

Web Lab 5 JSP/Servlet

实验目的

- 1) 复习和掌握TOMCAT配置
- 2) 掌握JSP 基本语法
- 3) 掌握使用 JSP , JavaBean , Servlet 进行编程
- 4) 掌握使用Session 进行数据传递的方法

实验任务

通过设计 JSP、Servlet、JavaBean 使用 MVC 模式实现用户登录验证、注销，购物车的添加、移除以及清除商品，页面之间使用 session 传递消息。

实验环境

Servlet/JSP 容器: Apache Tomcat 5.5 或以上

JDK: Sun JDK 1.5 或以上

浏览器: Internet Explorer 6 或以上, Firefox 1.5 或以上

IDE: 任意熟悉的 IDE (建议 MyEclipse)

实验检查

完成实验内容后，由助教检查编写的页面效果。

检查清单：

JSP

index.jsp (商品展示界面)

Login.html (用户登录)

Shopcart.jsp (购物车显示界面)

Servlet

Usercheck.java (登录注销处理)

Sessioncart.java (购物车操作处理)

JavaBean

cart.java (购物车中的单个商品)

Category.java (商品目录)

Product.java (商品信息)

Eshop.java (商店)

Userinfo.java (用户信息)

War

Lab6.war

实验检查通过后，请将自己编写的以上文件打包上传到ftp 上的WORK_UPLOAD/lab6/你的学号目录下，其中学号目录需要自己建立。

实验步骤

本实验使用MVC模式实现

View: 负责应用程序的显示 JSP/HTML

Modle:代表业务逻辑 JavaBean

Controller: 负责过程控制 Servlet

参考Lab1中关于MyEclipse和Tomcat的配置,配置成功后在MyEclipse中新建项目,在项目下新建包estore和servlet。

1. Model的设计

实验模拟一个简单的购物流程,涉及到的业务逻辑包括:用户信息、商品、商品目录、商店、由于购物车通过Session保存,在此只将购物车中的相同商品作为一个实体。

1.1 用户信息

用户信息包含用户名和密码,同时提供二者时候匹配的方法,在此项目的包estore下添加Java文件,代码中使用一个静态的ArrayList保存由“用户名”-“密码”配对组成的userinfo。可以根据自身情况变换其中的用户名和密码。其中最后一个方法实现用户验证。

1.2 商品信息、商品目录、商店、购物车单种商品的实体设计

在项目包estore下添加如下文件: cart.java、 Category.java、 Product.java、 Eshop.java

注:各个JavaBean的源文件在本文档目录给出。

2. View的设计

本实验中的购物,对外显示包括商品展示页面、用户登录页面以及显示购物车页面。

2.1 商品展示页面设计

此页面中需要根据商品的分类列出不同的商品,每件商品必须提供用户可以点击购买的链接,点击链接后,如果购买成功,应该跳转到购物车页面,提示用户购买成功。在页面上提供用户可以登录的链接,对于未登录的用户,只能浏览商品,不能购买(这个功能下一步会通过Servlet实现)。

其中判断当前时候有用户已登录需要通过session记录用户,实现方法如下

```
if (session.getAttribute("username") == null)
    out.print("Not Login. <a href='login.html'>Login Now!</a>");
else{
    out.print("Welcome " + session.getAttribute("username"));
    out.print("&nbsp;&nbsp;<a href=\"\" + request.getContextPath() +
\"/servlet/usercheck?status='logout'\">Logout</a>");}
```

展示商品的页面index.jsp源代码如下

```
<%@ page language="java" pageEncoding="ISO-8859-1"
import="java.util.*"%>
<%!
    public void jspInit() {
        getServletContext().setAttribute("cats",
estore.EShop.getCats()); }
    public void jspDestroy() {
```



```

<td><a href="<%= request.getContextPath() +
    "/servlet/sessioncart?action=add&sku=" +
    curItem.getSku() %>">
    <b>BUY</b></a></td>
</tr>
<% } %>
</table></td></tr></table>
</body>
</html>

```

购买商品链接到自定义的servlet sessioncart保存购买的商品，并重定向到购物车显示页面。效果图如下

Welcome admin [Logout](#)

Systems Software Books	<table border="1"> <thead> <tr> <th>Item</th> <th>Price</th> <th>Order</th> </tr> </thead> <tbody> <tr> <td>Pentium 4 - 4 GHz, 512 MB, 300 GB</td> <td>\$989.99</td> <td>BUY</td> </tr> <tr> <td>AMD Opteron - 4 GHz, 1 GB, 300 GB</td> <td>\$1200.99</td> <td>BUY</td> </tr> </tbody> </table>	Item	Price	Order	Pentium 4 - 4 GHz, 512 MB, 300 GB	\$989.99	BUY	AMD Opteron - 4 GHz, 1 GB, 300 GB	\$1200.99	BUY
Item	Price	Order								
Pentium 4 - 4 GHz, 512 MB, 300 GB	\$989.99	BUY								
AMD Opteron - 4 GHz, 1 GB, 300 GB	\$1200.99	BUY								

2.2 购物车显示页面设计

此页面显示用户所购商品的列表，同时用户可以单独增加和减少列表中的商品，以及清空购物车，这些操作都通过链接到自定义servlet的sessioncart下完成。显示用户购物车中的商品列表，需要从session中读取用户的购物车，实现如下

```
ArrayList mycart = (ArrayList)session.getAttribute("mycart");
```

购物车页面需要判断用户是否已登录，通过session实现，若未登录，跳转到登录页面

```

if(session.getAttribute("username") == null)
    response.sendRedirect(request.getContextPath()+"/login.html");

```

购物车页面shopcart.jsp源文件如下

```

<%@ page language="java" import="java.util.*"
pageEncoding="ISO-8859-1"%>
<jsp:directive.page import="estore.*"/>
<%!
    private static String SHOP_PAGE = "/index.jsp";
    private static String CART_PAGE = "/servlet/sessioncart";
    private String dispPrice( String price) {
        int len = price.length();
        if (len <= 2)
            return price;
        else
            return "$" + price.substring(0,len -2) + "." +

```

```

+ price.substring(len-2);    }    %>
<%if(session.getAttribute("username") == null)
    response.sendRedirect(request.getContextPath()+"/login.html");
%>
<html> <head> <title>My Shopping Cart</title> </head>
  <body>  <table width="600">
<tr><td>  <table border="1">
  <tr><th colspan="6">Your Shopping Cart</th></tr>
  <tr><th align="left">Quantity</th><th align="left">Item</th>
    <th align="right">Price</th><th align="right">Extended</th>
    <th align="left">Add</th><th align="left">Remove</th> </tr>
  <% long total = 0;
    ArrayList mycart = (ArrayList)session.getAttribute("mycart");
    if(mycart != null && mycart.size() > 0){
      for(int i = 0; i< mycart.size(); i++){
        cart ct = (cart)mycart.get(i);
        Product item = EShop.getItem(ct.getSku());
        total += item.getPrice() * ct.getCount();
      %>
    <tr>
    <td><%= ct.getCount() %></td>
    <td><%= item.getName() %></td>
    <td align="right">
    <%= dispPrice(String.valueOf(item.getPrice())) %></td>
    <td align="right">
    <%= dispPrice(String.valueOf(item.getPrice() * ct.getCount()))
%></td> <td>
      <a href="<%= request.getContextPath() + CART_PAGE +
        "?action=add&&sku=" + ct.getSku() %>">
      <b>Add 1</b></a> </td>
    <td > <a href="<%= request.getContextPath() + CART_PAGE +
        "?action=remove&sku=" + ct.getSku() %>">
      <b>Remove 1</b></a>
    </td></tr>
  <tr><%} %>
  <td colspan="3" align="right">Total</td>
  <td colspan="2" align="left"><%= dispPrice(String.valueOf(total))
%></td> </tr> <tr>
  <td colspan="6">
    <a href="<%= request.getContextPath() + CART_PAGE +
      "?action=clear&sku=0" %>">
    Clear Shopping Cart</a>
  </td></tr> <%} %> <tr>
  <td colspan="6">

```

```
<a href="<%= request.getContextPath() + SHOP_PAGE %>">
  Return to Shopping</a>
</td></tr></table></td></tr>
</table>
</body>
</html>
```

2.3 登录界面的设计

1) 登录功能是常见的Web 功能点，在很多Web 应用中都有应用。一般的登录验证主要通过用户名、密码两个属性来进行用户身份验证。进行用户身份验证的方法有很多，本次实验采用最为简单的一种方式来进行认证。

2) 将下面的代码6.1 保存为login.html 作为登录页面(建议将本次实验的所有jsp 文件保存在同一目录下)。登录页面有三个表单项，分别是用来表示用户登录id的username，登录状态的status，以及用来表示用户密码的password。

Login.html源文件如下

```
<html>
  <head>
    <title>login.html</title>
  </head>
  <body onload="javascript:document.f1.username.focus();">
    <form name="f1" id="f1" action="servlet/usercheck" method="post">
      <table border="0">
        <tr>
          <td>Username:</td>
          <td><input type="text" name="username" id="username"></td>
        </tr>
        <tr>
          <td>Password:</td>
          <td><input type="password" name="password" id="password">
            <input type="hidden" name="status" id="status"
value="login"></td>
        </tr>
        <tr>
          <td colspan="1" align="right"><input type="submit"
value="Login"></td>
          <td colspan="2" align="center"><input type="reset"
value="Reset"></td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

3. Controller的设计

用户的登录和注销操作，以及购物过程的操作都由servlet实现，在项目下添加包servlet，并在此包中添加两个servlet: usercheck, sessioncart
Servlet的doGet方法直接调用doPost方法，一下具体实现他们的doPost方法

3.1 用户验证的servlet设计

为了区别用户登录和注销，需要除用户名和密码之外的第三个参数status, 此servlet调用JavaBean userinfo验证用户，具体实现如下

```
String username = request.getParameter("username");
String passwd = request.getParameter("password");
String status = request.getParameter("status");
if(status != null){
    if(status.equals("login")){
        estore.userinfo ui = new estore.userinfo(username,passwd);
        if (ui.islogin()){
            HttpSession session = request.getSession(true);
            session.setAttribute("username", ui.getUsername());
            response.sendRedirect("../index.jsp");
        }
        else
            response.sendRedirect("../login.html");
    }
    else{
        HttpSession session = request.getSession(true);
        session.invalidate();
        response.sendRedirect("../index.jsp");
    }
}
```

3.2 购物车操作的servlet

购物过程包括添加商品，移除商品，清空购物车操作，用户的购物车需要保存在session中，用于在servlet于JSP页面之间传递参数，具体实现如下

```
HttpSession session = request.getSession(true);
if(session.getAttribute("username") == null)
    response.sendRedirect("../login.html");

response.setContentType("text/html");
PrintWriter out = response.getWriter();
ArrayList mycart = (ArrayList)session.getAttribute("mycart");
String action = request.getParameter("action");
String sku = request.getParameter("sku");
if(action != null && sku != null){
    if(action.equals("add")){
        if(mycart != null){
```

```


        int flag = 0;
        for(int i =0; i < mycart.size(); i++){
            estore.cart tc = (estore.cart)mycart.get(i);
            if(tc.getSku().equals(sku)){
                flag = 1;
                int cc = tc.getCount();
                tc.setCount(cc+1);
                mycart.set(i, tc);
                break;
            }
        }
        if(flag == 0){
            mycart.add(new estore.cart(1,sku));
        }
    }
    else{
        mycart = new ArrayList();
        mycart.add(new estore.cart(1,sku));
    }
    session.setAttribute("mycart", mycart);
}
else if(action.equals("remove")){
    if(mycart != null){
        int cc = 0, flag = -1;
        estore.cart tc = new estore.cart();
        for(int i =0; i < mycart.size(); i++){
            tc = (estore.cart)mycart.get(i);
            if(tc.getSku().equals(sku)){
                cc = tc.getCount();
                flag = i;
                break;
            }
        }
        if(flag != -1){
            if(cc > 1){
                tc.setCount(cc-1);
                mycart.set(flag, tc);
            }
            else
                mycart.remove(flag);
        }
        if(mycart.size() == 0)
            session.removeAttribute("mycart");
    }
    else

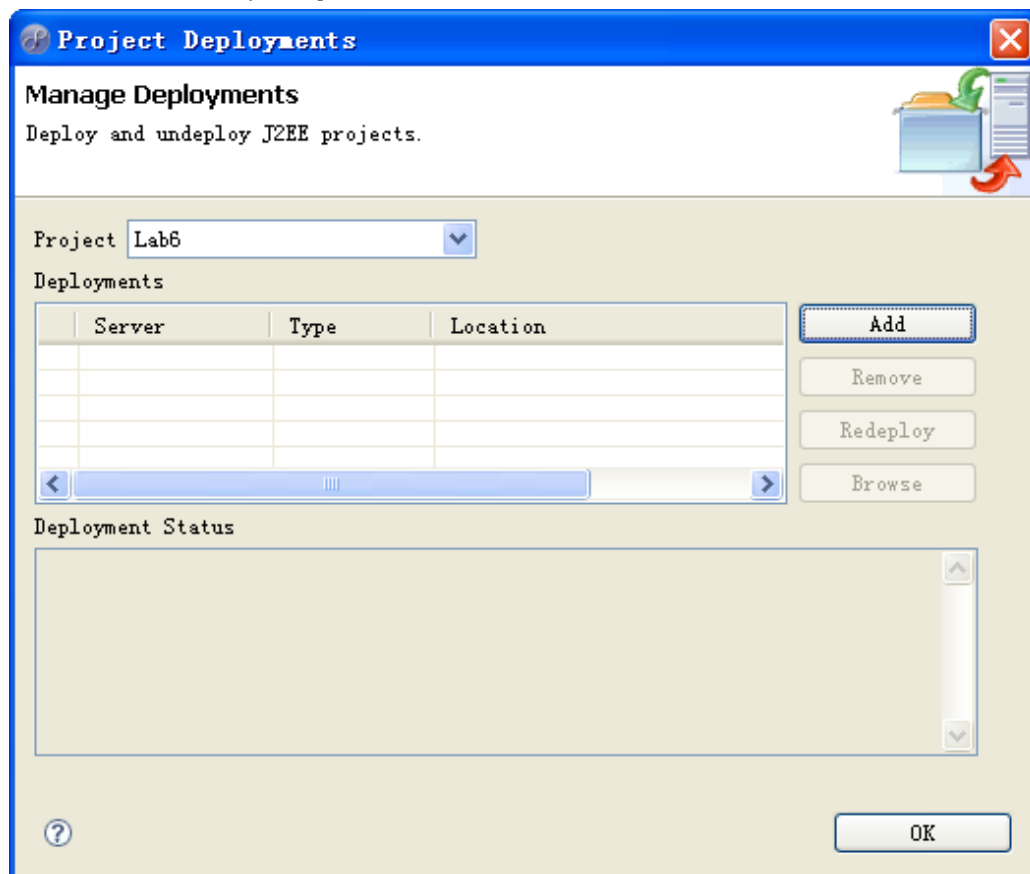
```



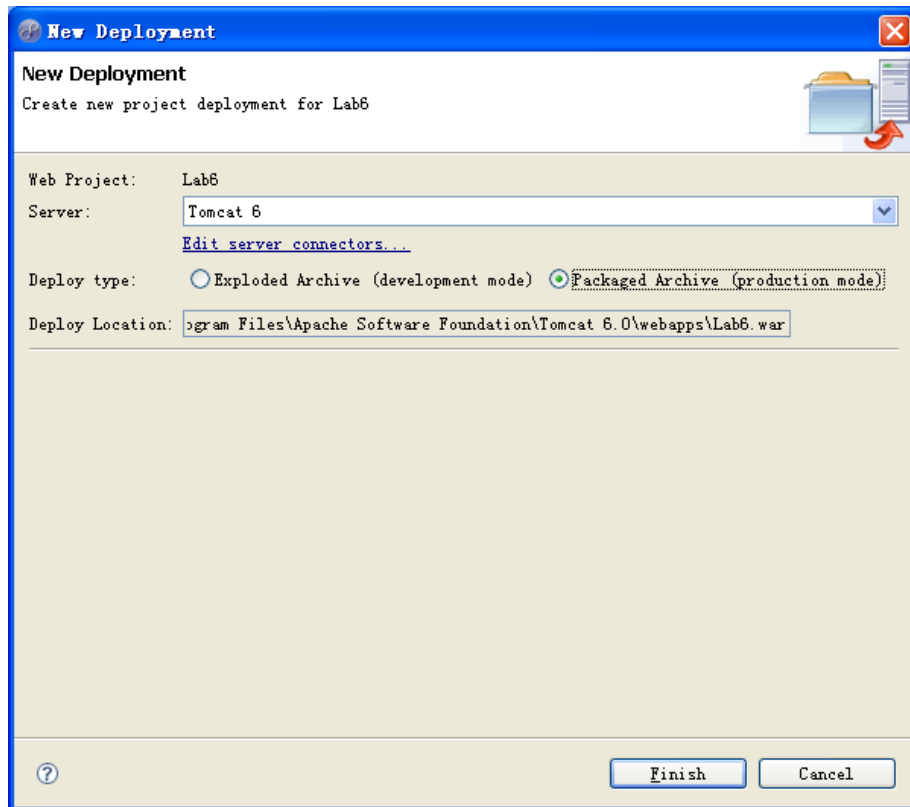
```
session.setAttribute("mycart",mycart);
    }
}
else if(action.equals("clear")){
    if(mycart != null && mycart.size() > 0){
        mycart.clear();
        session.removeAttribute("mycart");
    }
}
}
response.sendRedirect("../shopcart.jsp");
```

4. 项目部署

编辑完成后，点击MyEclipse的按钮弹出如下窗口



选定当前项目，点击“Add”，进入如下窗口



在Server项选择Tomcat，

Deploy type 选择 Packaged Archive (product mode)

点击Finish完成部署，此时在Tomcat的Webapp目录下可以看到有Lab6.war文件存在，此文件是当前项目的压缩包，包含各个页面以及Java文件生成的class文件。可以通过TOMCAT的<http://localhost:8080/manager/html>动态部署WAR文件

部署完成后再浏览器敲入地址：<http://localhost:8080/Lab6/>