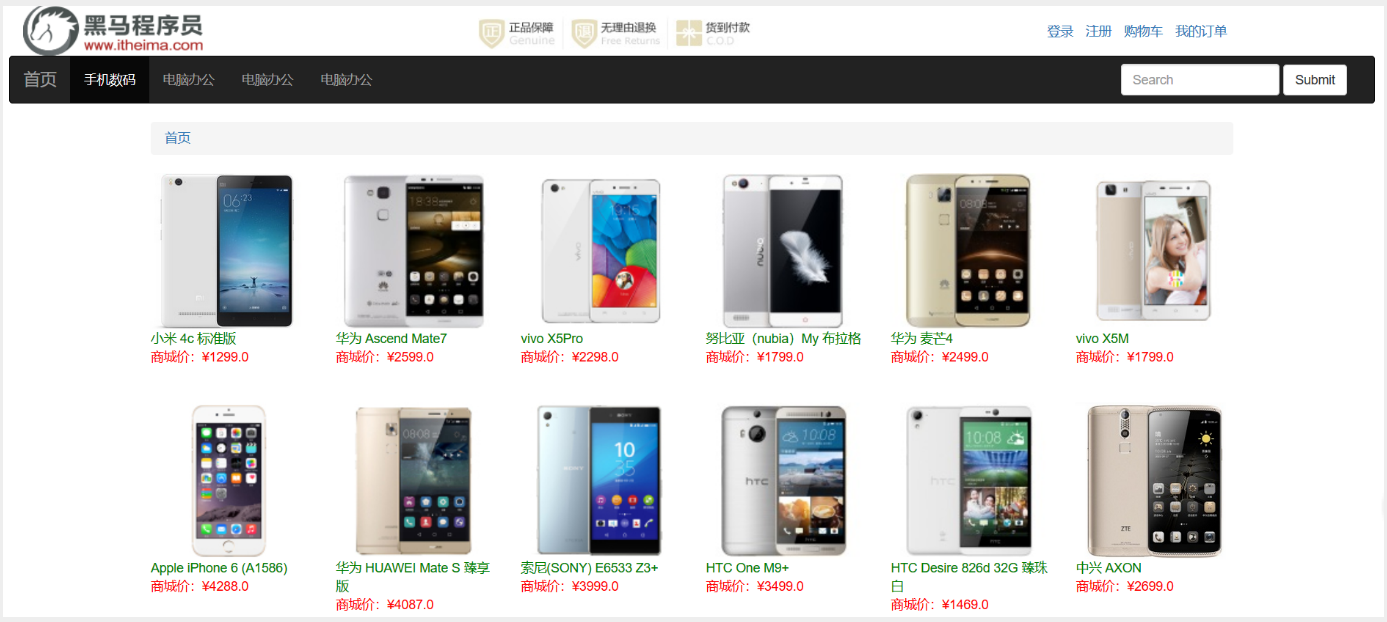
用EL和JSTL显示商品信息



小米 4c 标准版
商城价：¥1299.0

华为 Ascend Mate7
商城价：¥2599.0

vivo X5Pro
商城价：¥2298.0

努比亚（nubia）My 布拉格
商城价：¥1799.0

华为 麦芒4
商城价：¥2499.0

vivo X5M
商城价：¥1799.0

Apple iPhone 6 (A1586)
商城价：¥4288.0

华为 HUAWEI Mate S 臻享版
商城价：¥4087.0

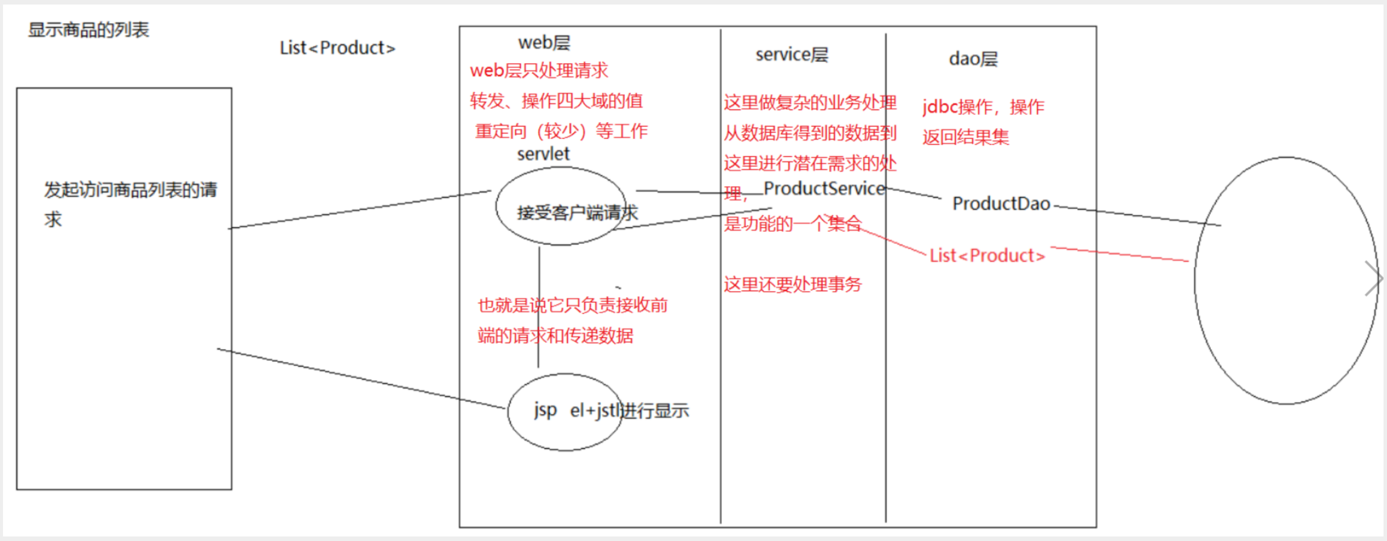
索尼(SONY) E6533 Z3+
商城价：¥3999.0

HTC One M9+
商城价：¥3499.0

HTC Desire 826d 32G 臻珠白
商城价：¥1469.0

中兴 AXON
商城价：¥2699.0

一、案例图解



web层只处理请求
转发、操作四大域的值
重定向（较少）等工作

这里做复杂的业务处理
从数据库得到的数据到
这里进行潜在需求的处
理，是功能的一个集合

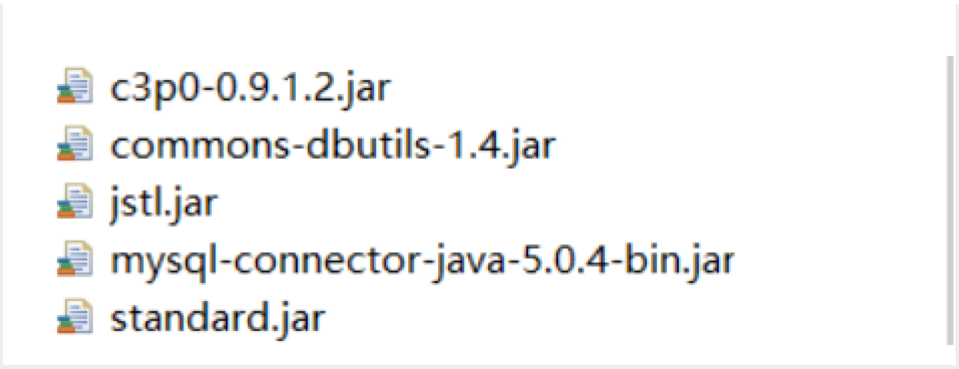
jdbc操作，操作
返回结果集

这里还要处理事务

也就是说它只负责接收前
端的请求和传递数据

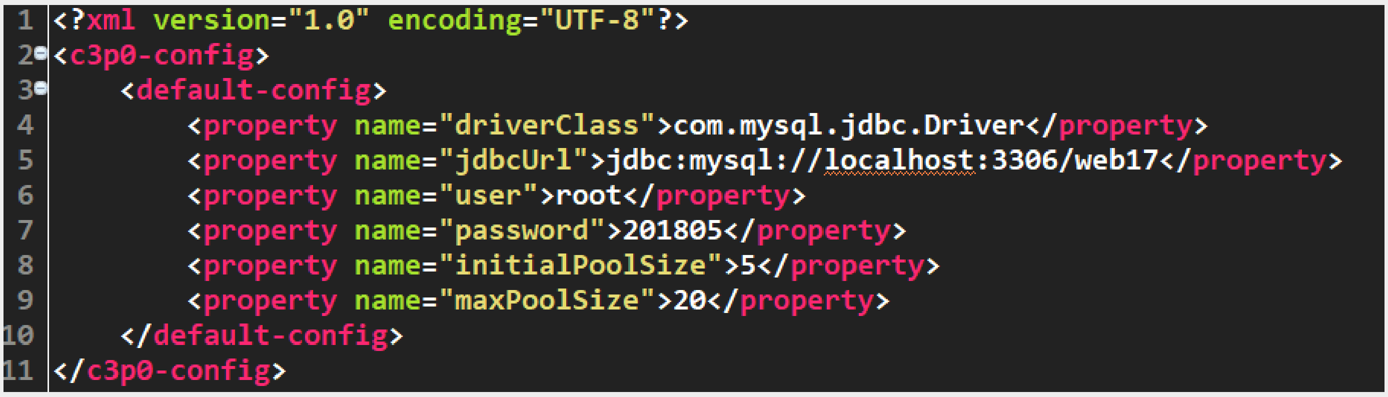
二、环境搭建

1. 先导入包:操作数据库的mysql驱动包、c3p0连接池包、DBUtils包、JSTL相关的包(两个,jstl.jar和standard.jar)



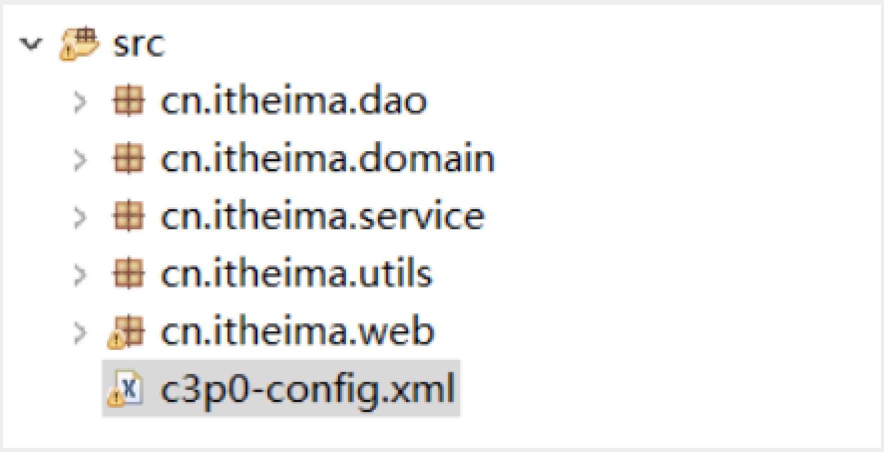
c3p0-0.9.1.2.jar
commons-dbutils-1.4.jar
jstl.jar
mysql-connector-java-5.0.4-bin.jar
standard.jar

2. 导入c3p0配置文件到src目录下



```xml
<?xml version="1.0" encoding="UTF-8"?>
<c3p0-config>
    <default-config>
        <property name="driverClass">com.mysql.jdbc.Driver</property>
        <property name="jdbcUrl">jdbc:mysql://localhost:3306/web17</property>
        <property name="user">root</property>
        <property name="password">201805</property>
        <property name="initialPoolSize">5</property>
        <property name="maxPoolSize">20</property>
    </default-config>
</c3p0-config>
```

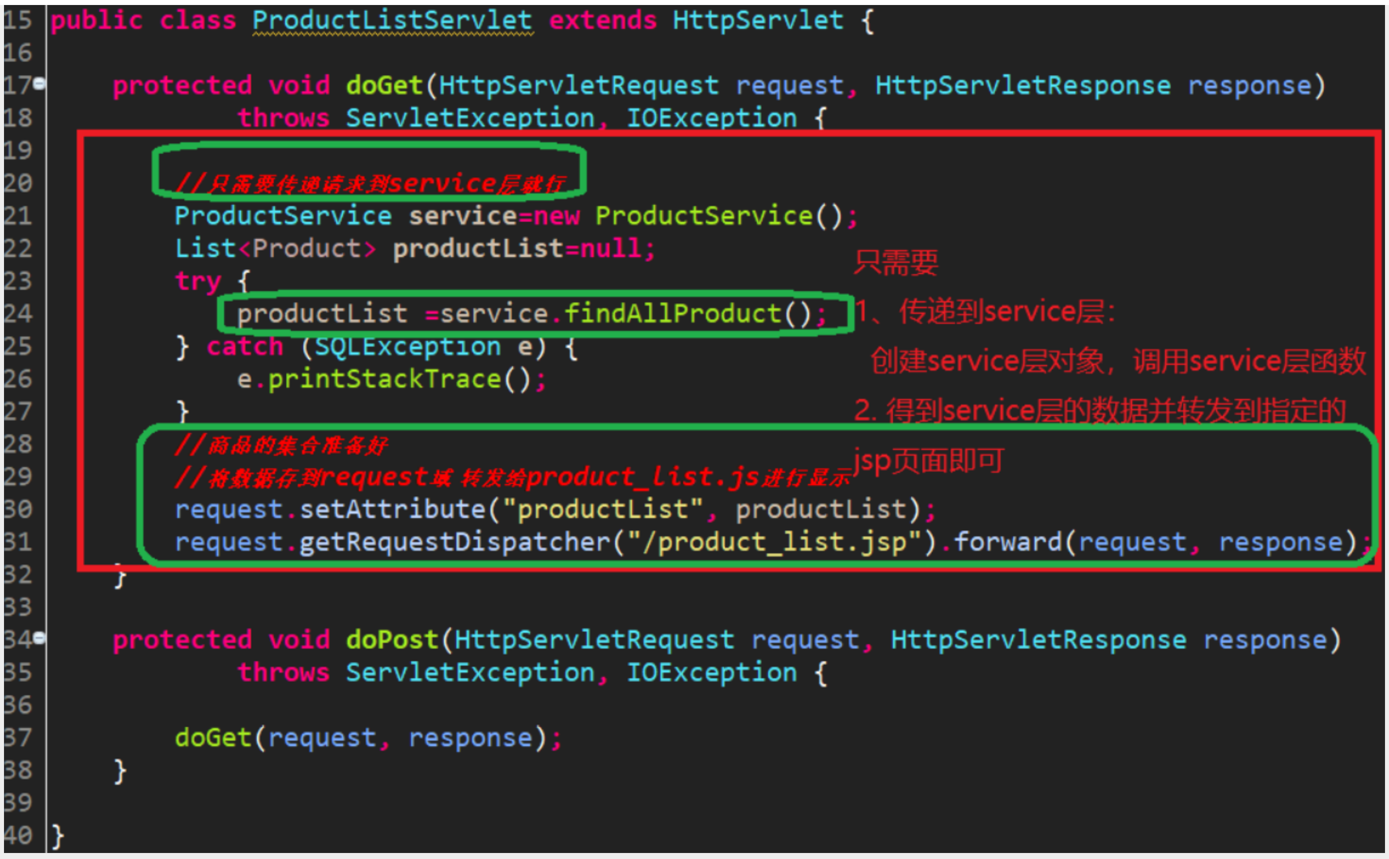
3. 建包web包、service包、dao包、domain包、dbutils包,web包处理请求，建立servlet；service包建立类，处理业务；dao包建立类，执行sql语句；domain包封装实体类；dbutils包封装数据库操作(此包每时每刻都要用基本)。



src
cn.itheima.dao
cn.itheima.domain
cn.itheima.service
cn.itheima.utils
cn.itheima.web
c3p0-config.xml

三、代码详解

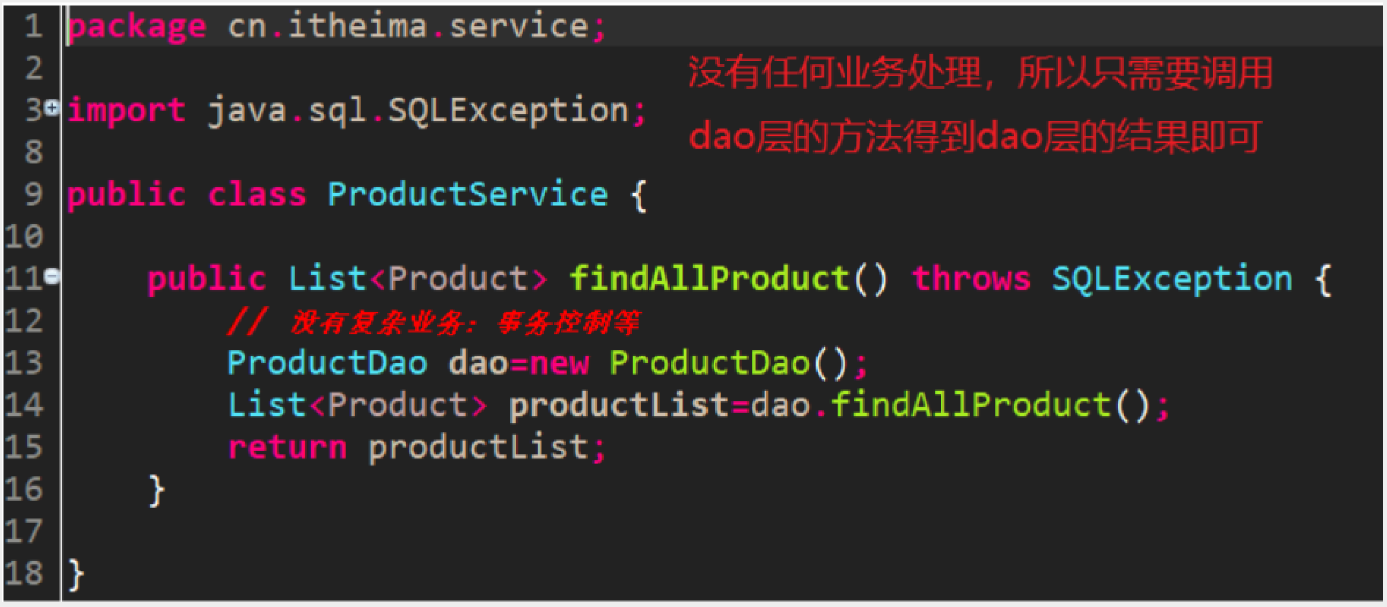
3.1 web层



```java
public class ProductListServlet extends HttpServlet {

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

        //只需要传递请求到service层就行
        ProductService service=new ProductService();
        List<Product> productList=null;
        try {
            productList =service.findAllProduct();
        } catch (SQLException e) {
            e.printStackTrace();
        }
        //商品的集合准备好
        //将数据存到request域 转发给product_list.js进行显示
        request.setAttribute("productList", productList);
        request.getRequestDispatcher("/product_list.jsp").forward(request, response);
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

        doGet(request, response);
    }

}
```

只需要
1、传递到service层：
创建service层对象，调用service层函数
2. 得到service层的数据并转发到指定的jsp页面即可

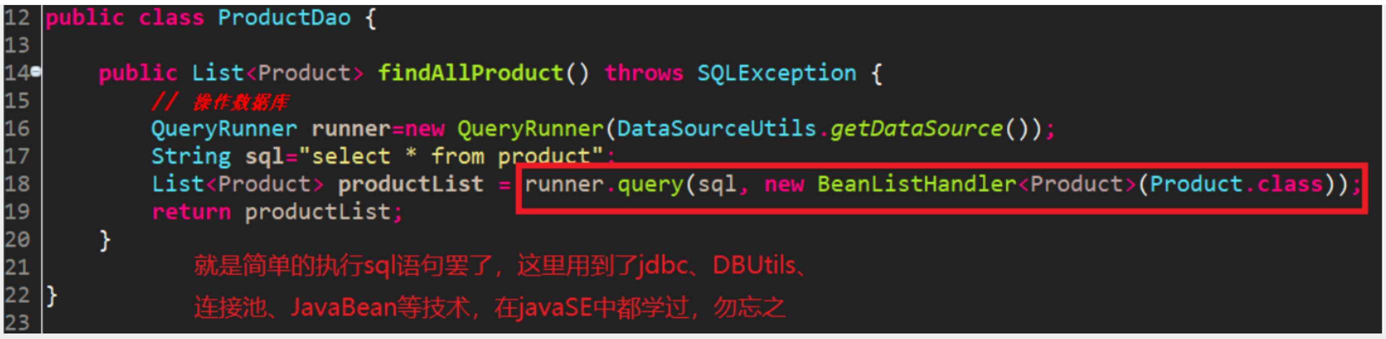
3.2 service层



```java
package cn.itheima.service;

import java.sql.SQLException;

public class ProductService {

    public List<Product> findAllProduct() throws SQLException {
        // 没有复杂业务：事务控制等
        ProductDao dao=new ProductDao();
        List<Product> productList=dao.findAllProduct();
        return productList;
    }

}
```

没有任何业务处理，所以只需要调用dao层的方法得到dao层的结果即可

3.3 dao层

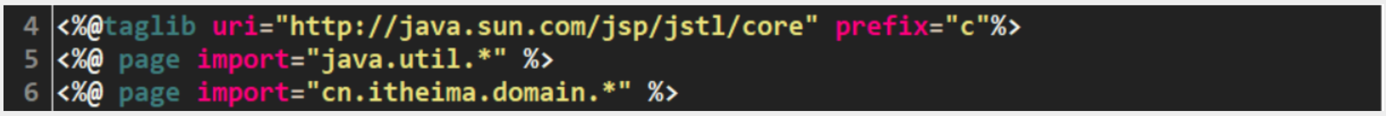


```java
public class ProductDao {

    public List<Product> findAllProduct() throws SQLException {
        // 操作数据库
        QueryRunner runner=new QueryRunner(DataSourceUtils.getDataSource());
        String sql="select * from product";
        List<Product> productList = runner.query(sql, new BeanListHandler<Product>(Product.class));
        return productList;
    }
}
```

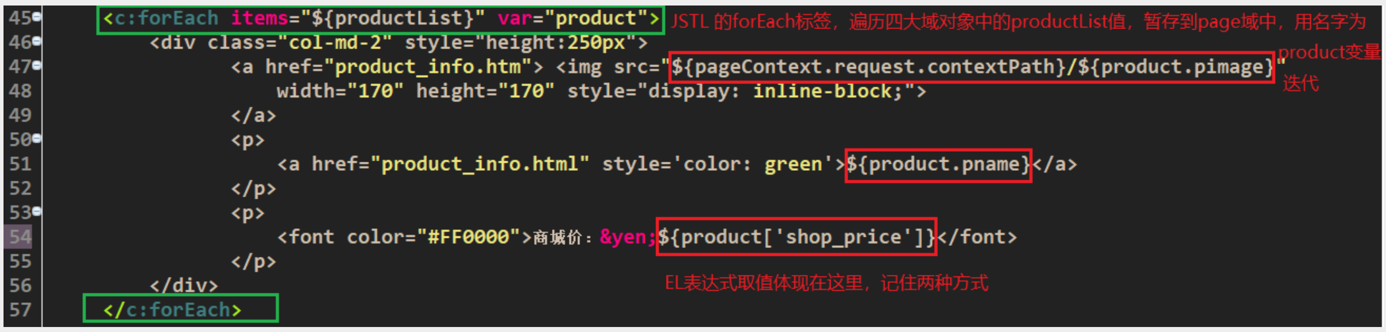
就是简单的执行sql语句罢了，这里用到了jdbc、DBUtils、
连接池、JavaBean等技术，在javaSE中都学过，勿忘之

3.4 jsp页面显示代码改写

找到每个商品显示的div代码，用JSTL代码替代，以显示动态资源



```jsp
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ page import="java.util.*" %>
<%@ page import="cn.itheima.domain.*" %>
```



```jsp
    <c:forEach items="${productList}" var="product">
        <div class="col-md-2" style="height:250px">
            <a href="product_info.htm"> <img src="${pageContext.request.contextPath}/${product.pimage}"
                width="170" height="170" style="display: inline-block;">
            </a>
            <p>
                <a href="product_info.html" style='color: green'>${product.pname}</a>
            </p>
            <p>
                <font color="#FF0000">商城价：&yen;${product['shop_price']}</font>
            </p>
        </div>
    </c:forEach>
```

JSTL 的forEach标签，遍历四大域对象中的productList值，暂存到page域中，用名字为product变量迭代

EL表达式取值体现在这里，记住两种方式

注意写路径的时候最好在前面加上`${pageContext.request.contextPath}`

四、注意

运行时是访问web层的servlet以间接访问显示商品的那个jsp页面