

## 知识点列表

编号	名称	描述	级别
1	【案例 1】session 验证 02	在 main 页面增加退出功能。	***
2	验证码	掌握 Servlet 产生图片验证码的写法	**
3	购物车	熟练掌握购物车练习，通过购物车练习对之前的知识进行综合应用	**

注：   \*\*"理解级别   \*\*\*"掌握级别   \*\*\*\*"应用级别

## 目录

1. 【案例 1】 session 验证 02* .....	错误！未定义书签。
1.1. 验证码 .....	4
1.1.1. 【案例 2】 Servlet 生成验证码图片** .....	7
1.1.2. 【案例 3】 验证码应用** .....	11
1.2. 购物车 .....	26
1.2.1. 【案例 4】 购物车** .....	26

## 1. 【案例 1】session 验证 02 \*\*\*

### 案例描述

在 main 页面增加退出功能。

### 参考代码

1) 请下载 web06\_login.zip

2) 修改 main.jsp

当用户点击“退出”后，session 被删除，重定向到登陆页面

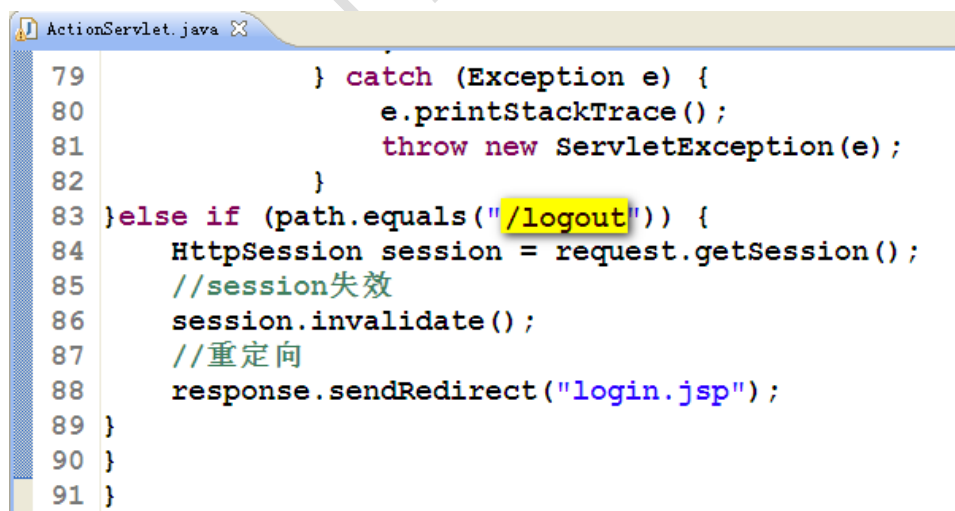


```

1  <%@page pageEncoding="utf-8"
2  contentType="text/html; charset=utf-8" %>
3  <%
4      //session验证
5      Object obj = session.getAttribute("user");
6      if(obj == null){
7          response.sendRedirect("login.jsp");
8          return;
9      }
10 %>
11 <h1>主功能页面</h1>
12 <a href="logout.do">退出</a>
13

```

3) 修改 ActionServlet



```

79         } catch (Exception e) {
80             e.printStackTrace();
81             throw new ServletException(e);
82         }
83     } else if (path.equals("/logout")) {
84         HttpSession session = request.getSession();
85         //session失效
86         session.invalidate();
87         //重定向
88         response.sendRedirect("login.jsp");
89     }
90 }
91 }

```

4) 重新部署

5) 在地址栏输入 localhost:8080/web06\_login/main.jsp

将跳转到登陆页面

## 1. 验证码 \*\*

第一步：填写账户信息 以下均为必填

会员名：

登录密码：

确认密码：

验证码：   看不清? [换一张](#) 

[同意以下协议并 注册](#)

[《淘宝服务协议》](#)

### 小知识

#### 如何写 Java 程序访问 web 应用？

Apache 提供了一个工具类 [http-client.jar](#)，封装了 http 协议，它提供一些接口及操作可以访问 web 应用。可到 Apache 官方下载。

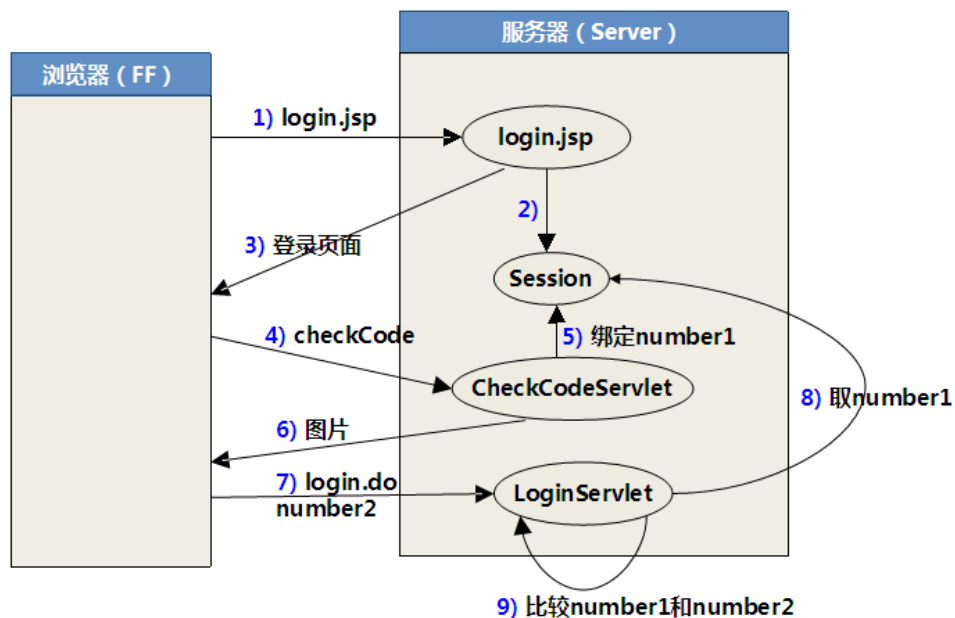
### 图示演示

#### 验证码流程图

服务器部署某个应用，该应用添加了 session 验证，必须登录才能访问。

- 1) 浏览器向 login.jsp 发送请求，请求获得登录页面
- 2) jsp 会创建一个 [session 对象](#)
- 3) Jsp 返回给浏览器一个登陆页面（html、js、css），  
该页面包含一行代码 ``（checkCode 是一个请求地址）
- 4) 浏览器立即给服务器发请求 checkCode，访问到 CheckCodeServlet
- 5) CheckCodeServlet 会生成一个随机数（验证数字），并绑定到 Session 对象上；
- 6) 同时，CheckCodeServlet 发送给浏览器一张图片（验证数字+图片）  
此时，用户在浏览器上看到完全的登录页面：表单
- 7) 用户填写完表单后提交，访问 LoginServlet
- 8) LoginServlet 从 Session 对象中取到验证数字 number1
- 9) LoginServlet 将 number1 与用户提交请求中的验证数字 number2 比较，如果匹配，则继续

正常的登录验证，不匹配则返回给浏览器信息“验证码不匹配”



### 案例演示



#### 1) 新建 test.html

```

test.html x  CheckcodeServlet.java
1 <html>
2 <head>
3 <meta http-equiv="Content-Type"
4   content="text/html; charset=UTF-8">
5 <title>Insert title here</title>
6 </head>
7 <body style="font-size:30px;">
8   hello<br/>
9   
10 </body>
11 </html>

```

#### 2) 新建 CheckCodeServlet

```

1 package web;
2
3 import java.awt.Color;
4
17
18 public class CheckcodeServlet extends HttpServlet {
19     public void service(
20         HttpServletRequest request,
21         HttpServletResponse response)
22         throws ServletException, IOException {
23         System.out.println("service...");
24     }
25 }

```

### 3) web.xml

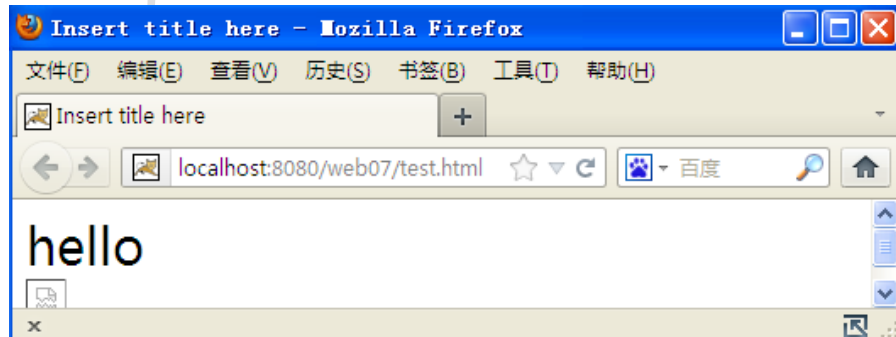
```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app version="2.4"
3     xmlns="http://java.sun.com/xml/ns/j2ee"
4     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5     xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
6     http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
7 >
8     <servlet>
9         <servlet-name>CheckcodeServlet</servlet-name>
10        <servlet-class>web.CheckcodeServlet</servlet-class>
11    </servlet>
12
13    <servlet-mapping>
14        <servlet-name>CheckcodeServlet</servlet-name>
15        <url-pattern>/checkcode</url-pattern>
16    </servlet-mapping>
17 </web-app>
18

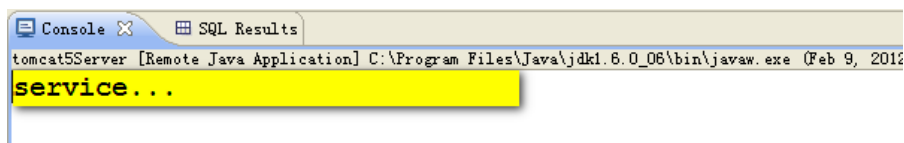
```

### 4) 部署项目

### 5) 访问 <http://localhost:8080/web07/test.html>



服务器立即打印 “service....”



说明浏览器访问页面 test.html 时，在执行语句时向服务器发送了 checkcode 请求

## 【案例 2】Servlet 生成验证码图片 \*\*

### 案例描述

通过 Servlet 生成一张图片返回给浏览器（图片是一张压缩为 jpg 格式的随机数图片）

### 小知识

以 com.sun 开头的 jar 包是 JDK 的扩展包，com.sun.image.codec.jpeg.JPEGCodec 是用于图片压缩的类。

### 参考代码

#### 1) CheckcodeServlet

```
package web;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.image.BufferedImage;
import java.io.IOException;
import java.io.OutputStream;
import java.util.Random;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.sun.image.codec.jpeg.JPEGCodec;
import com.sun.image.codec.jpeg.JPEGImageEncoder;

public class CheckcodeServlet extends HttpServlet {
```

```
public void service(HttpServletRequest request,
    HttpServletResponse response)
    throws ServletException, IOException {
    System.out.println("service...");

    /* 1 设置服务器返回文件的格式为 jpg 格式的图片 */
    //常见的还有：text/html 和 text/xml
    response.setContentType("image/jpeg");

    /* 2 画图 */
    //BufferedImage:内存映象对象
    BufferedImage image =
        new BufferedImage(60, 20,
            BufferedImage.TYPE_INT_RGB);
    Random r = new Random();
    //获得画笔
    Graphics g = image.getGraphics();
    //设置背景颜色
    g.setColor(new Color(
        r.nextInt(255),
        r.nextInt(255),
        r.nextInt(255)));
    //填充背景颜色
    g.fillRect(0, 0, 60, 20);
    //设置前景颜色
    g.setColor(new Color(0,0,0));
    //生成随机数
    String number = String.valueOf(r.nextInt(99999));
    //画到图片上
    g.drawString(number, 5, 15);

    /* 3 压缩图片并输出 */
    //获得字节输出流，因为要输出的是图像压缩之后
    //的字节数组，所以，不能用 PrintWriter。
    OutputStream os = response.getOutputStream();
    //将图片压缩，输出
    JPEGImageEncoder encoder =
```



```
        JPEGCodec.createJPEGEncoder(os);
        encoder.encode(image);
    }
}
```

## 2) web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.4"
    xmlns="http://java.sun.com/xml/ns/j2ee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
    http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
    <servlet>
        <servlet-name>CheckcodeServlet</servlet-name>
        <servlet-class>web.CheckcodeServlet</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>CheckcodeServlet</servlet-name>
        <url-pattern>/checkcode</url-pattern>
    </servlet-mapping>

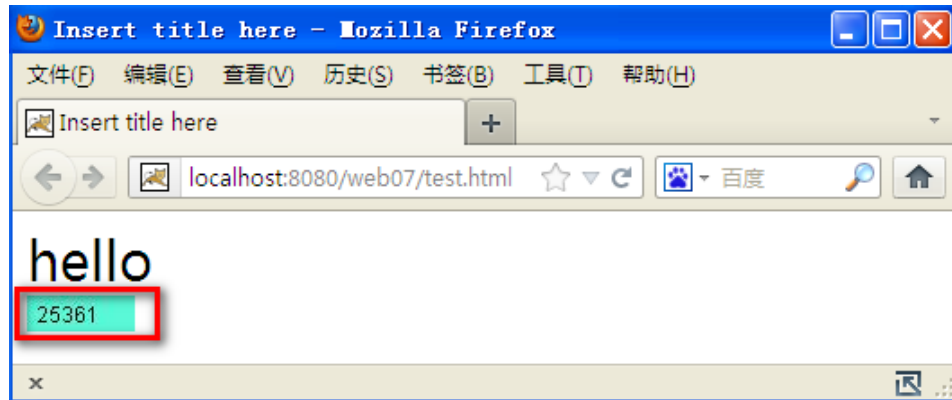
</web-app>
```

## 3) test.html

```
<html>
<head>
<meta http-equiv="Content-Type"
    content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body style="font-size:30px;">
    hello<br/>
    
</body>
</html>
```

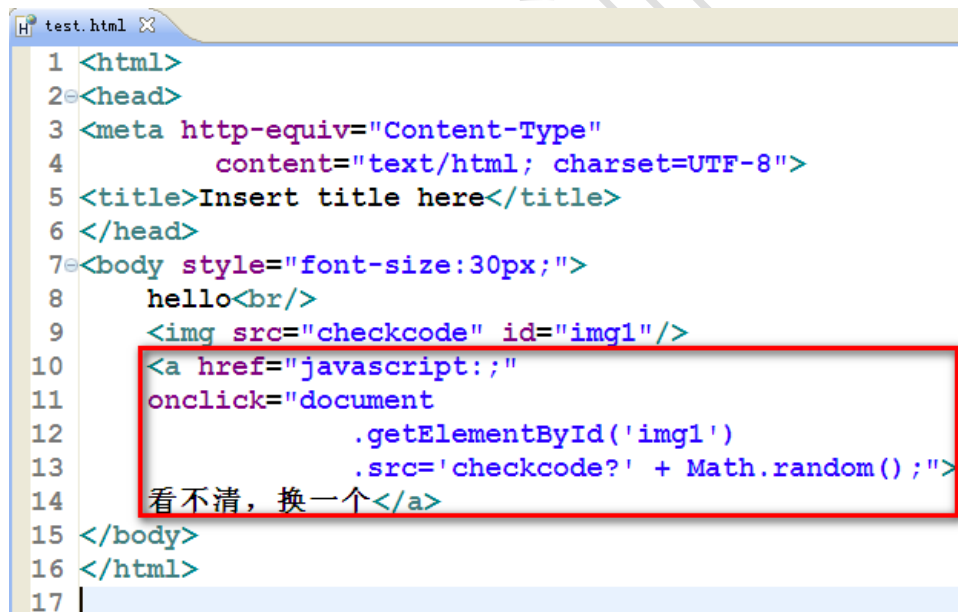
#### 4) 访问 <http://localhost:8080/web07/test.html>

每次刷新会生成一个新的随机数



#### ● 增加更换验证码功能

##### 5) 修改 test.html



#### 6) 访问 <http://localhost:8080/web07/test.html>



### 【案例 3】验证码应用 \*\*

#### 案例描述

为用户注册页面加入验证码

1) 请下载 web06\_login2.zip

2) 修改 regist.jsp

添加验证码

```
<%@page pageEncoding="utf-8"
contentType="text/html;charset=utf-8" %>
<html>
  <head>
    <title>regist</title>
    <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
    <link rel="stylesheet" type="text/css" href="css/style.css" />
  </head>
  <body>
    <div id="wrap">
      <div id="top_content">
        <div id="header">
          <div id="righthead">
            <p>
              2009/11/20
            <br />
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

```

        </p>
    </div>
    <div id="topheader">
        <h1 id="title">
            <a href="#">main</a>
        </h1>
    </div>
    <div id="navigation">
    </div>
</div>
<div id="content">
    <p id="whereami">
    </p>
    <h1>
        注册
    </h1>
    <form action="regist.do" method="post">
        <table cellpadding="0" cellspacing="0" border="0"
            class="form_table">
            <tr>
                <td valign="middle" align="right">
                    用户名:
                </td>
                <td valign="middle" align="left">
                    <input type="text" class="inputgri"
                        name="username" />
                    <%
                        String msg =
                            (String)request.getAttribute("regist_error");
                    %>
                    <span style="color:red;font-style:italic;">
                        <%= (msg == null ? "" : msg) %>
                    </span>
                </td>
            </tr>
            <tr>
                <td valign="middle" align="right">
                    真实姓名:

```

```

        </td>
        <td valign="middle" align="left">
            <input type="text" class="inputgri" name="name" />
        </td>
    </tr>
    <tr>
        <td valign="middle" align="right">
            密码:
        </td>
        <td valign="middle" align="left">
            <input type="password" class="inputgri" name="pwd" />
        </td>
    </tr>
    <tr>
        <td valign="middle" align="right">
            性别:
        </td>
        <td valign="middle" align="left">
            男
            <input type="radio" class="inputgri"
                name="gender" value="m" checked="checked" />
            女
            <input type="radio" class="inputgri"
                name="gender" value="f" />
        </td>
    </tr>
    <tr>
        <td valign="middle" align="right">
            验证码:
            
            <a href="javascript:;"
                onclick="document.getElementById('img1').src
                    = 'checkcode?' + Math.random();" >换一张</a>
        </td>
        <td valign="middle" align="left">
            <input type="text" class="inputgri"
                name="number" />
        </td>
    </tr>

```

```

        </td>
    </tr>
</table>
<p>
    <input type="submit" class="button"
    value="确认" />
</p>
</form>

    </div>
</div>
<div id="footer">
    <div id="footer_bg">
        ABC@126.com
    </div>
</div>
</div>
</body>
</html>

```

### 3) 拷贝 CheckedServlet 到项目下

```

package web;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.image.BufferedImage;
import java.io.IOException;
import java.io.OutputStream;
import java.util.Random;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import com.sun.image.codec.jpeg.JPEGCodec;
import com.sun.image.codec.jpeg.JPEGImageEncoder;

```

```
public class CheckcodeServlet extends HttpServlet {

    public void service(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        System.out.println("service...");
        //text/html;
        //text/xml;
        response.setContentType("image/jpeg");
        //BufferedImage:内存映象对象
        BufferedImage image =
            new BufferedImage(60, 20,
                BufferedImage.TYPE_INT_RGB);
        Random r = new Random();
        //获得画笔
        Graphics g = image.getGraphics();
        g.setColor(
            new Color(r.nextInt(255),
                r.nextInt(255), r.nextInt(255)));
        g.fillRect(0, 0, 60, 20);
        g.setColor(new Color(0,0,0));
        String number = String.
            valueOf(r.nextInt(99999));
        g.drawString(number, 5, 15);

        //获得字节输出流，因为要输出的是图像压缩之后
        //的字节数组，所以，不能用 PrintWriter。
        OutputStream os = response.getOutputStream();
        //将图片压缩，输出。
        JPEGImageEncoder encoder =
            JPEGCodec.createJPEGEncoder(os);
        encoder.encode(image);
    }

}
```

#### 4) 修改 web.xml



```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
5     http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
6   <servlet>
7     <servlet-name>ActionServlet</servlet-name>
8     <servlet-class>web.ActionServlet</servlet-class>
9   </servlet>
10  <servlet>
11    <servlet-name>CheckcodeServlet</servlet-name>
12    <servlet-class>web.CheckcodeServlet</servlet-class>
13  </servlet>
14
15  <servlet-mapping>
16    <servlet-name>CheckcodeServlet</servlet-name>
17    <url-pattern>/checkcode</url-pattern>
18  </servlet-mapping>
19  <servlet-mapping>
20    <servlet-name>ActionServlet</servlet-name>
21    <url-pattern>*.do</url-pattern>
22  </servlet-mapping>
23
24 </web-app>

```

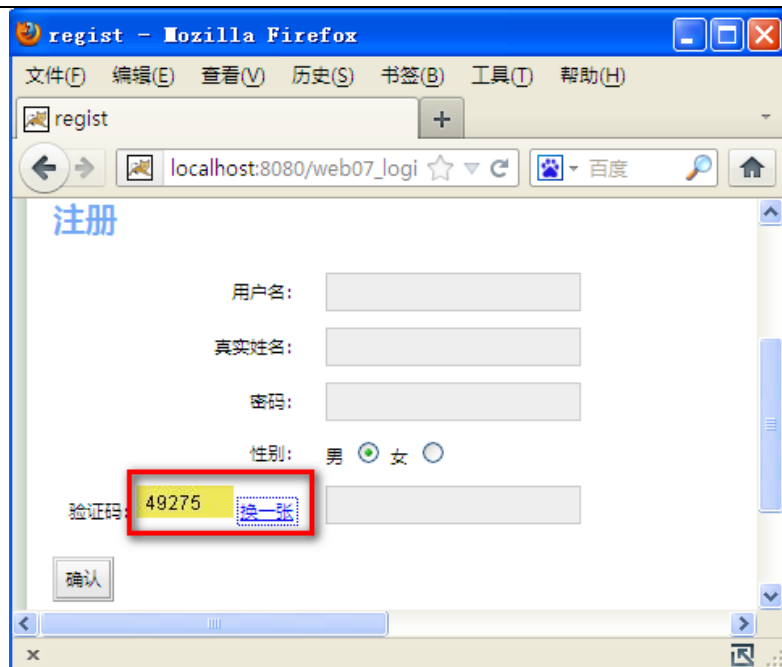
#### 5) 部署应用

#### 6) 测试

访问 [http://localhost:8080/web07\\_login/regist.jsp](http://localhost:8080/web07_login/regist.jsp)

点击“换一张”能更换验证码





- 注册验证

- 7) 修改 CheckcodeServlet

绑定生成的验证数字到 Session 对象上

注意：给用户看的是图片+验证数字；放入 Session 对象的是验证数字

```

CheckcodeServlet.java
37         r.nextInt(255), r.nextInt(255));
38     g.fillRect(0, 0, 60, 20);
39     g.setColor(new Color(0,0,0));
40     String number = String.
41     valueOf(r.nextInt(99999));
42     HttpSession session = request.getSession();
43     session.setAttribute("number", number);
44     g.drawString(number, 5, 15);
45
46     //获得字节输出流，因为要输出的是图像压缩之后
47     //的字节数组，所以，不能用PrintWriter。
48     OutputStream os = response.getOutputStream();
49     //将图片压缩，输出。
50     JPEGImageEncoder encoder =
51         JPEGCodec.createJPEGEncoder(os);
52     encoder.encode(image);
53 }
54 }
    
```

## 8) 修改 ActionServlet

添加“在注册时比较用户输入验证数字和 Session 中取出的验证数字”的代码

```
package web;

import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import util.DBUtil;
import dao.UserDAO;
import entity.User;

public class ActionServlet extends HttpServlet {

    public void service(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        request.setCharacterEncoding("utf-8");
        String uri = request.getRequestURI();
        String path = uri.substring(
            uri.lastIndexOf("/"), uri.lastIndexOf("."));

        if (path.equals("/regist")) {
            System.out.println("regist...");
            /* 1 先比较验证码 */
            //1.1 表单提交的验证码
            String number1 = request.getParameter("number");
            //1.2 session 中绑定的验证码
            HttpSession session = request.getSession();
            String number2 =
```

```
(String) session.getAttribute("number");

//用于测试
System.out.println("number1:" + number1);
System.out.println("number2:" + number2);

//当用户禁用 cookie，session 将失效，number2 为 null
if (number2 != null &&
    number2.equals(number1)) {

    // 正常注册
    String username =
        request.getParameter("username");
    UserDao dao = new UserDao();
    try {
        User user = dao.findByUsername(username);
        if (user == null) {
            // 用户名不存在，插入操作
            user = new User();
            user.setName(request.getParameter("name"));
            user.setPwd(request.getParameter("pwd"));
            user.setGendar(request.getParameter("gendar"));
            user.setUsername(username);
            dao.save(user);
            response.sendRedirect("login.jsp");
        } else {
            // 用户名存在，则提示用户
            request.setAttribute("regist_error", "用户名已经存在");
            request.getRequestDispatcher("regist.jsp").forward(
                request, response);
        }
    } catch (Exception e) {
        e.printStackTrace();
        throw new ServletException(e);
    }
} else {
    // 验证码错误
    request.setAttribute("checkcode_error",
        "验证码错误");
    request.getRequestDispatcher("regist.jsp")
```

```

        .forward(request,response);
    }
} else if (path.equals("/login")) {
    //比较验证码
    String username = request.getParameter("username");
    String pwd = request.getParameter("pwd");
    UserDao dao = new UserDao();
    try {
        User user = dao.findByUsername(username);
        if (user != null && user.getPwd().equals(pwd)) {
            // 登录成功
            // 向 session 中绑定数据
            HttpSession session = request.getSession();
            System.out.println("sessionId:" + session.getId());
            session.setAttribute("user", user);
            response.sendRedirect("main.jsp");
        } else {
            // 登录失败
            request.setAttribute("login_error", "用户名或者密码错误");
        }
    } catch (Exception e) {
        e.printStackTrace();
        throw new ServletException(e);
    }
} else if (path.equals("/logout")) {
    HttpSession session = request.getSession();
    session.invalidate();
    response.sendRedirect("login.jsp");
}

}

}

```

## 9) 修改 regist.jsp

### 加入验证失败的提示

```
<%@page pageEncoding="utf-8"
contentType="text/html;charset=utf-8" %>
<html>
  <head>
    <title>regist</title>
    <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
    <link rel="stylesheet" type="text/css" href="css/style.css" />
  </head>
  <body>
    <div id="wrap">
      <div id="top_content">
        <div id="header">
          <div id="righthead">
            <p>
              2009/11/20
            <br />
          </p>
        </div>
        <div id="topheader">
          <h1 id="title">
            <a href="#">main</a>
          </h1>
        </div>
        <div id="navigation">
        </div>
      </div>
      <div id="content">
        <p id="whereami">
        </p>
        <h1>
          注册
        </h1>
        <form action="regist.do" method="post">
          <table cellpadding="0" cellspacing="0" border="0"
            class="form_table">
```

```

<tr>
    <td valign="middle" align="right">
        用户名:
    </td>
    <td valign="middle" align="left">
        <input type="text" class="inputgri"
            name="username" />
        <%
            String msg =
            (String)request.getAttribute("regist_error");
        %>
        <span style="color:red;font-style:italic;">
            <%= (msg == null ? "" : msg) %>
        </span>
    </td>
</tr>
<tr>
    <td valign="middle" align="right">
        真实姓名:
    </td>
    <td valign="middle" align="left">
        <input type="text" class="inputgri" name="name" />
    </td>
</tr>
<tr>
    <td valign="middle" align="right">
        密码:
    </td>
    <td valign="middle" align="left">
        <input type="password" class="inputgri" name="pwd" />
    </td>
</tr>
<tr>
    <td valign="middle" align="right">
        性别:
    </td>
    <td valign="middle" align="left">
        男

```

```

        <input type="radio" class="inputgri"
            name="gendar" value="m" checked="checked"/>
        女
        <input type="radio" class="inputgri"
            name="gendar" value="f"/>
    </td>
</tr>

<tr>
    <td valign="middle" align="right">
        验证码:
        
        <a href="javascript:;"
            onclick="document.getElementById('img1').src
                = 'checkcode?' + Math.random();" >换一张</a>
    </td>
    <td valign="middle" align="left">
        <input type="text" class="inputgri"
            name="number" />
        <%
            String msg2 =
                (String)request.getAttribute("checkcode_error");
        %>
        <span style="color:red;font-style:italic;">
            <%= (msg2 == null ? "" : msg2) %>
        </span>
    </td>
</tr>
</table>
<p>
    <input type="submit" class="button"
        value="确认" />
</p>
</form>

</div>
</div>
<div id="footer">
    <div id="footer_bg">

```

```

        ABC@126.com
    </div>
</div>
</div>
</body>
</html>

```

## 10) 重新部署

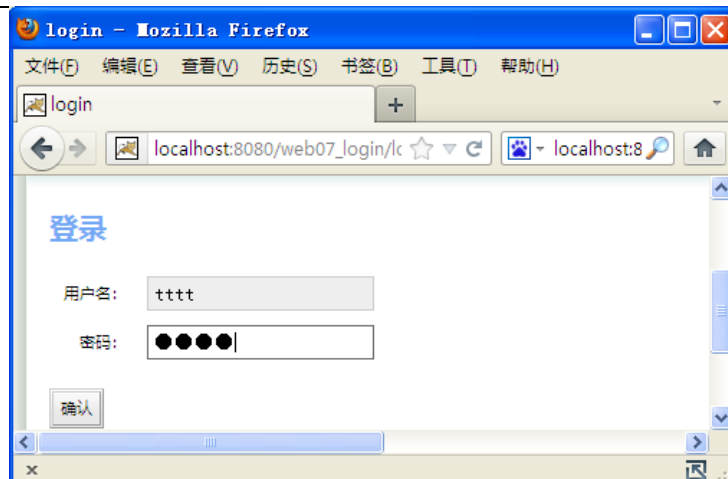
## 11) 测试

访问 [http://localhost:8080/web07\\_login/regist.jsp](http://localhost:8080/web07_login/regist.jsp)



## 12) 注册成功，跳入登录页面



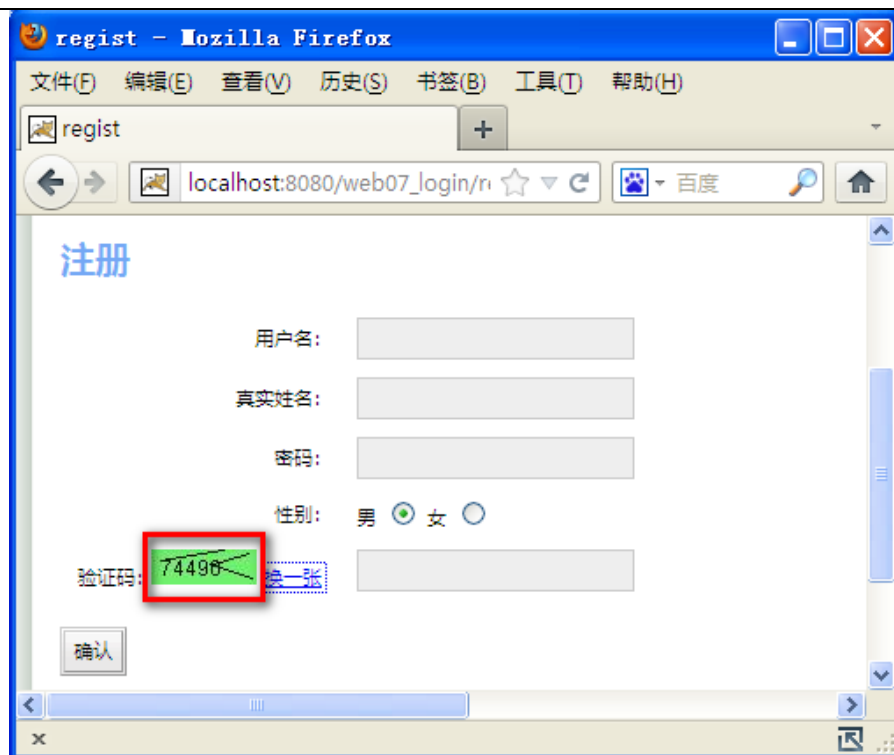


- 加干扰线或干扰点（防止机器人识别）

### 13) 修改 CheckcodeServlet

```
CheckcodeServlet.java
41         r.nextInt(255),
42         r.nextInt(255));
43     //填充背景颜色
44     g.fillRect(0, 0, 60, 20);
45     //设置前景颜色
46     g.setColor(new Color(0,0,0));
47
48     //划线线(起干扰识别的作用)
49     g.drawLine(r.nextInt(60), r.nextInt(20),
50               r.nextInt(60), r.nextInt(20));
51     g.drawLine(r.nextInt(60), r.nextInt(20),
52               r.nextInt(60), r.nextInt(20));
53     g.drawLine(r.nextInt(60), r.nextInt(20),
54               r.nextInt(60), r.nextInt(20));
55
56     //生成随机数
57     String number = String.valueOf(r.nextInt(9));
58     //画到图片上
59     g.drawString(number, 5, 15);
```

### 14) 访问 [http://localhost:8080/web07\\_login/regist.jsp](http://localhost:8080/web07_login/regist.jsp)



## 2. 购物车 \*\*

### 【案例 4】购物车 \*\*

- 功能 \*\*
  - a. 商品列表
  - b. 购买商品
  - c. 查看购物车
  - d. 删除购物车当中的商品
  - e. 修改购物车当中的商品的数量
  - f. 删除购物车中的所有商品
  - g. 购物车商品总价
- 实现 \*\*

## 实现 1：商品列表 \*\*

### 实现步骤

#### step1 建表

```
SQL> create table t_computer(  
    id bigint primary key auto_increment,  
    model varchar(20),  
    pic varchar(50),  
    prodDesc text,  
    price double);
```

#### step2 插入记录

```
SQL> insert into t_computer(model,pic,prodDesc,price)  
    values('x200','x200.jpg','还算便宜',2000);  
SQL> insert into t_computer(model,pic,prodDesc,price)  
    values('x500','x500.jpg','性价比最好',4000);  
SQL> insert into t_computer(model,pic,prodDesc,price)  
    values('x600','x600.jpg','性能不错',6000);
```

#### step3 Computer 实体类

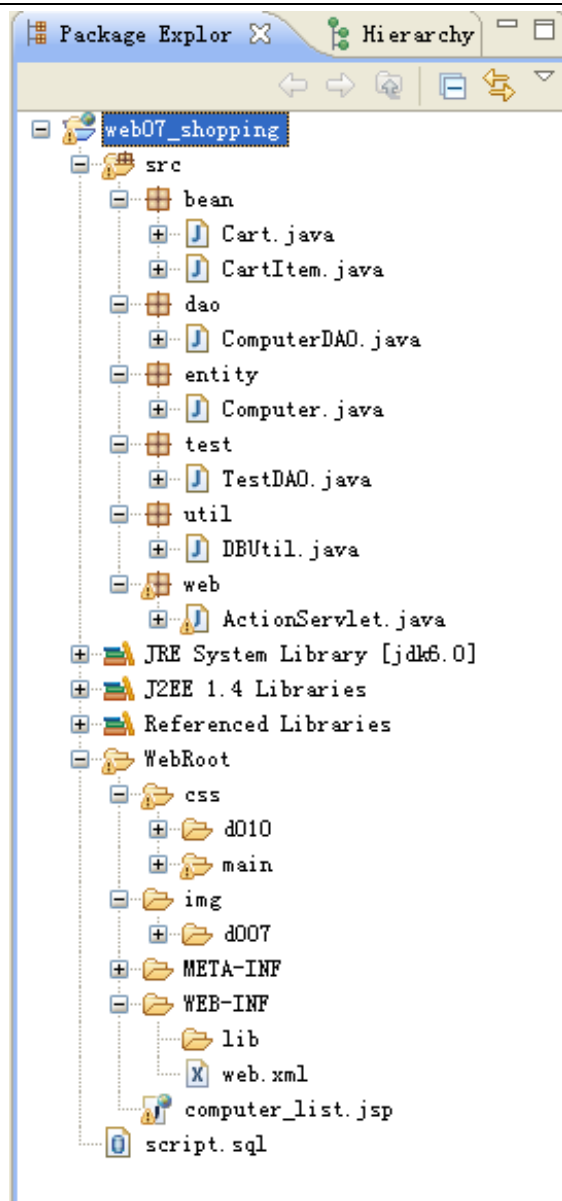
#### step4 ComputerDAO 类

#### step5 测试 ComputerDAO

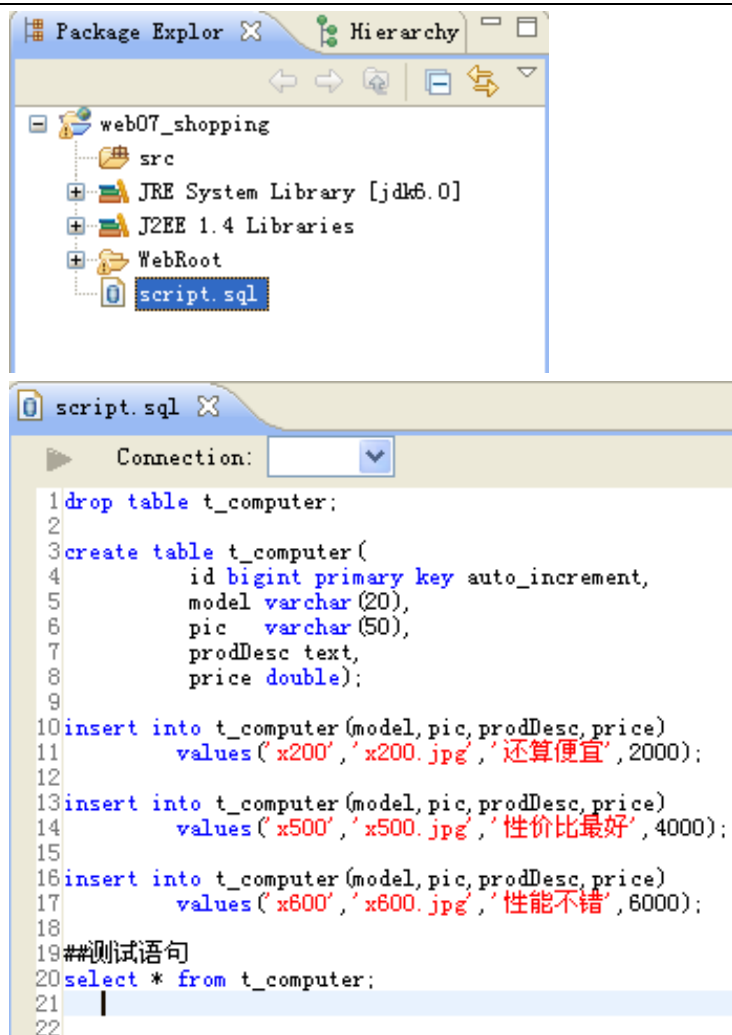
#### step6 ActionServlet

#### step7 computer\_list.jsp

### 项目结构预览



- 建表&&准备数据
  - 1) 新建 web 工程 web07\_shopping
  - 2) 新建 script.sql



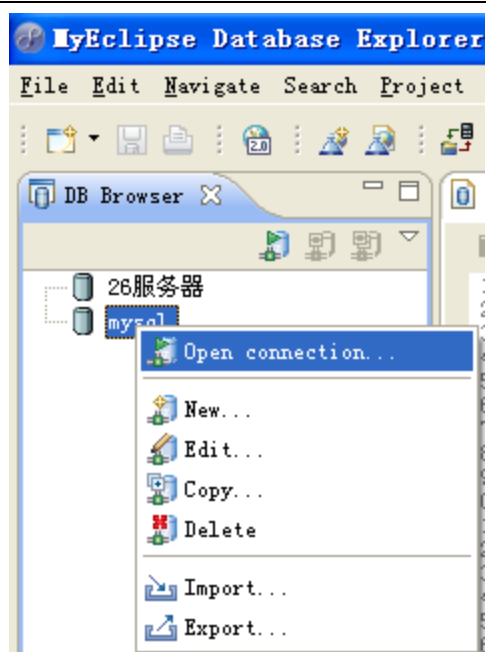
### 3) 启动 MyEclipse 数据库连接&&执行数据库脚本 script.sql

之前配置的 mysql 数据库，请参考 servlet 第 2 天课堂笔记

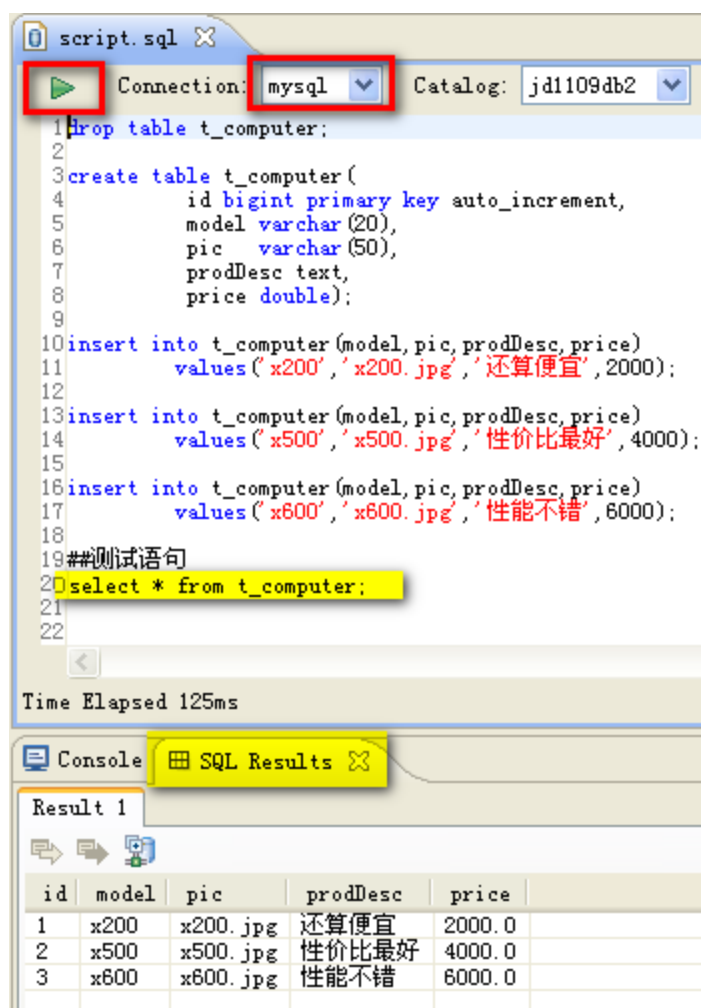
注意配置数据库连接时设置编码

jdbc:mysql://localhost:3306/jd1109db2?useUnicode=true&characterEncoding=utf8

#### a. 启动数据库连接



**b. 执行数据库脚本**



- 实体类&&DAO 操作类

#### 4) 根据数据库表新建实体类 Computer.java

```

package entity;

public class Computer {
    private long id;
    private String model;
    private String pic;
    private String prodDesc;
    private double price;

    public Computer() {}
    
```

```

public Computer(String model, String pic, String prodDesc, double price) {
    this.model = model;
    this.pic = pic;
    this.prodDesc = prodDesc;
    this.price = price;
}

public long getId() {return id;}
public void setId(long id) {this.id = id;}
public String getModel() {return model;}
public void setModel(String model) {this.model = model;}
public String getPic() {return pic;}
public void setPic(String pic) {this.pic = pic;}
public double getPrice() {return price;}
public void setPrice(double price) {this.price = price;}
public String getProdDesc() {return prodDesc;}
public void setProdDesc(String prodDesc) {this.prodDesc = prodDesc;}
}

```

## 5) 导入 jar 包

## 6) 拷贝 DBUtil.java

```

package util;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

/**
 * jdbc工具类，提供获得连接、关闭连接的方法
 * @author teacher
 *
 */
public class DBUtil {
    public static Connection getConnection()
        throws Exception{
        Connection conn = null;
        try {

```



```

        Class.forName("com.mysql.jdbc.Driver");
        conn = DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/jd1109db2?" +
            "useUnicode=true&characterEncoding=utf8",
            "root", "1234");
    } catch (Exception e) {
        e.printStackTrace();
        throw e;
    }
    return conn;
}

public static void close(Connection conn){
    if(conn!=null){
        try {
            conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}

public static void main(String[] args)
throws Exception {
    // TODO Auto-generated method stub
    System.out.println(getConnection());
}
}

```

## 7) 新建 DAO 方法 ComputerDAO

```

package dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;

```

```
import util.DBUtil;

import entity.Computer;

public class ComputerDAO {
    public List<Computer> findAll() throws Exception{
        List<Computer> computers =
            new ArrayList<Computer>();
        Connection conn =
            DBUtil.getConnection();
        PreparedStatement prep =
            conn.prepareStatement(
                "select * from t_computer");
        ResultSet rst =
            prep.executeQuery();
        while(rst.next()){
            Computer c =
                new Computer();
            c.setId(rst.getLong("id"));
            c.setModel(rst.getString("model"));
            c.setPic(rst.getString("pic"));
            c.setPrice(rst.getDouble("price"));
            c.setProdDesc(rst.getString("prodDesc"));
            computers.add(c);
        }
        DBUtil.close(conn);
        return computers;
    }

    public Computer findById(long id) throws Exception{
        Computer c = null;
        Connection conn =
            DBUtil.getConnection();
        PreparedStatement prep =
            conn.prepareStatement(
                "select * from t_computer where id=?");
        prep.setLong(1, id);
        ResultSet rst =
```

```

        prep.executeQuery();
        if(rst.next()){
            c = new Computer();
            c.setId(rst.getLong("id"));
            c.setModel(rst.getString("model"));
            c.setPic(rst.getString("pic"));
            c.setPrice(rst.getDouble("price"));
            c.setProdDesc(rst.getString("prodDesc"));
        }
        DBUtil.close(conn);
        return c;
    }
}

```

## 8) 测试 ComputerDAO 测试很重要!

### 新建 TestDAO.java

```

1 package test;
2
3 import dao.ComputerDAO;
4
5 public class TestDAO {
6
7     public static void main(String[] args)
8     throws Exception {
9         // TODO Auto-generated method stub
10        ComputerDAO dao = new ComputerDAO();
11        System.out.println(dao.findAll().size());
12    }
13 }
14

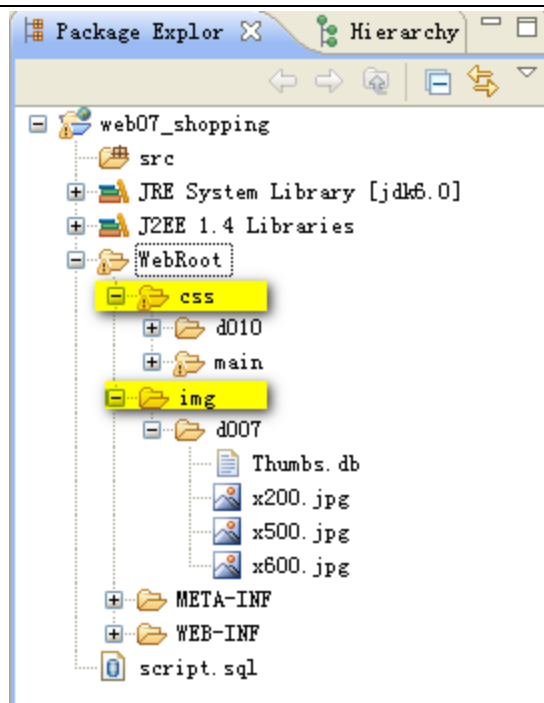
```

Console: <terminated> TestDAO (2) [Java Application] C:\Program Files\Java\jdk1.6.0\_06\bin\javaw.exe (Feb 10, 2010 3:00:00 PM)

## ● JSP&&Servlet

### 9) 拷贝 css&html 到项目下

请下载 shoppingcart\_demo.zip



10) 拷贝 computer\_list.html 到项目下，改名为 computer\_list.jsp

11) 新建 ActionServlet

添加查看页面功能

```
package web;

import java.io.IOException;
import java.util.List;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import bean.Cart;
import bean.CartItem;
import dao.ComputerDAO;
import entity.Computer;

public class ActionServlet extends HttpServlet {
```

```

public void service(HttpServletRequest request,
                    HttpServletResponse response)
    throws ServletException, IOException {
    String uri = request.getRequestURI();
    String path =
        uri.substring(uri.lastIndexOf("/"),
                      uri.lastIndexOf("."));
    if(path.equals("/list")){

        ComputerDAO dao =
            new ComputerDAO();
        try {
            List<Computer> computers =
                dao.findAll();
            request.setAttribute("computers", computers);
            request.getRequestDispatcher("computer_list.jsp")
                .forward(request, response);
        } catch (Exception e) {
            e.printStackTrace();
            throw new ServletException(e);
        }
    }
}

```

## 12) web.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.4"
    xmlns="http://java.sun.com/xml/ns/j2ee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
        http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
    <servlet>
        <servlet-name>ActionServlet</servlet-name>

```

```
<servlet-class>web.ActionServlet</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>ActionServlet</servlet-name>
  <url-pattern>*.do</url-pattern>
</servlet-mapping>
</web-app>
```

### 13) 修改 computer\_list.jsp

[illegible]

```

        </tbody>
    </table>
    <br />

    <table class="tableborder" cellSpacing="0"
        cellPadding="0" width="100%" border="0">
        <tbody>
            <tr class="header">
                <td class="altbg1" width="15%">
                    <b>型号</B>
                </td>
                <td class="altbg1" width="20%">
                    <b>产品图片</b>
                </td>
                <td class="altbg1" width="30%">
                    <b>产品说明</b>
                </td>

                <td class="altbg1" width="10%">
                    <b>产品报价</b>
                </td>
                <td class="altbg1">&quot;
                </td>
            </tr>
        </tbody>
    </table>
    <tbody>
        <%
            List<Computer> computers =
            (List<Computer>)request.getAttribute("computers");
            for(int i=0;i<computers.size();i++){
                Computer c = computers.get(i);
                %>
            <tr>
                <td class="altbg2">
                    &nbsp;&nbsp;&nbsp;<%=c.getModel()%>
                </td>
                <td class="altbg2">
                    
                                </td>
                                <td class="altbg2">
                                    <%=c.getProdDesc()%>
                                </td>
                                <td class="altbg2">
                                    ¥ <%=c.getPrice()%>
                                </td>
                                <td class="altbg2">
                                    <a href="">购买</a>
                                </td>
                            </tr>
                            <%=
                        }
                    %>
                </tbody>
            </table>

                                <br />
                                <center>
                                    <input class="button" type="button"
                                        value="查看购物车"
                                        name="settingsubmit" onclick="">
                                </center>
                            </td>
                        </tr>
                    </tbody>
                </table>
            </body>
        </html>

```

#### 14) 部署应用

#### 15) 测试

访问 [http://localhost:8080/web07\\_shopping/list.do](http://localhost:8080/web07_shopping/list.do)



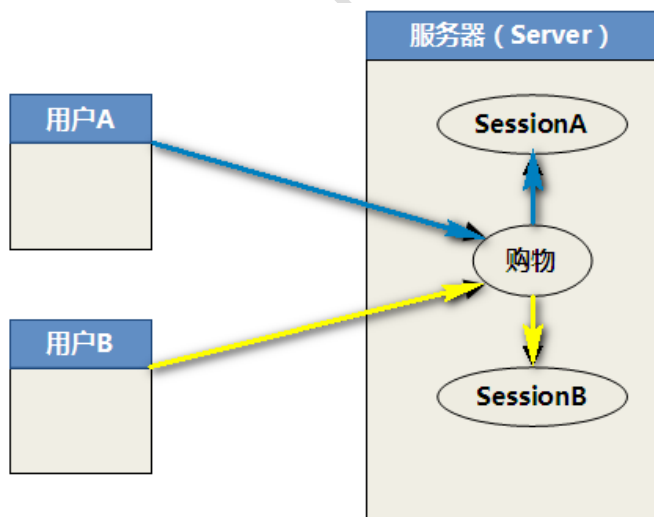


查看功能实现：)

实现 2：购买商品 \*\*

图示演示 1

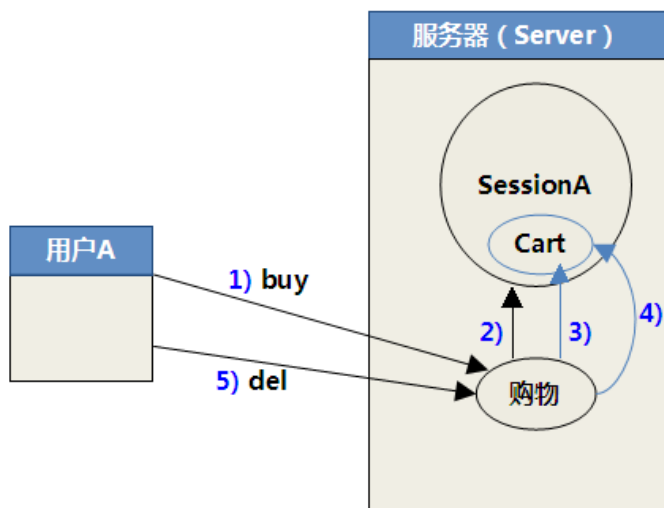
不同用户购买数据存放在不同 Session 中



图示演示 2

购买流程图

- 1) 用户发送请求 ( buy ) 购买
  - 2) 服务器创建 Session 对象
  - 3) 不再直接访问 session 的方法，而是绑定一个 Cart 对象
  - 4) 用户购买商品时，只调用 Cart 的 add 方法进行操作，而不再直接操作 Session
  - 5) 删除 ( 修改 ) 时也一样
- 如此，我们以后只要操作 Cart 类的方法就 OK 了，不再直接操作 Session。这是一个经典的设计。



### Cart 和 CartItem 伪代码

```

1 class CartItem{
2     private Computer c;
3     private int qty;
4 }
5
6 class Cart{
7     private List<CartItem> computers =
8     new ArrayList<CartItem>();
9
10    public void add(CartItem c){
11        computers.add(c);
12    }
13 }
14

```

### 实现步骤

#### 16) 新建 CartItem.java

商品条目类，作用是为了更方便实现购物车 Cart 的实现

```
package bean;
```

```
import entity.Computer;

/**
 * 商品条目类，其作用是，为了
 * 更方便实现Cart类的操作。
 * @author teacher
 *
 */
public class CartItem {
    private Computer c;
    private int qty;

    public Computer getC() {return c;}
    public void setC(Computer c) {this.c = c;}
    public int getQty() {return qty;}
    public void setQty(int qty) {this.qty = qty;}
}
```

## 17) 新建 Cart.java

```
package bean;

import java.util.ArrayList;
import java.util.List;

/**
 * 购物车
 * 实现了商品的添加、删除、列表、清空、总价
 * @author teacher
 *
 */
public class Cart {
    //items存放用户购买的所有商品
    private List<CartItem> items =
        new ArrayList<CartItem>();

    public boolean add(CartItem item){
        //先判断有没有购买过该商品，如果买过，
```

```

//则返回false，否则，添加到items集合。
for(int i=0;i<items.size();i++){
    CartItem curr = items.get(i);
    if(curr.getC().getId() == item.getC().getId()){
        //已经购买
        return false;
    }
}
items.add(item);
return true;
}

public List<CartItem> list(){
    return items;
}

public void delete(long id){
    for(int i=0;i<items.size();i++){
        CartItem curr = items.get(i);
        if(curr.getC().getId() == id){
            items.remove(curr);
            return;
        }
    }
}

public void clear(){
    items.clear();
}

public void modify(long id,int qty){
    for(int i=0;i<items.size();i++){
        CartItem curr = items.get(i);
        if(curr.getC().getId() == id){
            curr.setQty(qty);
            return;
        }
    }
}

```

```

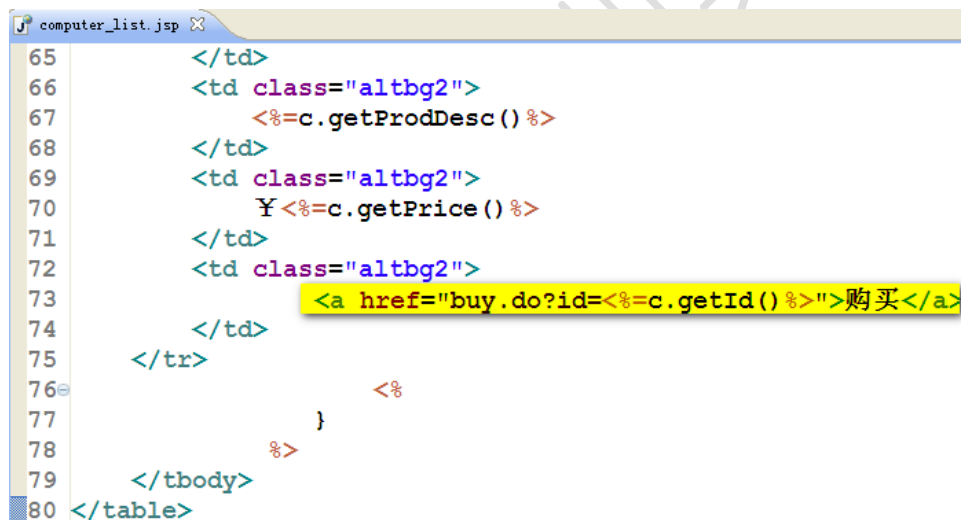
    }

    public double total(){
        double total = 0;
        for(int i=0;i<items.size();i++){
            CartItem curr = items.get(i);
            total += curr.getQty() * curr.getC().getPrice();
        }
        return total;
    }
}

```

## 18) 修改 computer\_list.jsp

添加购买的链接地址



```

65         </td>
66         <td class="altbg2">
67             <%=c.getProdDesc() %>
68         </td>
69         <td class="altbg2">
70             ￥<%=c.getPrice() %>
71         </td>
72         <td class="altbg2">
73             <a href="buy.do?id=<%=c.getId() %>">购买</a>
74         </td>
75     </tr>
76 </tbody>
77 </table>
78
79
80

```

## 19) 修改 ActionServlet

实现购买功能

```

package web;

import java.io.IOException;
import java.util.List;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;

```

```
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import bean.Cart;
import bean.CartItem;
import dao.ComputerDAO;
import entity.Computer;

public class ActionServlet extends HttpServlet {

    public void service(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        String uri = request.getRequestURI();
        String path =
            uri.substring(uri.lastIndexOf("/"),
                uri.lastIndexOf("."));
        if(path.equals("/list")){
            ComputerDAO dao =
                new ComputerDAO();
            try {
                List<Computer> computers =
                    dao.findAll();
                request.setAttribute("computers", computers);
                request.getRequestDispatcher("computer_list.jsp")
                    .forward(request, response);
            } catch (Exception e) {
                e.printStackTrace();
                throw new ServletException(e);
            }
        }else if(path.equals("/buy")){
            long id = Long.parseLong(request.getParameter("id"));
            ComputerDAO dao =
                new ComputerDAO();
            try {
                Computer c = dao.findById(id);
                CartItem item = new CartItem();
```

```

        item.setC(c);
        item.setQty(1);
        HttpSession session =
            request.getSession();
        Cart cart = (Cart)session.getAttribute("cart");
        if(cart == null){
            //如果是第一次购买，需要先创建好 cart 对象
            //然后绑定到 session 对象上。
            cart = new Cart();
            session.setAttribute("cart", cart);
        }
        boolean flag = cart.add(item);
        if(!flag){
            //已经购买过该商品，则提示用户
            request.setAttribute("buy_error" + id,
                "已经购买过该商品");
            request.getRequestDispatcher("list.do")
                .forward(request, response);
        }else{
            //没有买过，返回到商品列表
            response.sendRedirect("list.do");
        }
    } catch (Exception e) {
        e.printStackTrace();
        throw new ServletException(e);
    }
}
}
}

```

20) 修改 computer\_list.jsp  
加入显示错误信息代码

```

66                                     </td>
67                                     <td class="altbg2">
68                                         Y<%=c.getPrice() %>
69                                     </td>
70     <td class="altbg2">
71         <a href="buy.do?id=<%=c.getId() %>">购买</a>
72     <%
73         String msg =
74             (String)request.getAttribute(
75                 "buy_error" + c.getId());
76     %>
77     <span style="color:red;">
78         <%= (msg == null ? "" : msg) %>
79     </span>
80 </td>
81                                     </tr>
82     <%
83     %>
84                                     }
85 </tbody>
86 </table>

```

## 21) 测试

a. 访问 [http://localhost:8080/web07\\_shopping/list.do](http://localhost:8080/web07_shopping/list.do)

b. 点击“购买”

地址跳向 [http://localhost:8080/web07\\_shopping/buy.do?id=1](http://localhost:8080/web07_shopping/buy.do?id=1)

c. 继续点击“购买”，出现提示“已经购买过该产品”



购买功能实现：)