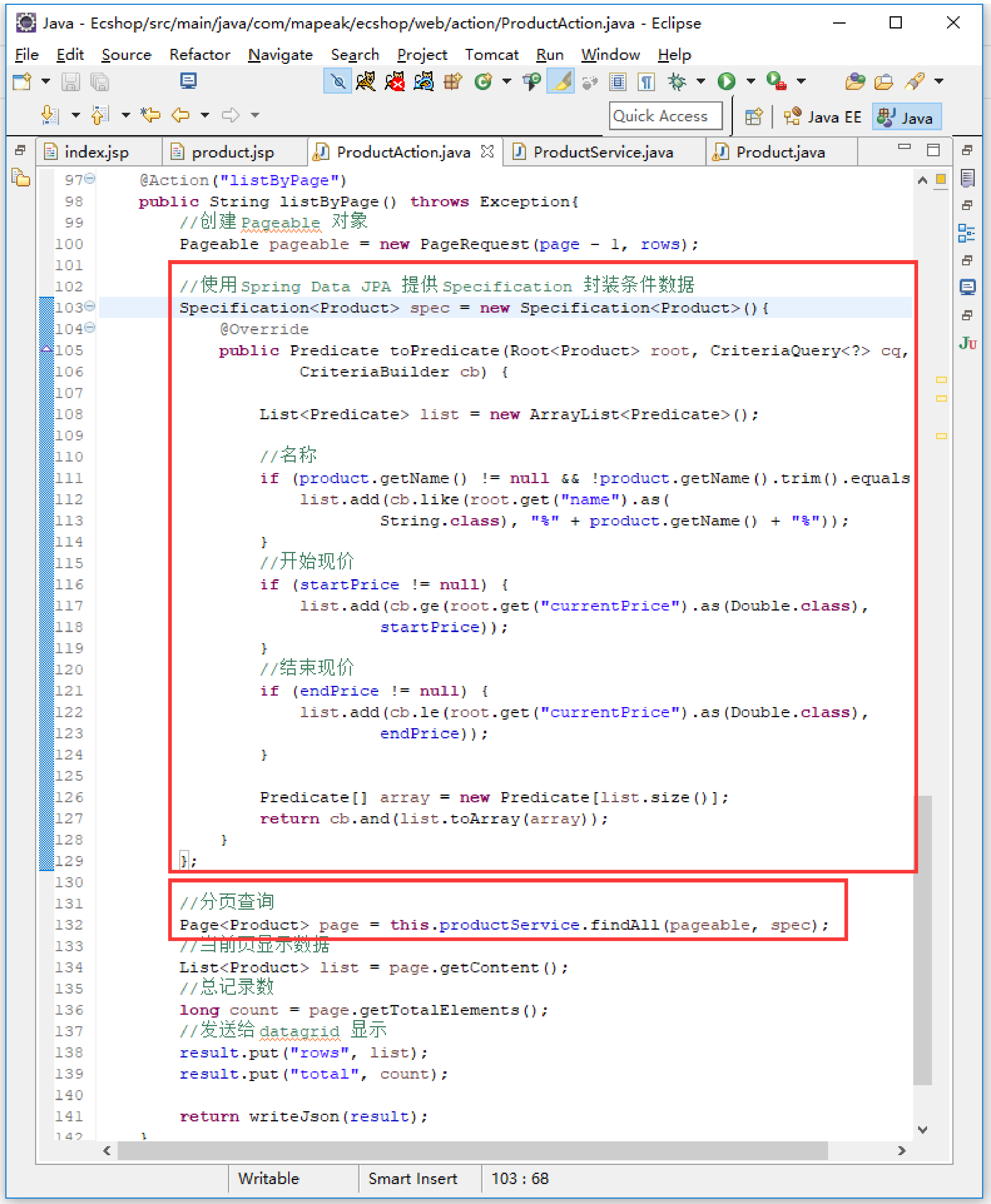
```java

import java.util.List;

public interface ProductService {

    //查询所有商品
    public List<Product> findAll();

    //分页查询商品
    public Page<Product> findAll(Pageable pageable);

    public Page<Product> findAll(Pageable pageable, Specification<Product> spec);
}

```

```java
    }

    @Override
    public Page<Product> findAll(Pageable pageable) {
        return dao.findAll(pageable);
    }

    @Override
    public Page<Product> findAll(Pageable pageable, Specification<Product> spec)
        return dao.findAll(spec, pageable);
    }

}

```

action:

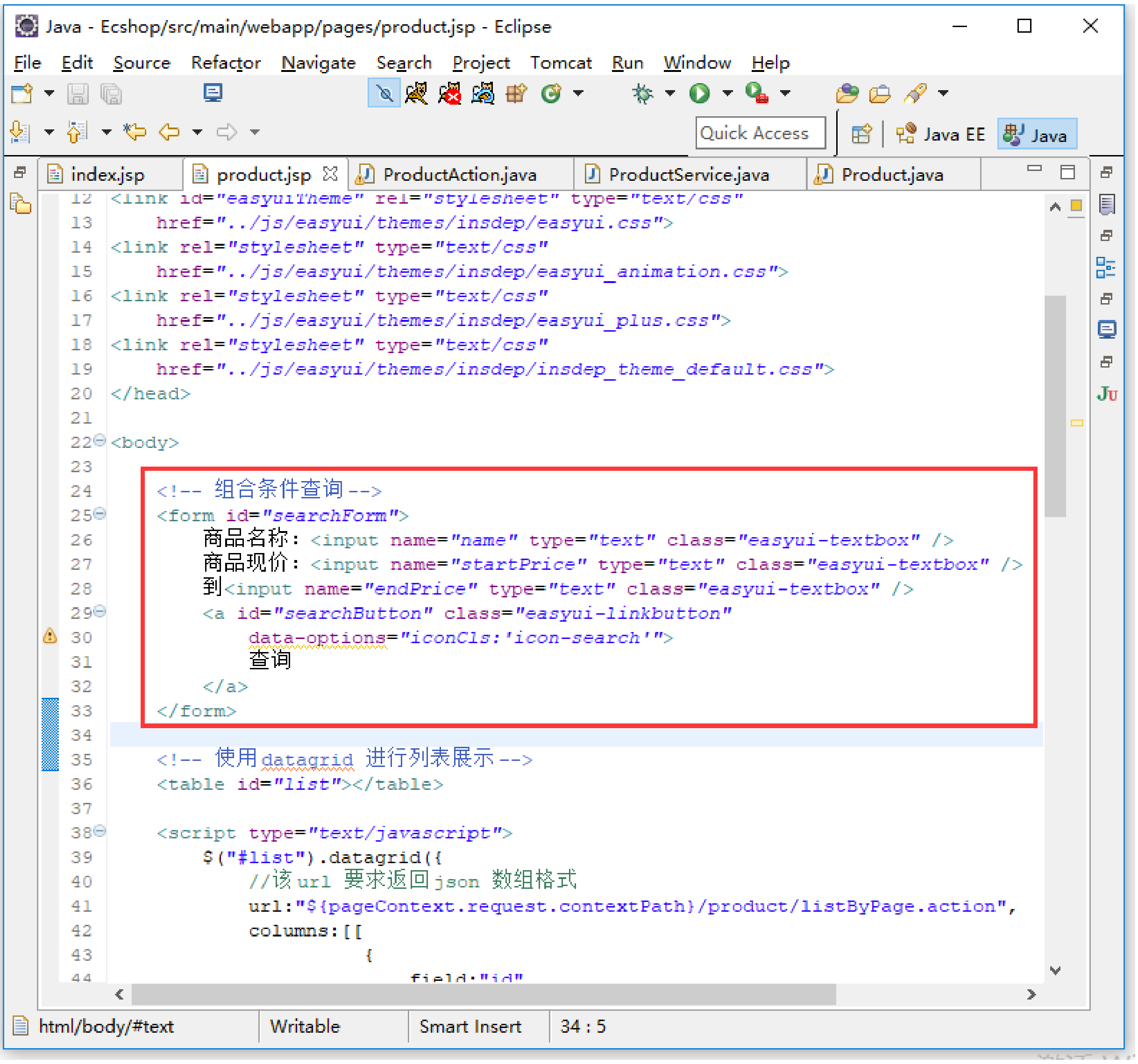

Java - Ecshop/src/main/java/com/mapeak/ecshop/web/action/ProductAction.java - Eclipse
File Edit Source Refactor Navigate Search Project Tomcat Run Window Help
Quick Access
Java EE
Java
index.jsp
product.jsp
ProductAction.java
ProductService.java
Product.java

```
Controller
Scope("prototype")
ParentPackage("struts-default")
Namespace("/product")
ublic class ProductAction extends ActionSupport implements ModelDriven<Product> {

    private Product product = new Product();

    @Override
    public Product getModel() {
        return product;
    }

    //接收 startPrice 和 endPrice
    private Double startPrice;
    private Double endPrice;

    public void setStartPrice(Double startPrice) {
        this.startPrice = startPrice;
    }

    public void setEndPrice(Double endPrice) {
        this.endPrice = endPrice;
    }

    //注入 Service 对象
    @Resource
    private ProductService productService;

    //查询所有商品
    @Action("list")
```

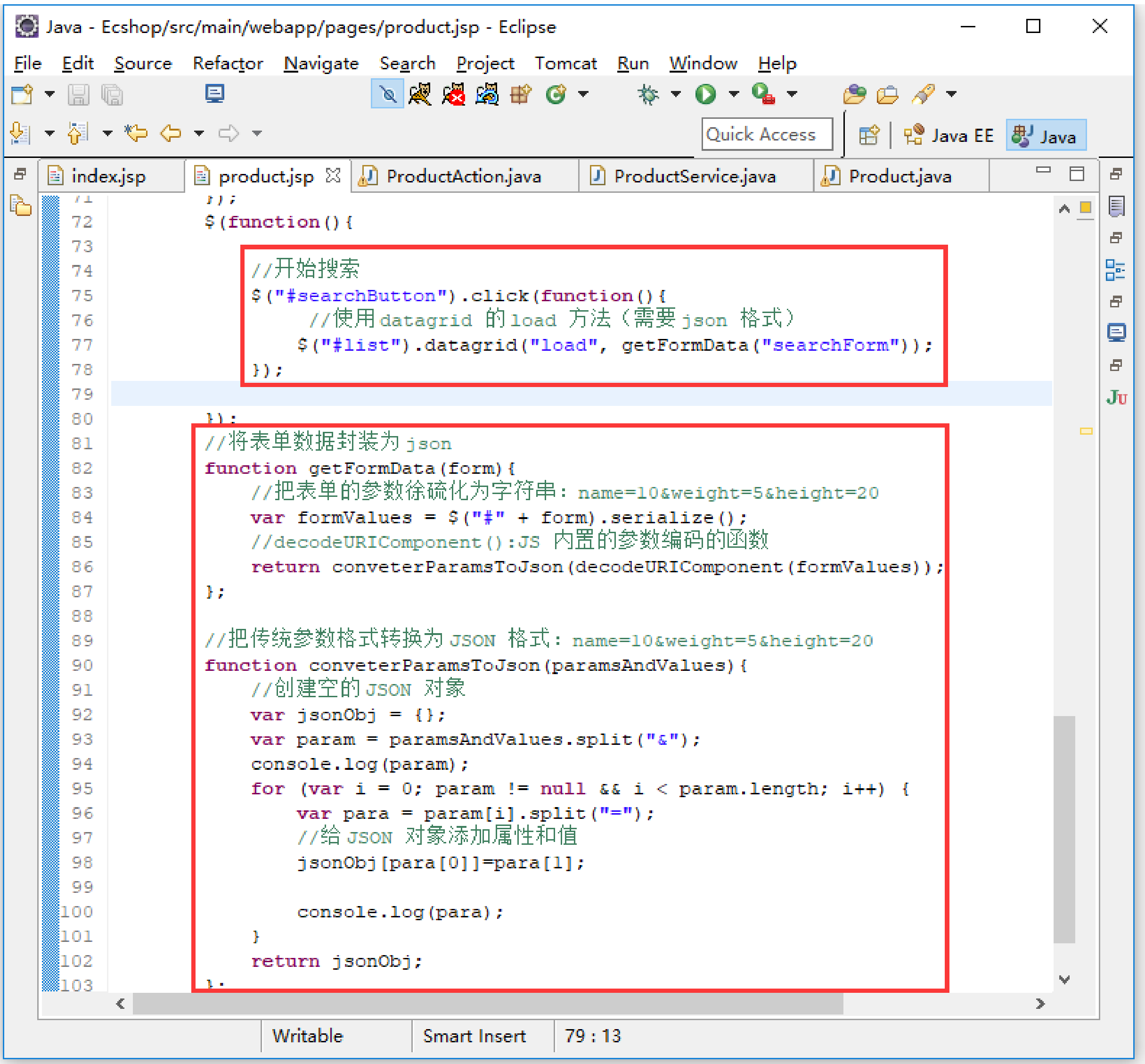


```
    @Action("listByPage")
    public String listByPage() throws Exception{
        //创建 Pageable 对象
        Pageable pageable = new PageRequest(page - 1, rows);

        //使用Spring Data JPA 提供Specification 封装条件数据
        Specification<Product> spec = new Specification<Product>(){
            @Override
            public Predicate toPredicate(Root<Product> root, CriteriaQuery<?> cq,
                    CriteriaBuilder cb) {

                List<Predicate> list = new ArrayList<Predicate>();

                //名称
                if (product.getName() != null && !product.getName().trim().equals
                    list.add(cb.like(root.get("name").as(
                            String.class), "%" + product.getName() + "%"));
                }
                //开始现价
                if (startPrice != null) {
                    list.add(cb.ge(root.get("currentPrice").as(Double.class),
                            startPrice));
                }
                //结束现价
                if (endPrice != null) {
                    list.add(cb.le(root.get("currentPrice").as(Double.class),
                            endPrice));
                }

                Predicate[] array = new Predicate[list.size()];
                return cb.and(list.toArray(array));
            }
        };

        //分页查询
        Page<Product> page = this.productService.findAll(pageable, spec);
        //当前页显示数据
        List<Product> list = page.getContent();
        //总记录数
        long count = page.getTotalElements();
        //发送给 datagrid 显示
        result.put("rows", list);
        result.put("total", count);

        return writeJson(result);
    }
```

UI:

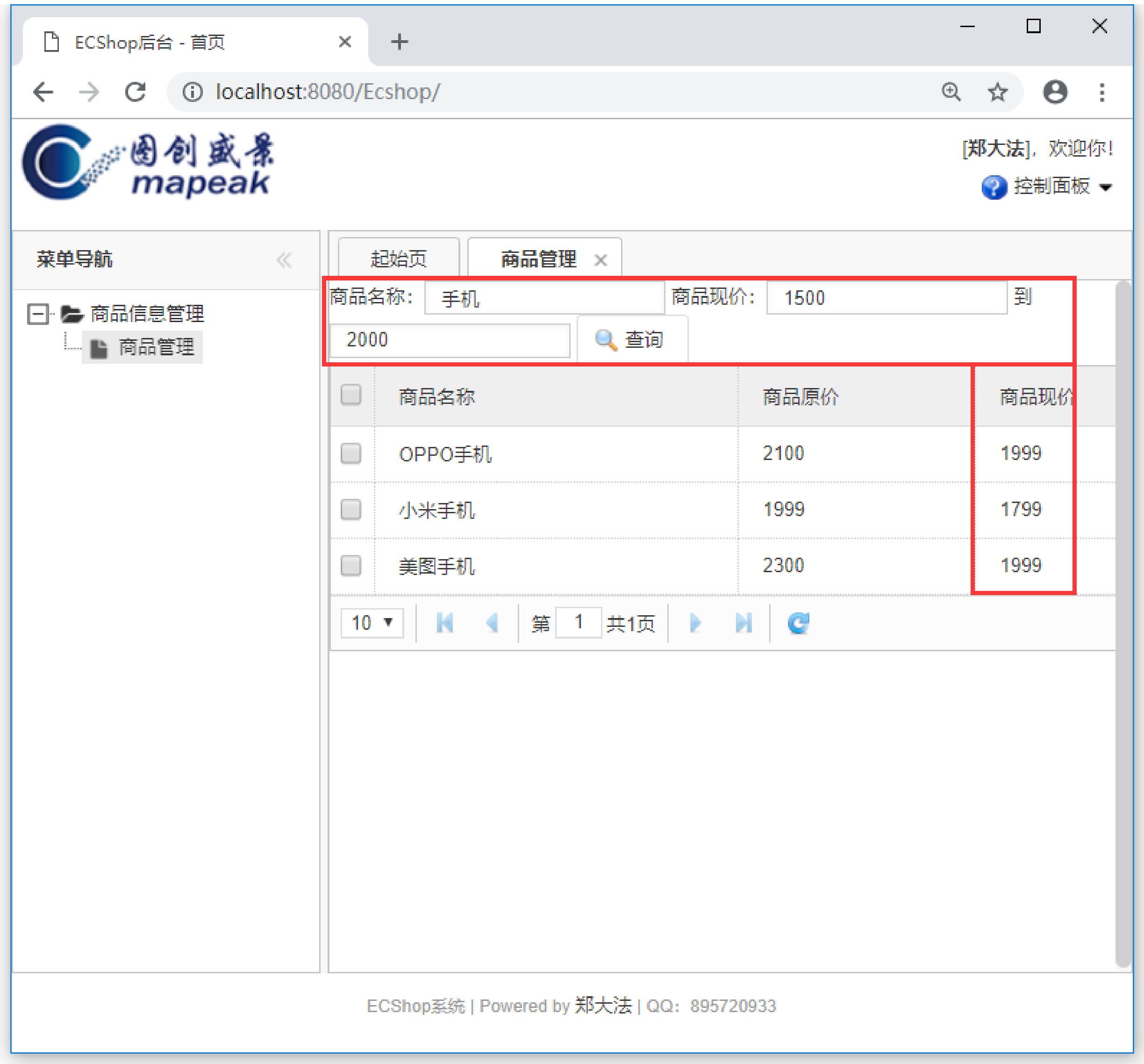


```
<link id="easyuiTheme" rel="stylesheet" type="text/css"
    href="../js/easyui/themes/insdep/easyui.css">
<link rel="stylesheet" type="text/css"
    href="../js/easyui/themes/insdep/easyui_animation.css">
<link rel="stylesheet" type="text/css"
    href="../js/easyui/themes/insdep/easyui_plus.css">
<link rel="stylesheet" type="text/css"
    href="../js/easyui/themes/insdep/insdep_theme_default.css">
</head>

<body>

    <!-- 组合条件查询 -->
    <form id="searchForm">
        商品名称：<input name="name" type="text" class="easyui-textbox" />
        商品现价：<input name="startPrice" type="text" class="easyui-textbox" />
        到<input name="endPrice" type="text" class="easyui-textbox" />
        <a id="searchButton" class="easyui-linkbutton"
            data-options="iconCls:'icon-search'">
            查询
        </a>
    </form>

    <!-- 使用datagrid 进行列表展示 -->
    <table id="list"></table>

    <script type="text/javascript">
        $("#list").datagrid({
            //该url 要求返回json 数组格式
            url:"${pageContext.request.contextPath}/product/listByPage.action",
            columns:[[
                      {
                          field:"id"
```


Java - Ecshop/src/main/webapp/pages/product.jsp - Eclipse
File Edit Source Refactor Navigate Search Project Tomcat Run Window Help
Quick Access
Java EE
Java
index.jsp
product.jsp
ProductAction.java
ProductService.java
Product.java
Writable
Smart Insert
79 : 13

```
});
$(function(){

    //开始搜索
    $("#searchButton").click(function(){
        //使用datagrid 的load 方法（需要json 格式）
        $("#list").datagrid("load", getFormData("searchForm"));
    });

});
//将表单数据封装为json
function getFormData(form){
    //把表单的参数徐疏化为字符串: name=10&weight=5&height=20
    var formValues = $("#" + form).serialize();
    //decodeURIComponent():JS 内置的参数编码的函数
    return conveterParamsToJson(decodeURIComponent(formValues));
};

//把传统参数格式转换为JSON 格式: name=10&weight=5&height=20
function conveterParamsToJson(paramsAndValues){
    //创建空的JSON 对象
    var jsonObj = {};
    var param = paramsAndValues.split("&");
    console.log(param);
    for (var i = 0; param != null && i < param.length; i++) {
        var para = param[i].split("=");
        //给JSON 对象添加属性和值
        jsonObj[para[0]]=para[1];

        console.log(para);
    }
    return jsonObj;
};
```

访问:



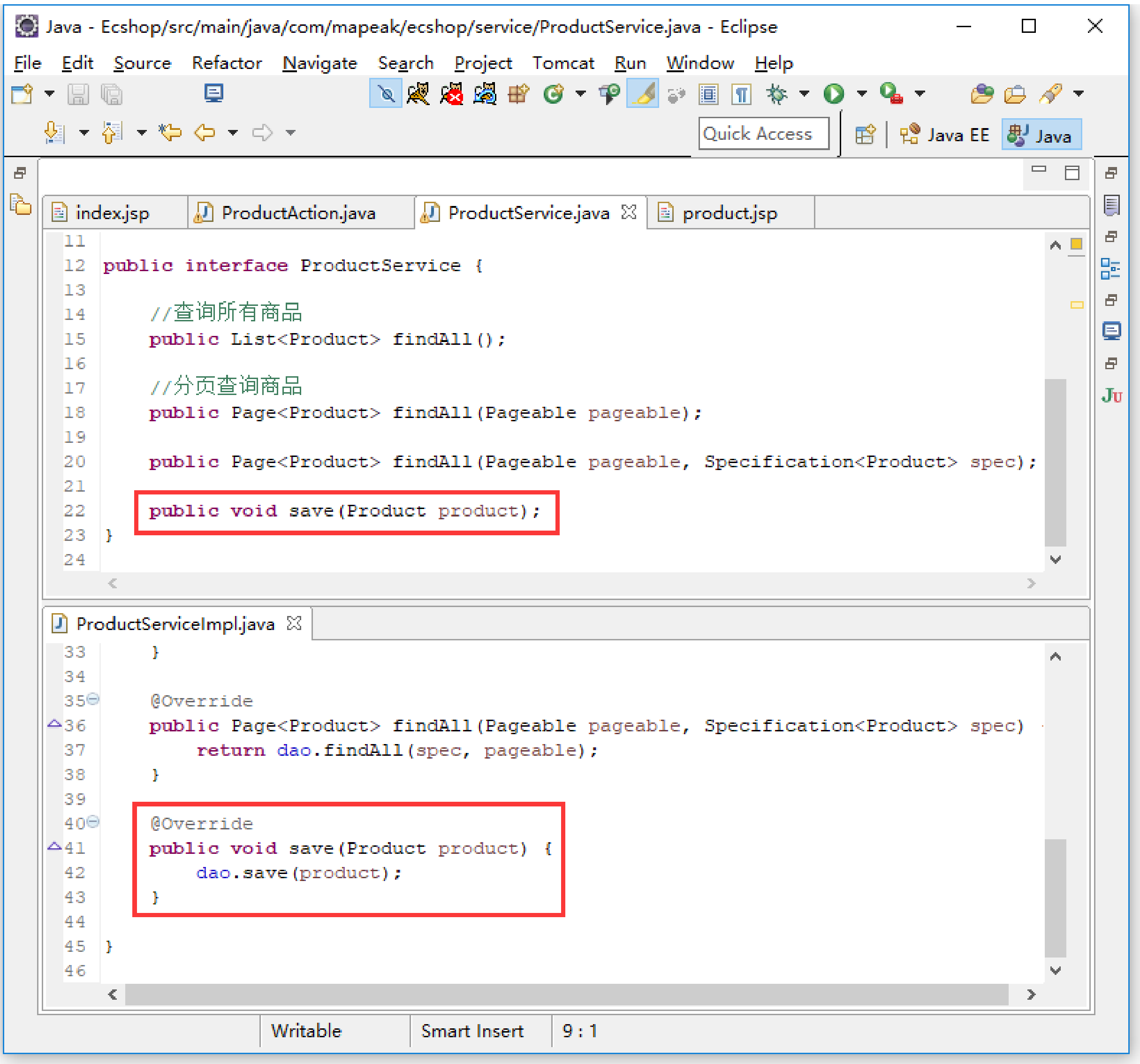
商品名称： 手机 商品现价： 1500 到 2000 查询

□	商品名称	商品原价	商品现价
□	OPPO手机	2100	1999
□	小米手机	1999	1799
□	美图手机	2300	1999

10 ▼ 第 1 共1页

ECShop系统 | Powered by 郑大法 | QQ：895720933

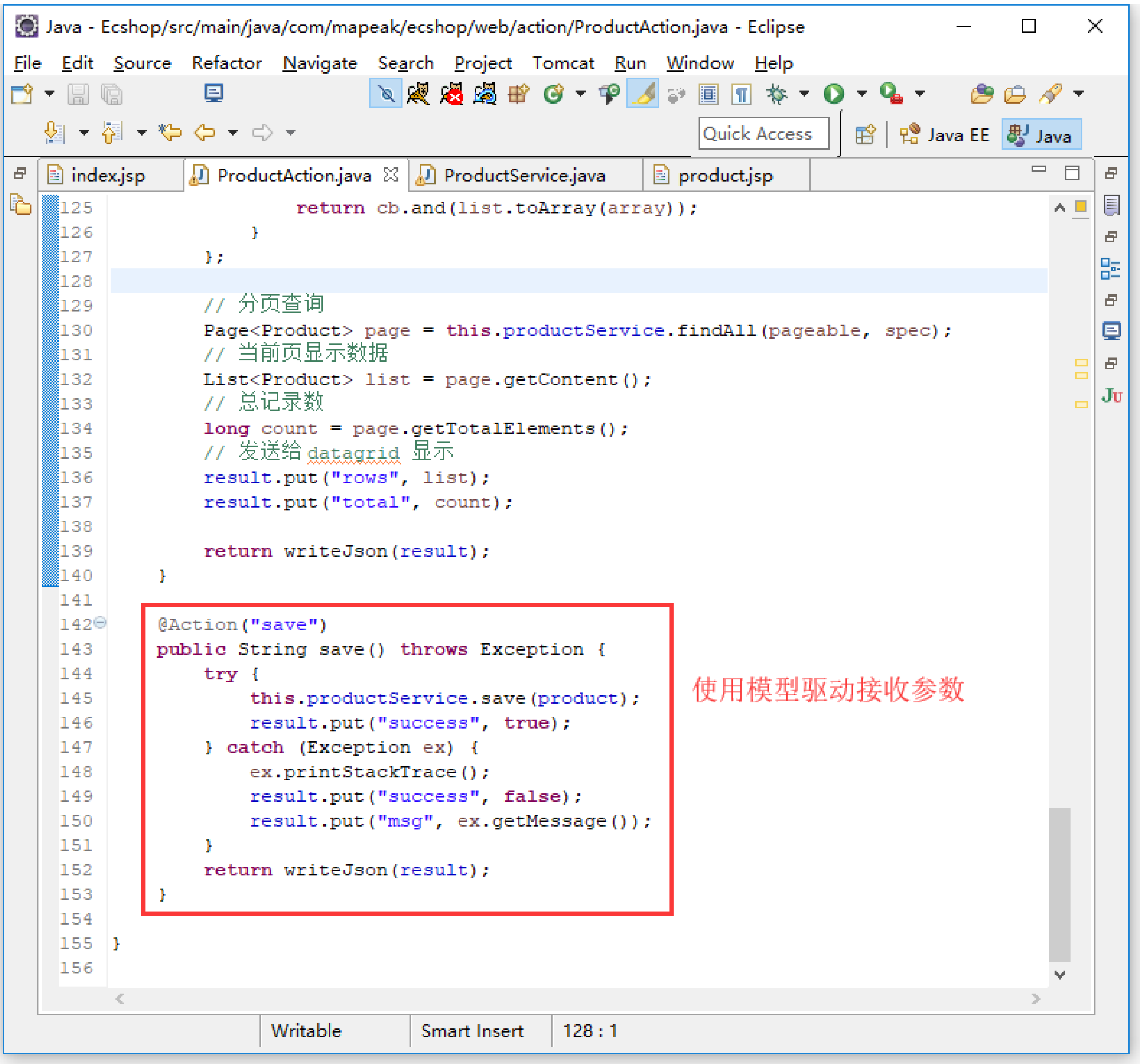
6、Spring Data JPA-商品添加：
service：



```java

public interface ProductService {

    //查询所有商品
    public List<Product> findAll();

    //分页查询商品
    public Page<Product> findAll(Pageable pageable);

    public Page<Product> findAll(Pageable pageable, Specification<Product> spec);

    public void save(Product product);
}

```

```java
    }

    @Override
    public Page<Product> findAll(Pageable pageable, Specification<Product> spec)
        return dao.findAll(spec, pageable);
    }

    @Override
    public void save(Product product) {
        dao.save(product);
    }

}

```

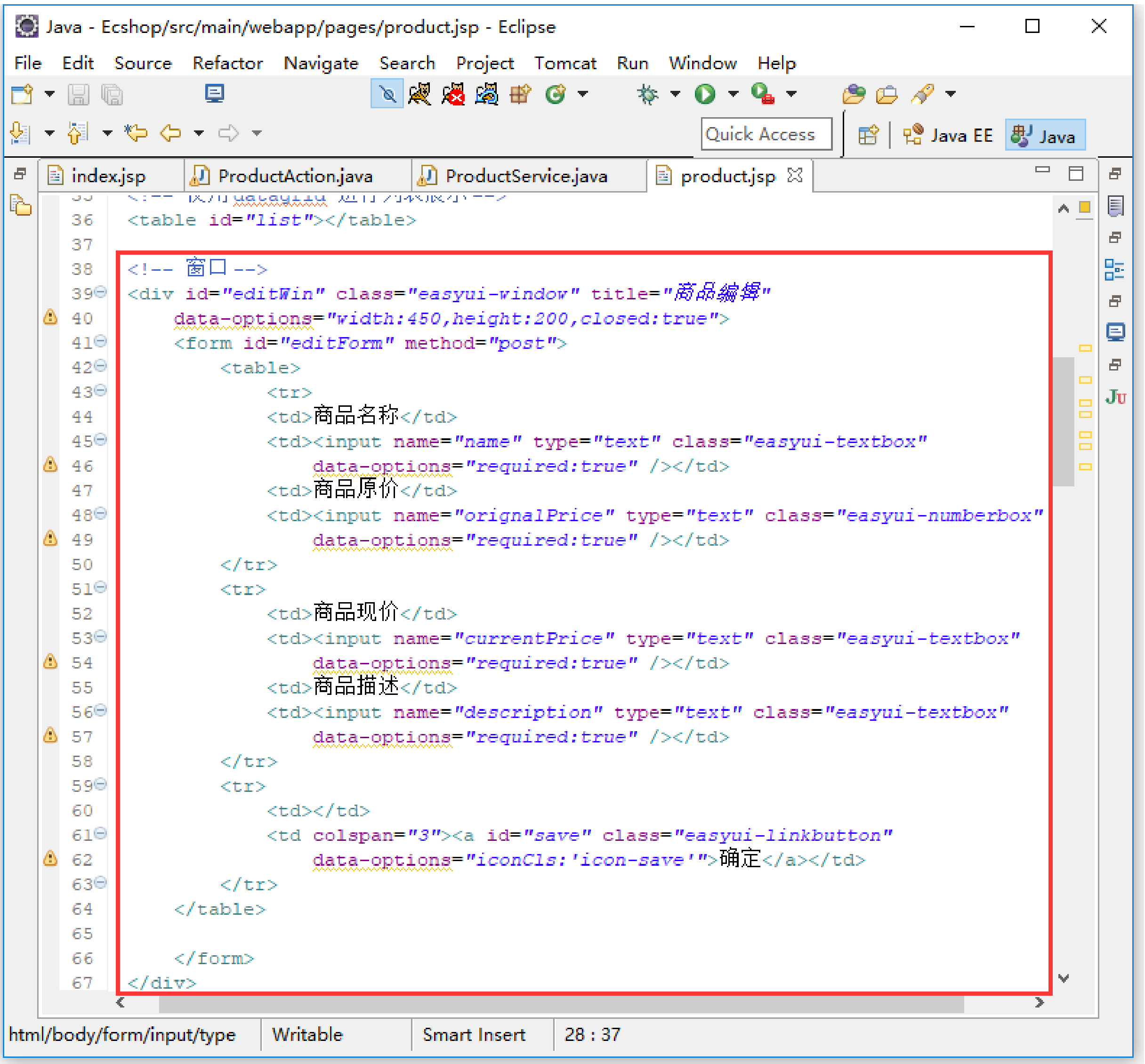
action:


Java - Ecshop/src/main/java/com/mapeak/ecshop/web/action/ProductAction.java - Eclipse
File Edit Source Refactor Navigate Search Project Tomcat Run Window Help
Quick Access
Java EE
Java
index.jsp
ProductAction.java
ProductService.java
product.jsp
Writable
Smart Insert
128 : 1

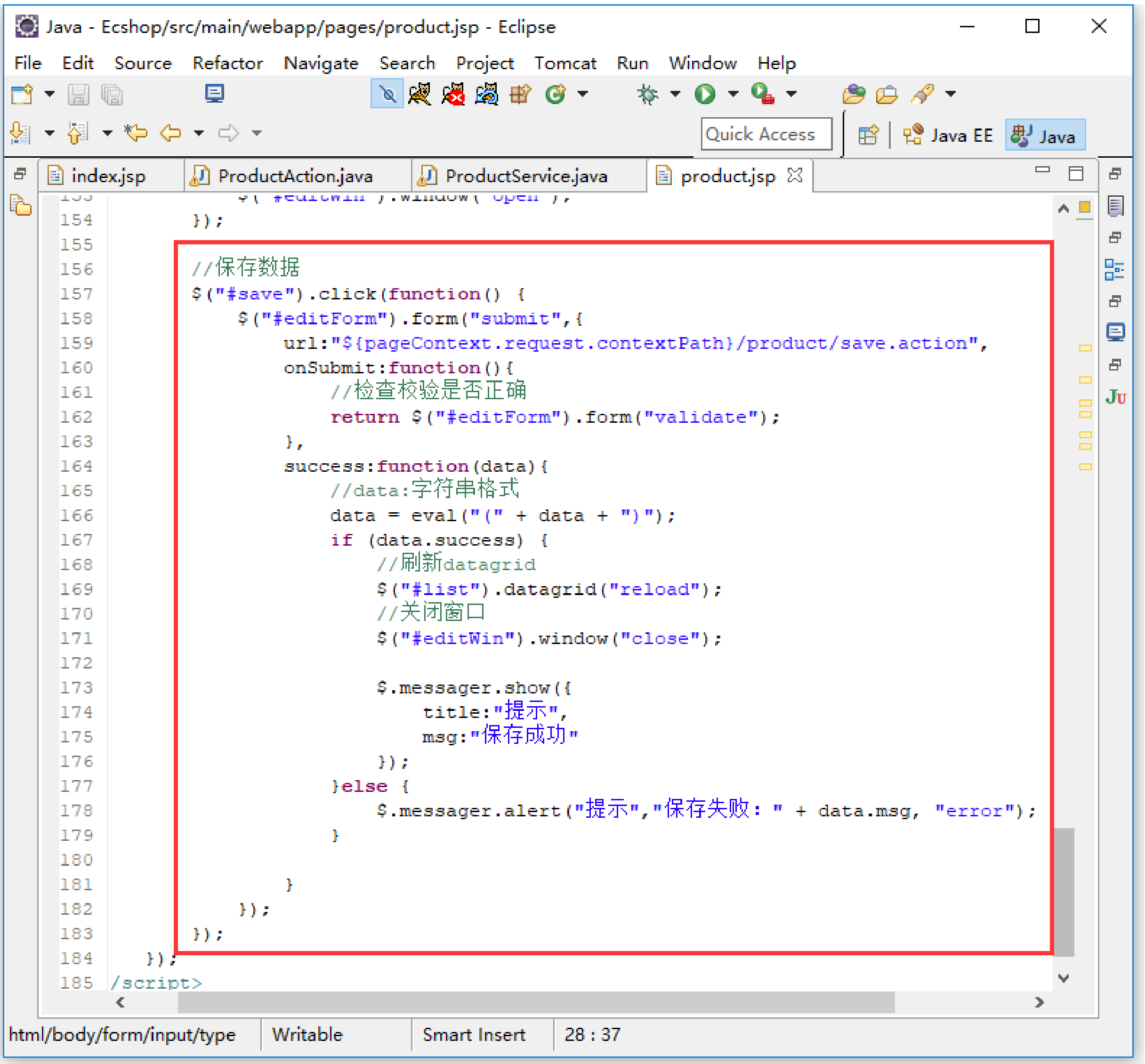
```
                return cb.and(list.toArray(array));
            }
        };

        // 分页查询
        Page<Product> page = this.productService.findAll(pageable, spec);
        // 当前页显示数据
        List<Product> list = page.getContent();
        // 总记录数
        long count = page.getTotalElements();
        // 发送给 datagrid 显示
        result.put("rows", list);
        result.put("total", count);

        return writeJson(result);
    }

    @Action("save")
    public String save() throws Exception {
        try {
            this.productService.save(product);
            result.put("success", true);
        } catch (Exception ex) {
            ex.printStackTrace();
            result.put("success", false);
            result.put("msg", ex.getMessage());
        }
        return writeJson(result);
    }

}
```

使用模型驱动接收参数

UI:


Java - Ecshop/src/main/webapp/pages/product.jsp - Eclipse
File Edit Source Refactor Navigate Search Project Tomcat Run Window Help
Quick Access
Java EE
Java
index.jsp
ProductAction.java
ProductService.java
product.jsp
html/body/form/input/type
Writable
Smart Insert
28 : 37

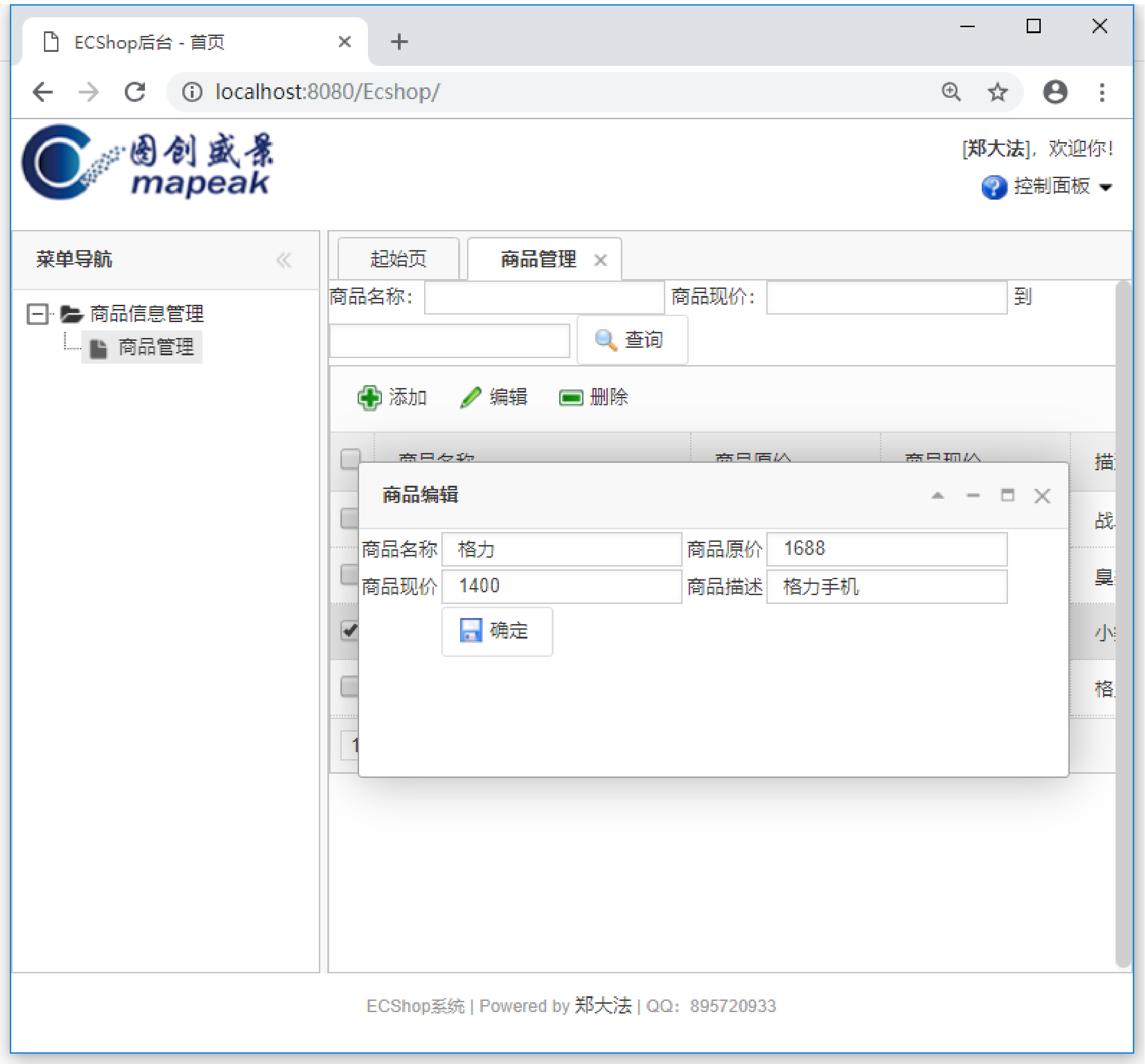
```
<!-- 使用datagrid 进行列表展示 -->
<table id="list"></table>

<!-- 窗口 -->
<div id="editWin" class="easyui-window" title="商品编辑"
    data-options="width:450,height:200,closed:true">
    <form id="editForm" method="post">
        <table>
            <tr>
                <td>商品名称</td>
                <td><input name="name" type="text" class="easyui-textbox"
                    data-options="required:true" /></td>
                <td>商品原价</td>
                <td><input name="orignalPrice" type="text" class="easyui-numberbox"
                    data-options="required:true" /></td>
            </tr>
            <tr>
                <td>商品现价</td>
                <td><input name="currentPrice" type="text" class="easyui-textbox"
                    data-options="required:true" /></td>
                <td>商品描述</td>
                <td><input name="description" type="text" class="easyui-textbox"
                    data-options="required:true" /></td>
            </tr>
            <tr>
                <td></td>
                <td colspan="3"><a id="save" class="easyui-linkbutton"
                    data-options="iconCls:'icon-save'">确定</a></td>
            </tr>
        </table>

    </form>
</div>
```



```
            $("#editWin").window("open");
        });

        //保存数据
        $("#save").click(function() {
            $("#editForm").form("submit",{
                url:"${pageContext.request.contextPath}/product/save.action",
                onSubmit:function(){
                    //检查校验是否正确
                    return $("#editForm").form("validate");
                },
                success:function(data){
                    //data:字符串格式
                    data = eval("(" + data + ")");
                    if (data.success) {
                        //刷新datagrid
                        $("#list").datagrid("reload");
                        //关闭窗口
                        $("#editWin").window("close");

                        $.messager.show({
                            title:"提示",
                            msg:"保存成功"
                        });
                    }else {
                        $.messager.alert("提示","保存失败: " + data.msg, "error");
                    }

                }
            });
        });
    });
/script>
```

html/body/form/input/type | Writable | Smart Insert | 28 : 37

访问:

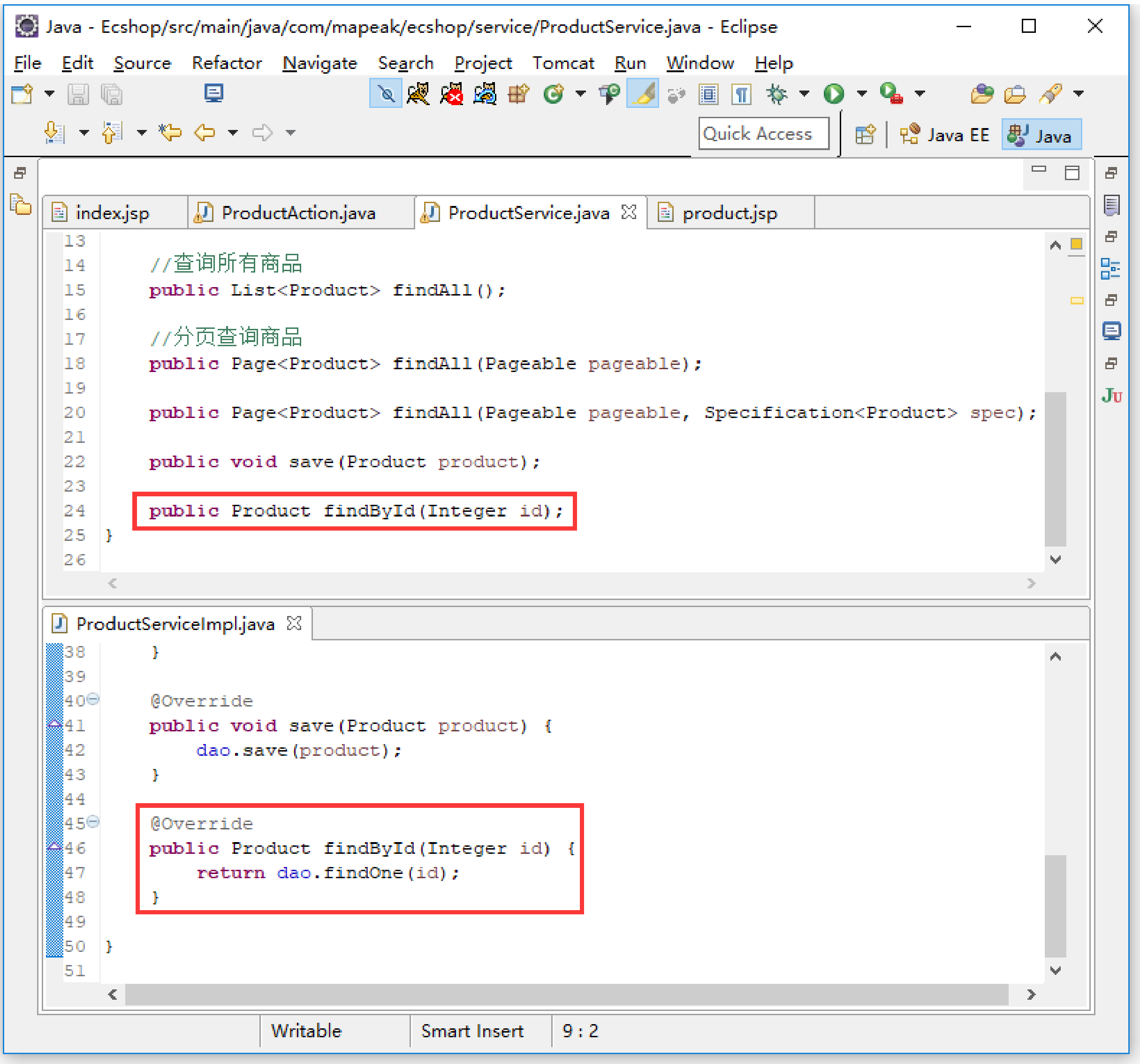


商品编辑

商品名称 格力　商品原价 1688

商品现价 1400　商品描述 格力手机

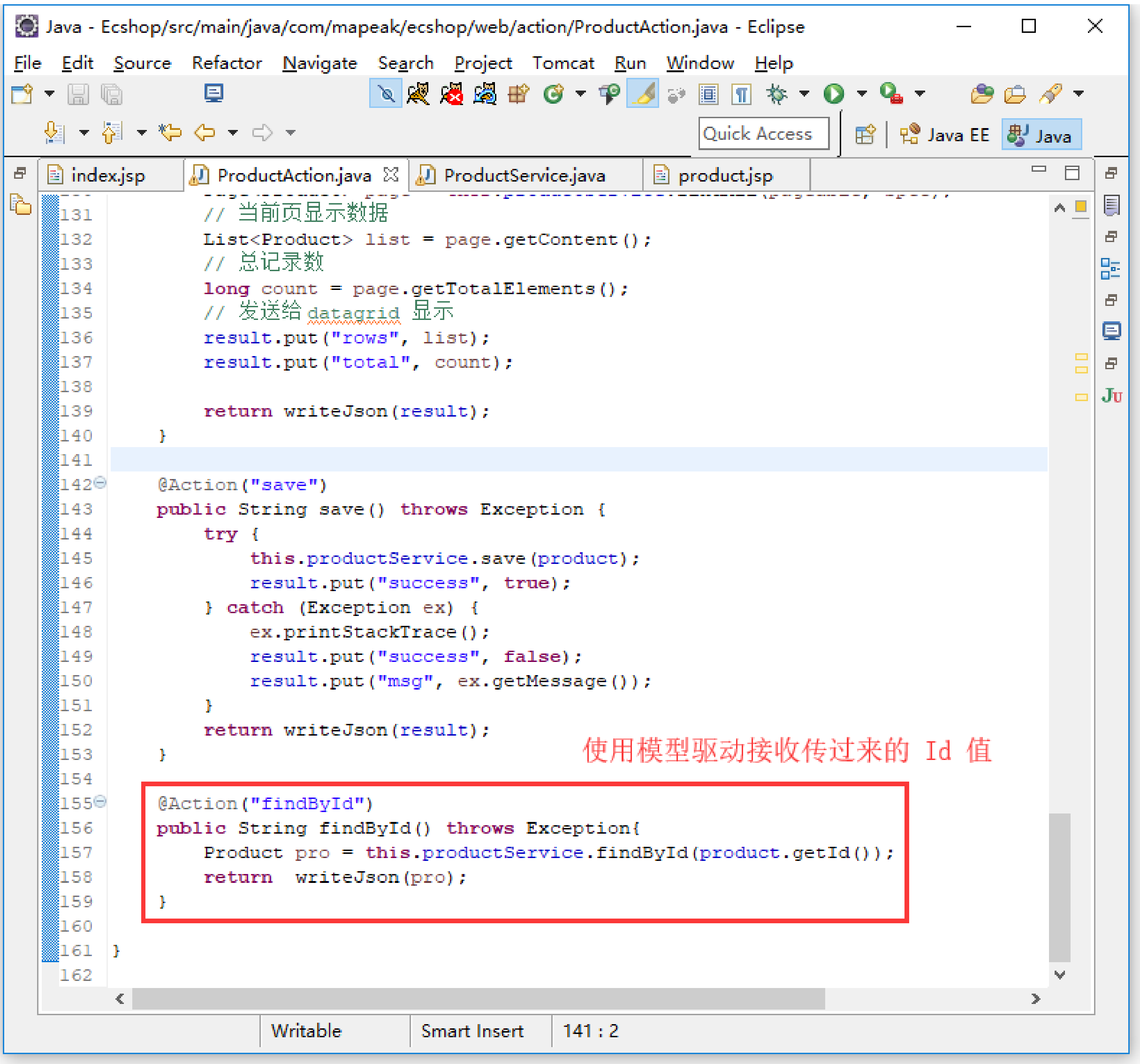
7、Spring Data JPA-商品编辑：
service：



```

    //查询所有商品
    public List<Product> findAll();

    //分页查询商品
    public Page<Product> findAll(Pageable pageable);

    public Page<Product> findAll(Pageable pageable, Specification<Product> spec);

    public void save(Product product);

    public Product findById(Integer id);
}

```

```
    }

    @Override
    public void save(Product product) {
        dao.save(product);
    }

    @Override
    public Product findById(Integer id) {
        return dao.findOne(id);
    }

}

```

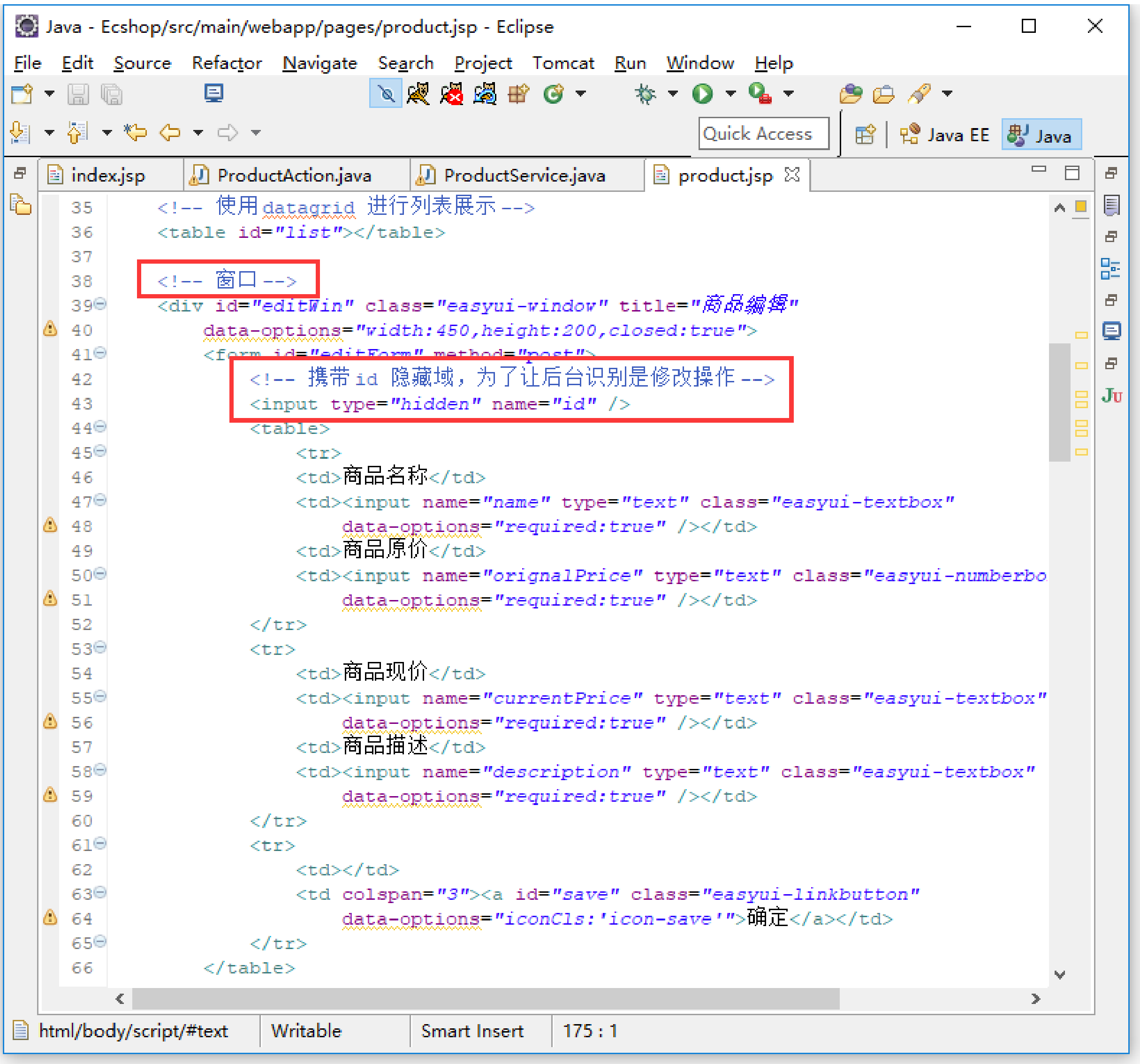
action：


Java - Ecshop/src/main/java/com/mapeak/ecshop/web/action/ProductAction.java - Eclipse
File Edit Source Refactor Navigate Search Project Tomcat Run Window Help
Quick Access
Java EE
Java
index.jsp
ProductAction.java
ProductService.java
product.jsp
Writable
Smart Insert
141 : 2

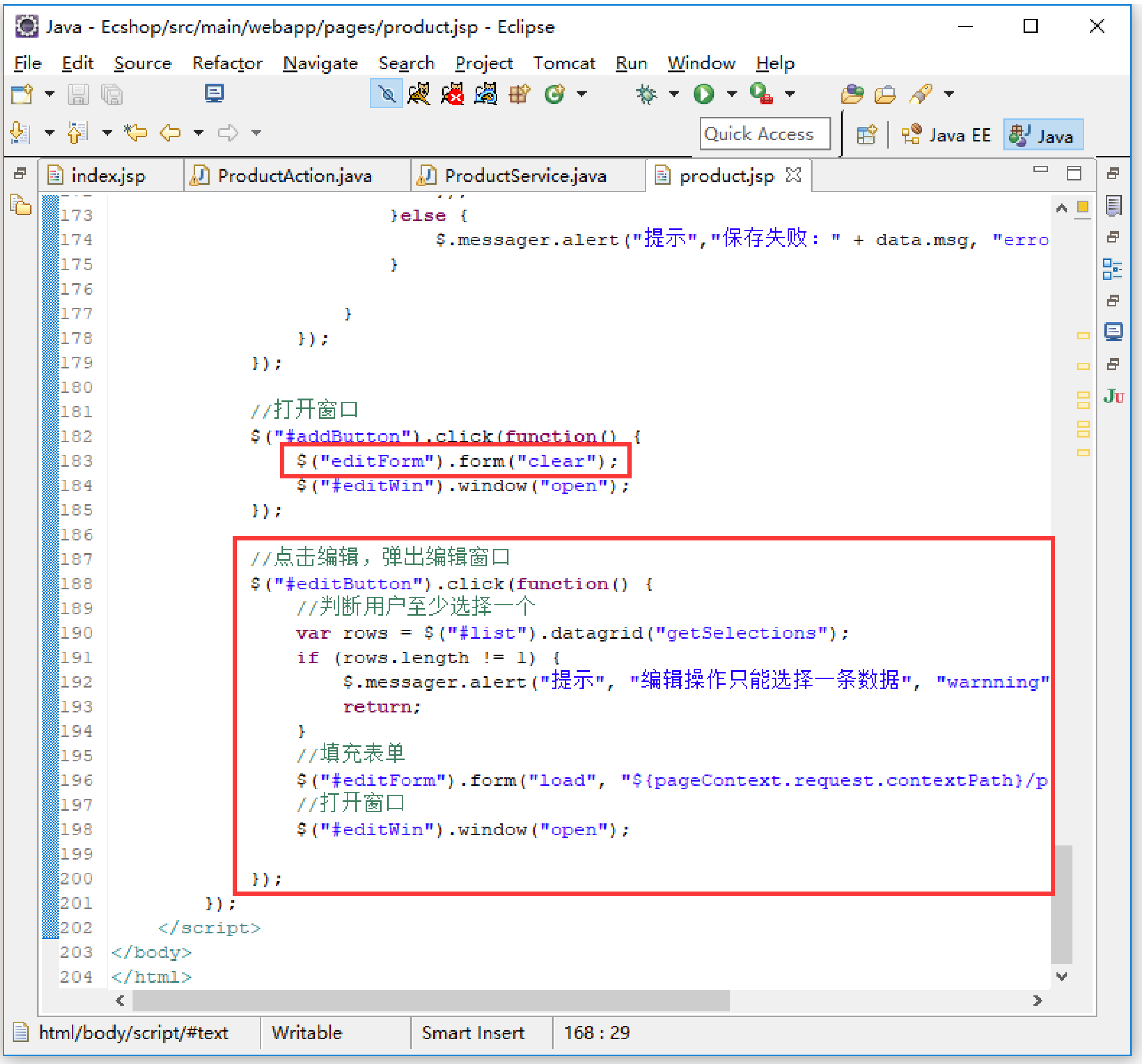
```java
        // 当前页显示数据
        List<Product> list = page.getContent();
        // 总记录数
        long count = page.getTotalElements();
        // 发送给 datagrid 显示
        result.put("rows", list);
        result.put("total", count);

        return writeJson(result);
    }

    @Action("save")
    public String save() throws Exception {
        try {
            this.productService.save(product);
            result.put("success", true);
        } catch (Exception ex) {
            ex.printStackTrace();
            result.put("success", false);
            result.put("msg", ex.getMessage());
        }
        return writeJson(result);
    }

    @Action("findById")
    public String findById() throws Exception{
        Product pro = this.productService.findById(product.getId());
        return  writeJson(pro);
    }

}
```

使用模型驱动接收传过来的 Id 值

UI:


Java - Ecshop/src/main/webapp/pages/product.jsp - Eclipse
File Edit Source Refactor Navigate Search Project Tomcat Run Window Help
Quick Access
Java EE
Java
index.jsp
ProductAction.java
ProductService.java
product.jsp
html/body/script/#text
Writable
Smart Insert
175 : 1

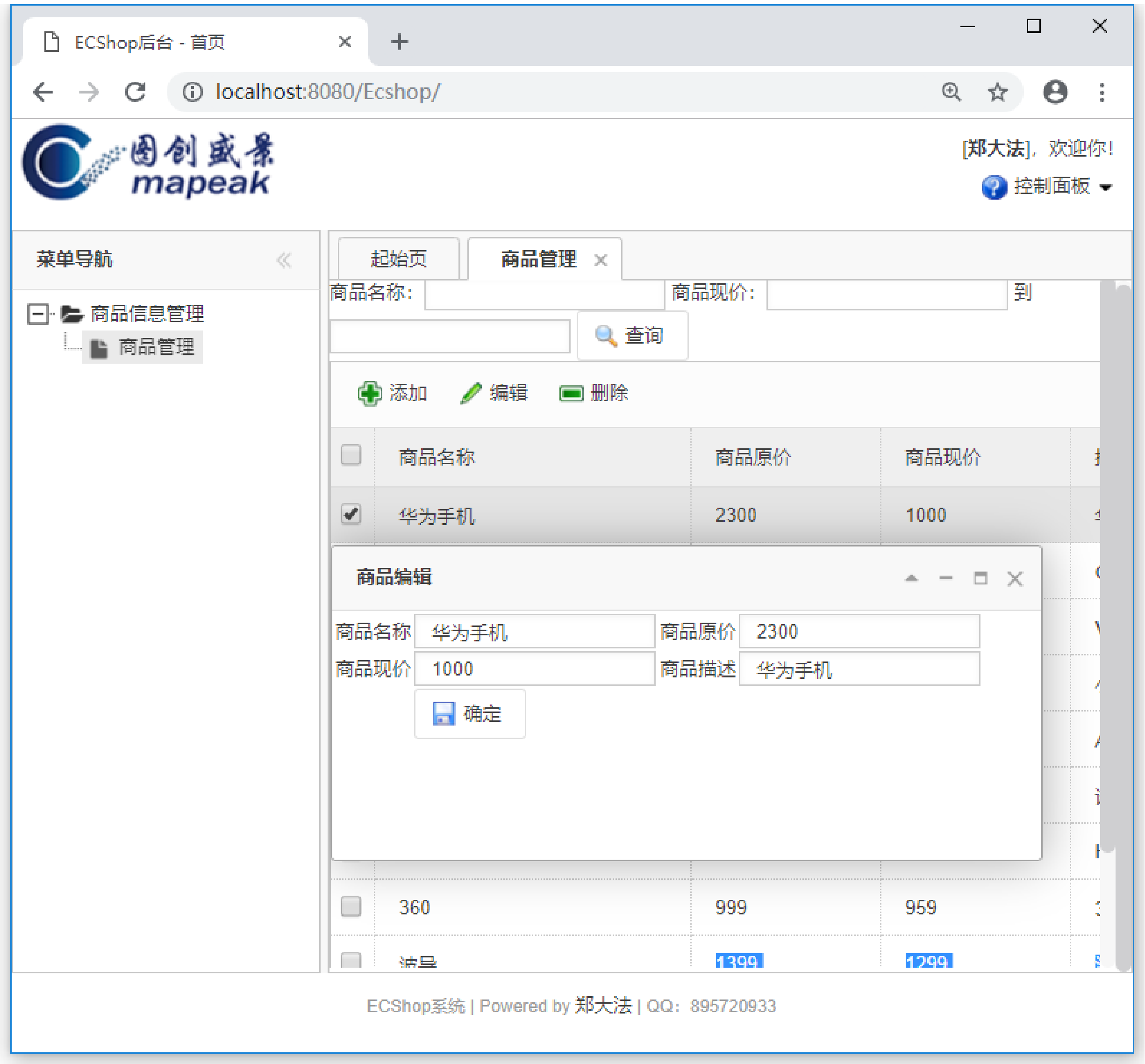
```
    <!-- 使用datagrid 进行列表展示 -->
    <table id="list"></table>

    <!-- 窗口 -->
    <div id="editWin" class="easyui-window" title="商品编辑"
        data-options="width:450,height:200,closed:true">
        <form id="editForm" method="post">
            <!-- 携带id 隐藏域，为了让后台识别是修改操作 -->
            <input type="hidden" name="id" />
            <table>
                <tr>
                    <td>商品名称</td>
                    <td><input name="name" type="text" class="easyui-textbox"
                        data-options="required:true" /></td>
                    <td>商品原价</td>
                    <td><input name="orignalPrice" type="text" class="easyui-numberbo
                        data-options="required:true" /></td>
                </tr>
                <tr>
                    <td>商品现价</td>
                    <td><input name="currentPrice" type="text" class="easyui-textbox"
                        data-options="required:true" /></td>
                    <td>商品描述</td>
                    <td><input name="description" type="text" class="easyui-textbox"
                        data-options="required:true" /></td>
                </tr>
                <tr>
                    <td></td>
                    <td colspan="3"><a id="save" class="easyui-linkbutton"
                        data-options="iconCls:'icon-save'">确定</a></td>
                </tr>
            </table>
```



```
            }else {
              $.messager.alert("提示","保存失败： " + data.msg, "erro
            }

          }
        });
      });

      //打开窗口
      $("#addButton").click(function() {
        $("editForm").form("clear");
        $("#editWin").window("open");
      });

      //点击编辑，弹出编辑窗口
      $("#editButton").click(function() {
        //判断用户至少选择一个
        var rows = $("#list").datagrid("getSelections");
        if (rows.length != 1) {
          $.messager.alert("提示", "编辑操作只能选择一条数据", "warnning"
          return;
        }
        //填充表单
        $("#editForm").form("load", "${pageContext.request.contextPath}/p
        //打开窗口
        $("#editWin").window("open");

      });
    });
  </script>
</body>
</html>
```

```
//点击编辑，弹出编辑窗口
$("#editButton").click(function() {
  //判断用户至少选择一个
  var rows = $("#list").datagrid("getSelections");
  if (rows.length != 1) {
    $.messager.alert("提示", "编辑操作只能选择一条数据", "warnning");
    return;
  }
  //填充表单
  $("#editForm").form("load", "${pageContext.request.contextPath}/product/findById.action?id="+rows[0].id);
  //打开窗口
  $("#editWin").window("open");
});
```

访问：



	商品名称	商品原价	商品现价
☑	华为手机	2300	1000
□	360	999	959
□	波导	1399	1299

商品编辑

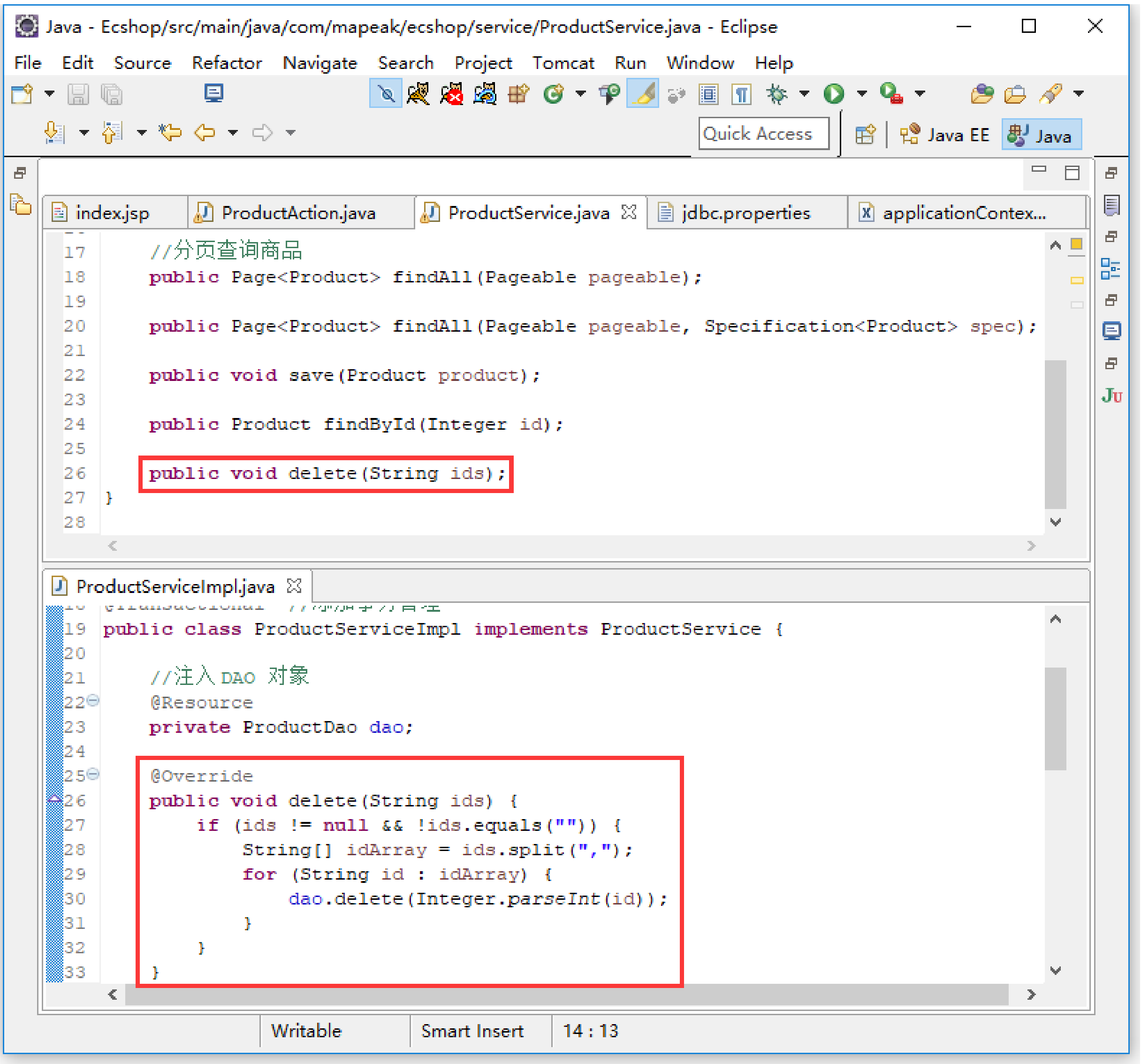
商品名称 华为手机 商品原价 2300

商品现价 1000 商品描述 华为手机

ECShop系统 | Powered by 郑大法 | QQ：895720933

8、Spring Data JPA-商品批量删除：

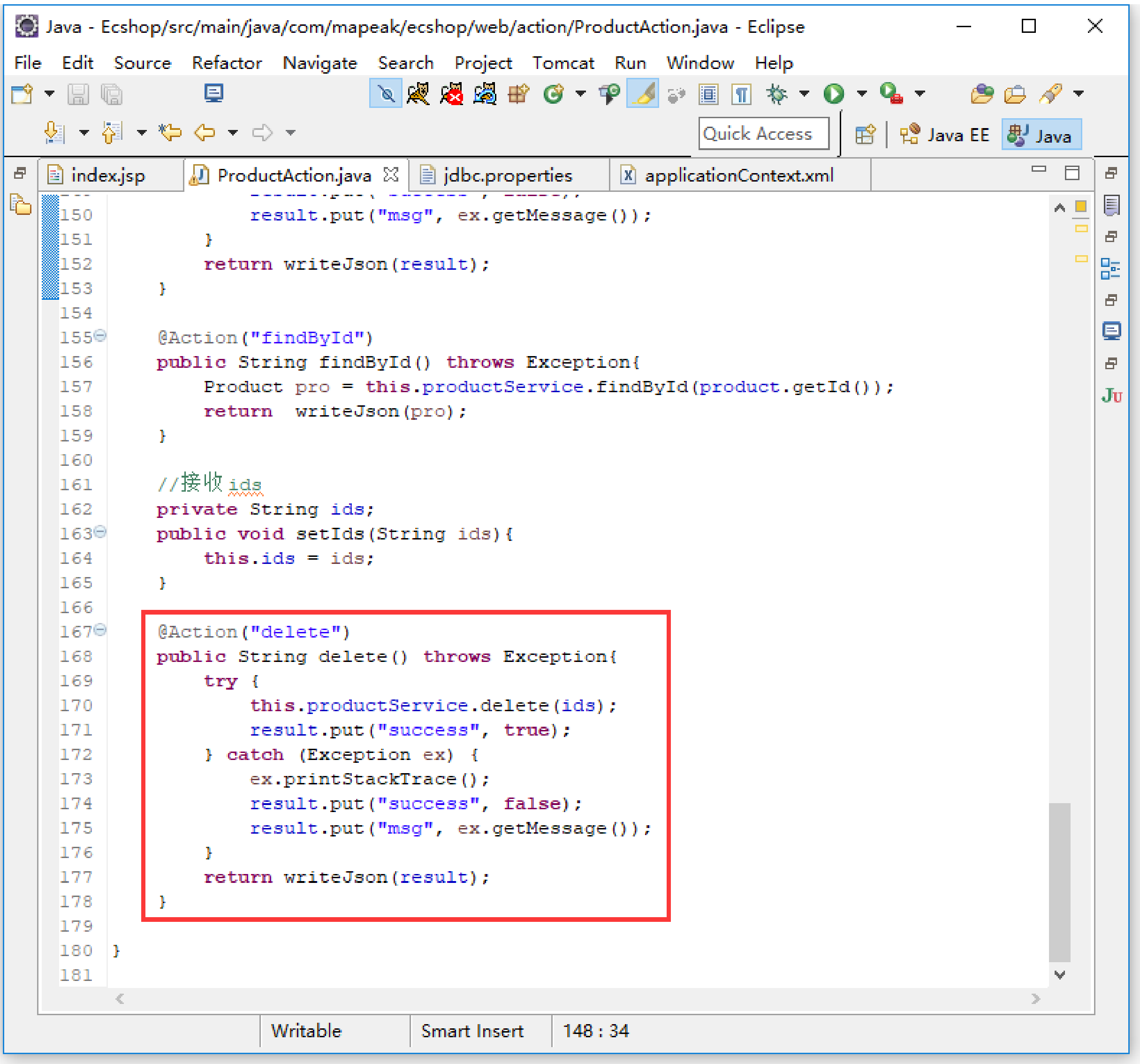
service：


Java - Ecshop/src/main/java/com/mapeak/ecshop/service/ProductService.java - Eclipse
File Edit Source Refactor Navigate Search Project Tomcat Run Window Help
Quick Access
Java EE
Java
index.jsp
ProductAction.java
ProductService.java
jdbc.properties
applicationContex...
ProductServiceImpl.java
Writable
Smart Insert
14 : 13

```java
    //分页查询商品
    public Page<Product> findAll(Pageable pageable);

    public Page<Product> findAll(Pageable pageable, Specification<Product> spec);

    public void save(Product product);

    public Product findById(Integer id);

    public void delete(String ids);
}

```

```java
@Transactional  //添加事务管理
public class ProductServiceImpl implements ProductService {

    //注入 DAO 对象
    @Resource
    private ProductDao dao;

    @Override
    public void delete(String ids) {
        if (ids != null && !ids.equals("")) {
            String[] idArray = ids.split(",");
            for (String id : idArray) {
                dao.delete(Integer.parseInt(id));
            }
        }
    }
```

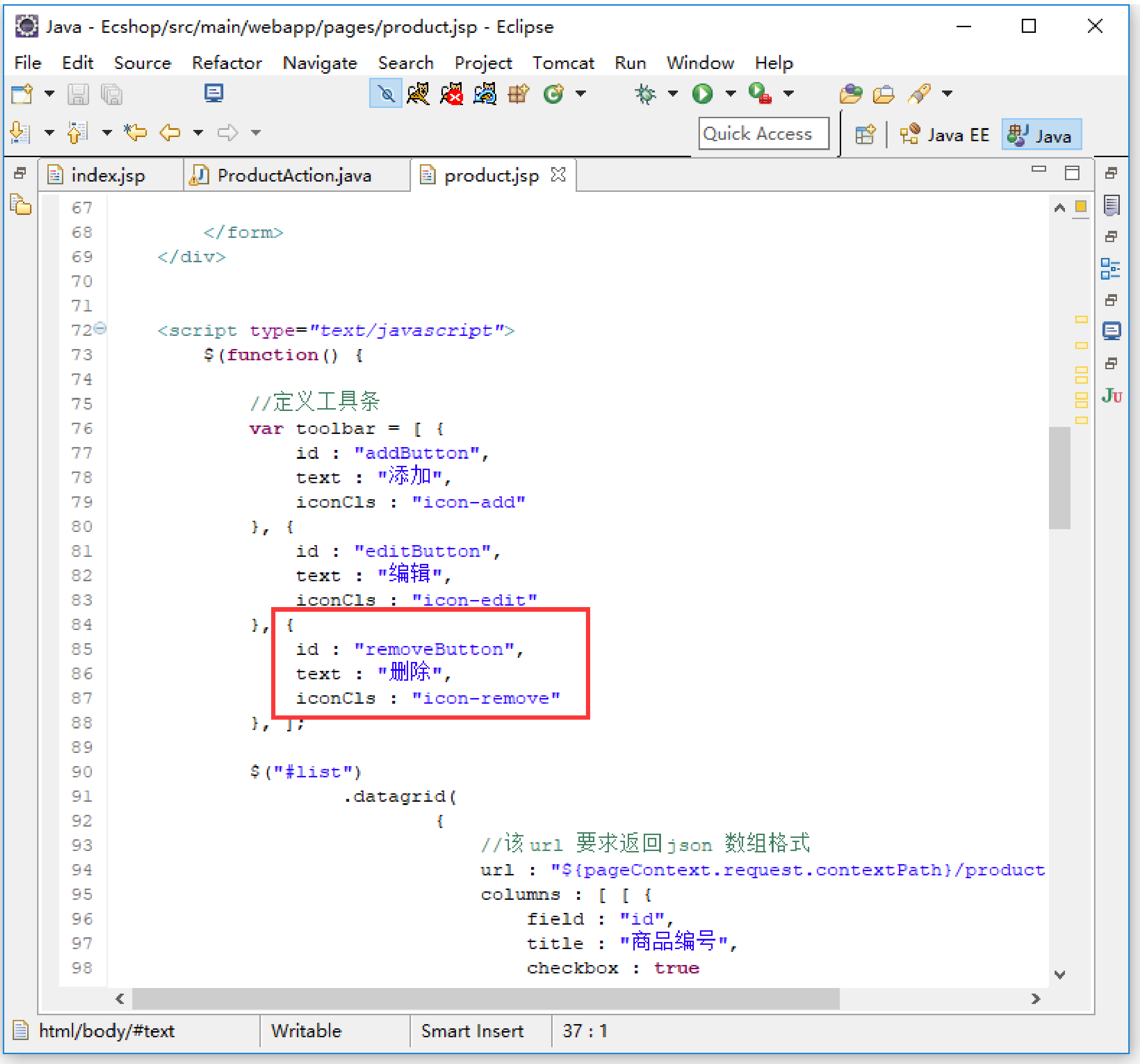
action：



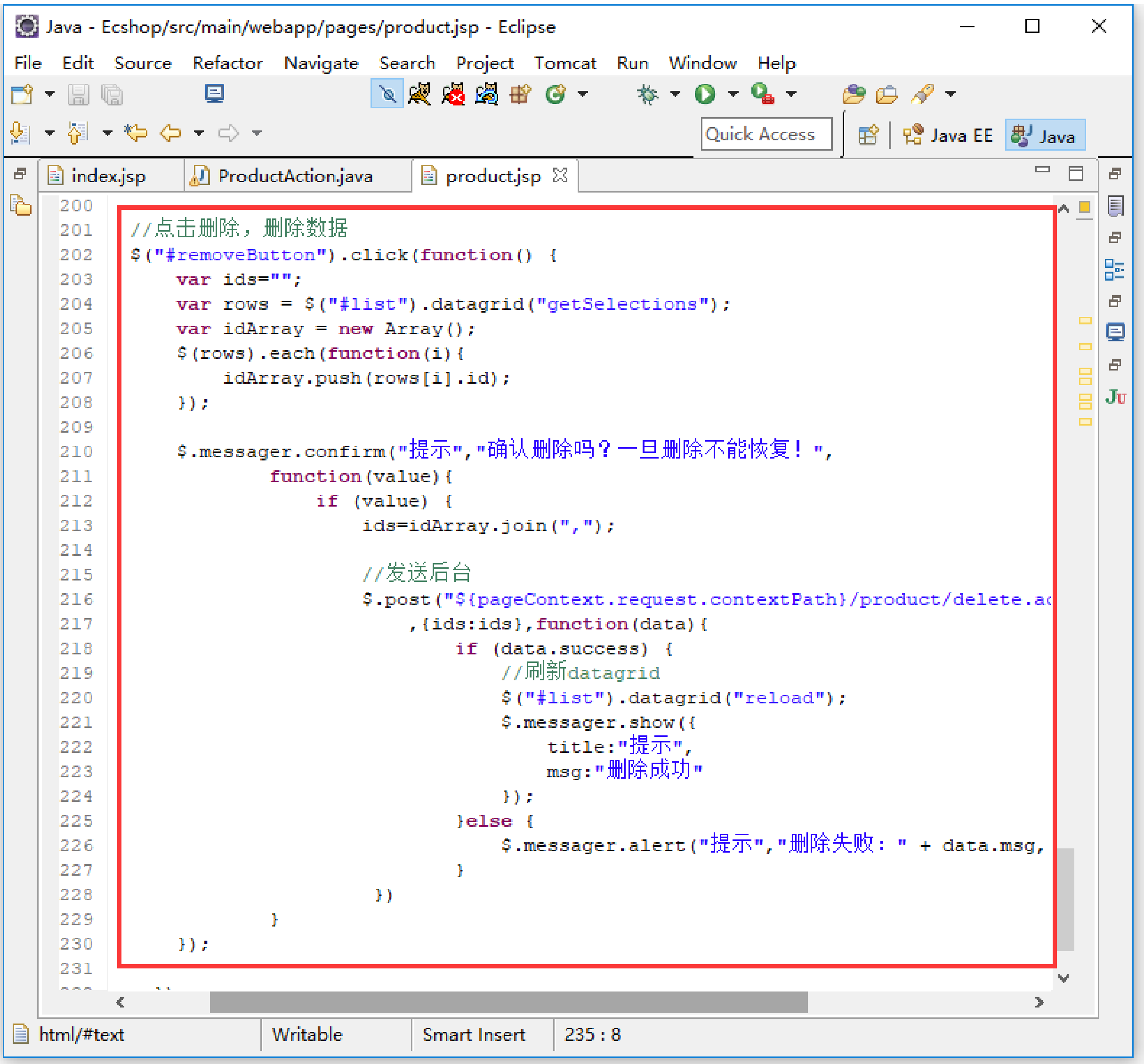
Java - Ecshop/src/main/java/com/mapeak/ecshop/web/action/ProductAction.java - Eclipse
File Edit Source Refactor Navigate Search Project Tomcat Run Window Help
Quick Access
Java EE
Java
index.jsp
ProductAction.java
jdbc.properties
applicationContext.xml
Writable
Smart Insert
148 : 34

```java
            result.put("msg", ex.getMessage());
        }
        return writeJson(result);
    }

    @Action("findById")
    public String findById() throws Exception{
        Product pro = this.productService.findById(product.getId());
        return  writeJson(pro);
    }

    //接收 ids
    private String ids;
    public void setIds(String ids){
        this.ids = ids;
    }

    @Action("delete")
    public String delete() throws Exception{
        try {
            this.productService.delete(ids);
            result.put("success", true);
        } catch (Exception ex) {
            ex.printStackTrace();
            result.put("success", false);
            result.put("msg", ex.getMessage());
        }
        return writeJson(result);
    }

}
```

UI:


Java - Ecshop/src/main/webapp/pages/product.jsp - Eclipse
File Edit Source Refactor Navigate Search Project Tomcat Run Window Help
Quick Access
Java EE
Java
index.jsp
ProductAction.java
product.jsp
html/body/#text
Writable
Smart Insert
37 : 1

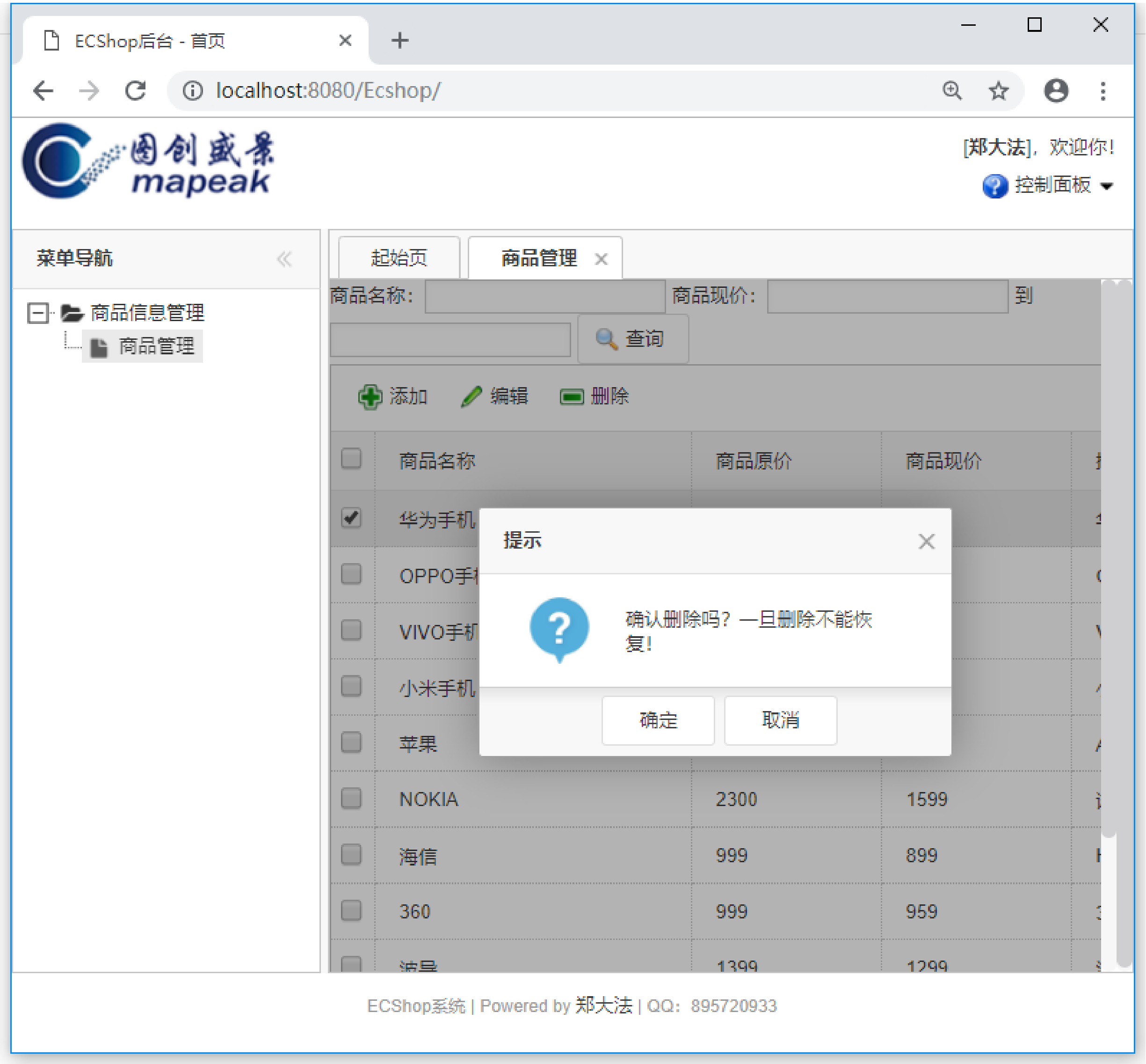
```

            </form>
        </div>


        <script type="text/javascript">
            $(function() {

                //定义工具条
                var toolbar = [ {
                    id : "addButton",
                    text : "添加",
                    iconCls : "icon-add"
                }, {
                    id : "editButton",
                    text : "编辑",
                    iconCls : "icon-edit"
                }, {
                    id : "removeButton",
                    text : "删除",
                    iconCls : "icon-remove"
                }, ];

                $("#list")
                        .datagrid(
                                {
                                    //该 url 要求返回 json 数组格式
                                    url : "${pageContext.request.contextPath}/product
                                    columns : [ [ {
                                        field : "id",
                                        title : "商品编号",
                                        checkbox : true
```


Java - Ecshop/src/main/webapp/pages/product.jsp - Eclipse
File Edit Source Refactor Navigate Search Project Tomcat Run Window Help
Quick Access
Java EE
Java
index.jsp
ProductAction.java
product.jsp
html/#text
Writable
Smart Insert
235 : 8

```
//点击删除，删除数据
$("#removeButton").click(function() {
    var ids="";
    var rows = $("#list").datagrid("getSelections");
    var idArray = new Array();
    $(rows).each(function(i){
        idArray.push(rows[i].id);
    });

    $.messager.confirm("提示","确认删除吗？一旦删除不能恢复！",
            function(value){
                if (value) {
                    ids=idArray.join(",");

                    //发送后台
                    $.post("${pageContext.request.contextPath}/product/delete.ac
                        ,{ids:ids},function(data){
                            if (data.success) {
                                //刷新datagrid
                                $("#list").datagrid("reload");
                                $.messager.show({
                                    title:"提示",
                                    msg:"删除成功"
                                });
                            }else {
                                $.messager.alert("提示","删除失败: " + data.msg,
                            }
                     })
            }
    });
```

```
//点击删除，删除数据
$("#removeButton").click(function() {
    var ids="";
    var rows = $("#list").datagrid("getSelections");
    var idArray = new Array();
    $(rows).each(function(i){
        idArray.push(rows[i].id);
    });

    $.messager.confirm("提示","确认删除吗？一旦删除不能恢复！",
            function(value){
                if (value) {
                    ids=idArray.join(",");

                    //发送后台
                    $.post("${pageContext.request.contextPath}/product/delete.action"
                        ,{ids:ids},function(data){
                            if (data.success) {
                                //刷新datagrid
                                $("#list").datagrid("reload");
                                $.messager.show({
                                    title:"提示",
                                    msg:"删除成功"
                                });
                            }else {
                                $.messager.alert("提示","删除失败： " + data.msg, "error");
                            }
                     })
            }
    });
});
```

访问：



确认删除吗？一旦删除不能恢复！

☐	商品名称	商品原价	商品现价
☑	华为手机		
☐	OPPO手		
☐	VIVO手机		
☐	小米手机		
☐	苹果		
☐	NOKIA	2300	1599
☐	海信	999	899
☐	360	999	959
☐	油号	1399	1299

至此：

项目完成ヾ(๑╹◡╹)ﾉ"