**每日作业卷**

**javaweb第38天cookie\_session**

# 关卡1

## 训练案例1 记录用户上次访问时间

### 训练考核知识点

1\_创建Cookie,设置Cookie有效期,路径

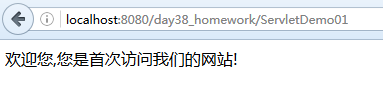
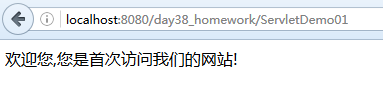
2\_发送Cookie

3\_获取Cookie

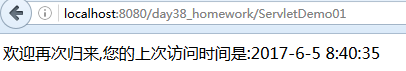
### 训练描述

1\_利用Cookie技术记录用户的上次访问时间

如果用户是首次访问本WEB应用,显示如下内容:



2\_如果用户不是首次访问本WEB应用,显示如下内容:



### 操作步骤描述

1\_由于我们再WEB开发中,经常要获取到客户端传递到服务端的指定名称的cookie对象,

所以我们先抽取一个工具类CookUtils. 代码如下:

public class CookUtils {

public static Cookie getCookieByName(String name,Cookie[] cks){

if(null==cks||cks.length==0){

return null;

}else{

for (Cookie ck : cks) {

if(ck.getName().equals(name)){

return ck;

}

}

}

return null;

}

}

2\_获取名称为history的cookie对象

//获取名称为history的cookie对象

Cookie ck=CookUtils.getCookieByName("history", request.getCookies());

3\_如果获取不到名称为history的cookie对象,创建名称为history,值为当前时间戳类型的

Cookie对象,设置有效期和路径之后将其发送,代码如下:

if(null==ck){

//如果获取不到

//说明首次访问,创建cookie对象(history:当前时间时间戳) ,设置cookie有效期,路径,

//将history cookie发送到客户端

ck=new Cookie("history",System.currentTimeMillis()+"");

ck.setPath("/day38\_homework");

ck.setMaxAge(Integer.MAX\_VALUE);

response.addCookie(ck);

//同时提示,您是首次访问本网站

out.println("欢迎您,您是首次访问我们的网站!");;

}

4\_如果可以获取到名称为history的cookie对象,设置新有效期和路径之后将其发送,

内容如下:

//可以获取到history的cookie对象,之前访问过本网站

//获取到history对应的值2342342,将其转换为long===>Date()类型,

//向客户端提示:您的上次访问时间为###. 同时设置history的value为此时的时间戳值,

//设置有效期,路径,将history发送到客户端

String value = ck.getValue(); // "234234234234"

long longTime = Long.parseLong(value);

Date dt=new Date(longTime);

out.println("欢迎再次归来,您的上次访问时间是:"+dt.toLocaleString());

ck.setValue(System.currentTimeMillis()+"");

ck.setPath("/day38\_homework");

ck.setMaxAge(Integer.MAX\_VALUE);

response.addCookie(ck);

完整代码如下:

public class ServletDemo01 extends HttpServlet {

private static final long serialVersionUID = 1L;

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

doGet(request, response);

}

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

response.setContentType("text/html;charset=utf-8");

PrintWriter out = response.getWriter();

//获取名称为history的cookie对象

Cookie ck=CookUtils.getCookieByName("history", request.getCookies());

if(null==ck){

//如果获取不到

//说明首次访问,创建cookie对象(history:当前时间时间戳) ,

//设置cookie有效期,路径,将history cookie发送到客户端

ck=new Cookie("history",System.currentTimeMillis()+"");

ck.setPath("/day38\_homework");

ck.setMaxAge(Integer.MAX\_VALUE);

response.addCookie(ck);

//同时提示,您是首次访问本网站

out.println("欢迎您,您是首次访问我们的网站!");;

}else{

//可以获取到history的cookie对象,之前访问过本网站

//获取到history对应的值2342342,将其转换为long===>Date()类型,

//向客户端提示:您的上次访问时间为###.

//同时设置history的value为此时的时间戳值,设置有效期,路径,将history发送到客户端

String value = ck.getValue(); // "234234234234"

long longTime = Long.parseLong(value);

Date dt=new Date(longTime);

out.println("欢迎再次归来,您的上次访问时间是:"+dt.toLocaleString());

ck.setValue(System.currentTimeMillis()+"");

ck.setPath("/day38\_homework");

ck.setMaxAge(Integer.MAX\_VALUE);

response.addCookie(ck);

}

}

}

## 训练案例1 记住用户名

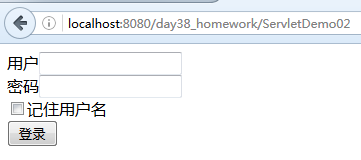
### 训练考核知识点

\*\_创建cookie对象,设置有效期,路径

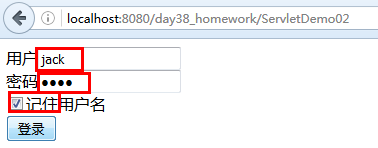
\*\_获取cookie对象

### 训练描述

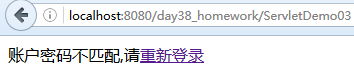
1\_用户首次访问系统登录界面显示如下效果



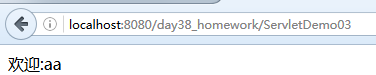
2\_用户录入账户,密码而且选中记住用户名,点击登录按钮

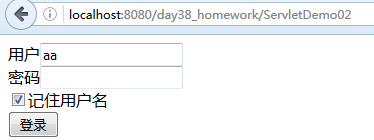


3\_登录失败显示如下效果



4\_登录成功显示如下效果:



5\_此时用户再次访问登录界面,显示如下效果:

### 操作步骤描述

1\_建立仓库day38\_homework,建立以下表

CREATE TABLE users(

uid INT PRIMARY KEY AUTO\_INCREMENT,

username VARCHAR(32),

PASSWORD VARCHAR(32)

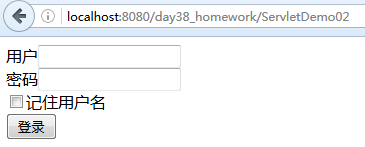
);

INSERT INTO users VALUES(1,'jack','jack');

INSERT INTO users VALUES(2,'lucy','lucy');

INSERT INTO users VALUES(3,'mary','mary');

2\_在ServletDemo02中向客户端输出表单,用于接收用户的账户和密码,效果如下



代码如下:

response.setContentType("text/html;charset=utf-8");

PrintWriter out = response.getWriter();

String str="";

String str01="";

str="<input type='checkbox' name='remUser' value='yes'/>记住用户名<br/>";

str01="用户<input type='text' name='user'/><br/>";

out.println("<form action='/day38\_homework/ServletDemo03' method='post'>");

out.println(str01);

out.println("密码<input type='password' name='pass'/><br/>");

out.println(str);

out.println("<input type='submit' value='登录'/>");

out.println("</form>");

3\_在ServletDemo03中接收用户表单数据,实现登录操作

response.setContentType("text/html;charset=utf-8");

PrintWriter out = response.getWriter();

String um = request.getParameter("user");

String up = request.getParameter("pass");

String re = request.getParameter("remUser");

User user = new User();

user.setUsername(um);

user.setPassword(up);

UserService UserService = new UserService();

User uu = UserService.login(user);

if (null != uu) {

out.print("欢迎:" + uu.getUsername());

} else {

out.print("账户密码不匹配,请<a href='/day38\_homework/ServletDemo02'>重新登录</a>");

}

} catch (Exception e) {

e.printStackTrace();

}

4\_实现UserService代码

public class UserService {

UserDao UserDao=new UserDao();

public User login(User user) throws SQLException {

return UserDao.login(user);

}

}

5\_实现UserDao代码

public User login(User user) throws SQLException {

String sql="select \* from users where username= ? and password=?";

QueryRunner qr=new QueryRunner(JDBCUtils.getDataSource());

Object[] params={user.getUsername(),user.getPassword()};

return qr.query(sql, new BeanHandler<User>(User.class),params);

}

6\_在登录成功的基础上实现发送cookie代码

Cookie ck=new Cookie("username", um);

if (null != re && "yes".equals(re)) {

ck.setMaxAge(60 \* 60 \* 24 \* 365);

ck.setPath("/day38\_homework");

}else{

ck.setMaxAge(0);

}

response.addCookie(ck);

7\_在ServletDemo02中判断是否存在名称为username的cookie来显示不同的表单内容.

代码如下:

String str="";

String str01="";

Cookie ck = CookUtils.getCookieByName("username", request.getCookies());

if(null==ck||"".equals(ck)){

str="<input type='checkbox' name='remUser' value='yes'/>记住用户名<br/>";

str01="用户<input type='text' name='user'/><br/>";

}else{

String username=ck.getValue();

str="<input type='checkbox' name='remUser' value='yes' checked='checked'/>记住用户名<br/>";

str01="用户<input type='text' name='user' value='"+username+"'/><br/>";

}

response.setContentType("text/html;charset=utf-8");

PrintWriter out = response.getWriter();

out.println("<form action='/day38\_homework/ServletDemo03' method='post'>");

out.println(str01);

out.println("密码<input type='password' name='pass'/><br/>");

out.println(str);

out.println("<input type='submit' value='登录'/>");

out.println("</form>");

# 关卡2

## 训练案例1 session实现简易版本购物车

### 训练考核知识点

1\_创建session对象

2\_向session对象内放入键值对数据

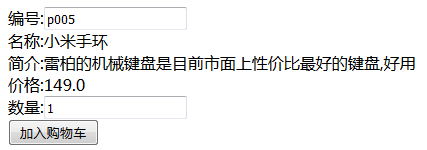
3\_获取session中指定的数据

### 训练描述

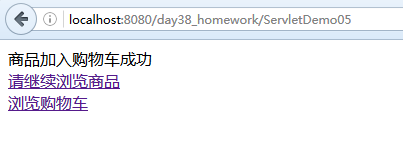
1\_当用户访问demo01.html实现如下效果



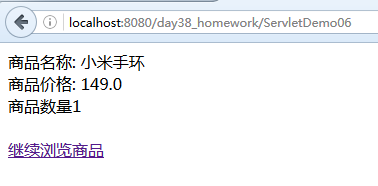
2\_当用户点击查看,可以查看每件商品的详细信息



3\_当加入购物车,可以将当前商品加入购物车,效果如下



4\_当点击浏览购物车,可以出现购物车中的内容,效果如下



### 操作步骤描述

1\_新建demo01.html页面,实现如下效果



代码如下:

<table border="1" width="50%" align="center">

<tr><td>商品名称</td><td>查看</td></tr>

<tr>

<td>雷柏机械键盘</td>

<td><a href='/day38\_homework/ServletDemo04?pid=p001'>查看</a></td></tr>

<tr>

<td>联想笔记本电脑</td>

<td><a href='/day38\_homework/ServletDemo04?pid=p002'>查看</a></td></tr>

<tr>

<td>小米6手机</td>

<td><a href='/day38\_homework/ServletDemo04?pid=p003'>查看</a></td></tr>

<tr>

<td>华为荣耀8手机</td>

<td><a href='/day38\_homework/ServletDemo04?pid=p004'>查看</a></td></tr>

<tr>

<td>小米手环</td>

<td><a href='/day38\_homework/ServletDemo04?pid=p005'>查看</a></td></tr>

<tr>

<td colspan='2'>

<a href="/day38\_homework/ServletDemo06">查看购物车内容</a></td>

</tr>

</table>

2\_实现Product类,代表各个商品信息

public class Product {

private String pid;//商品编号

private String pname;//商品名称

private String pdesc; //商品描述

private double price; //价格

private int num;//商品数量

//getter and setter....

}

3\_实现类DB,在本类中我们用一个MAP来模拟了仓库中的各个商品信息,

并在其中实现一个根据PID获取商品信息的方法

public class DB {

private static Map<String,Product> map=new HashMap<String,Product>();

static{

//public Product(String pid, String pname, String pdesc, double price) {

Product p1=new Product("p001","雷柏机械键盘","雷柏的机械键盘是目前市面上性价比最好的键盘,好用",199);

Product p2=new Product("p002","联想笔记本电脑","雷柏的机械键盘是目前市面上性价比最好的键盘,好用",7999);

Product p3=new Product("p003","小米6手机","雷柏的机械键盘是目前市面上性价比最好的键盘,好用",1999);

Product p4=new Product("p004","华为荣耀8手机","雷柏的机械键盘是目前市面上性价比最好的键盘,好用",2001);

Product p5=new Product("p005","小米手环","雷柏的机械键盘是目前市面上性价比最好的键盘,好用",149);

map.put("p001", p1);

map.put("p002", p2);

map.put("p003", p3);

map.put("p004", p4);

map.put("p005", p5);

}

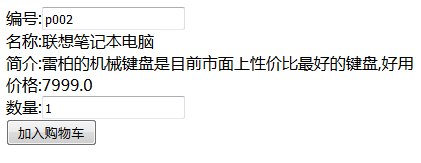
public static Product getProductByPid(String pid){

return map.get(pid);

}

}

4\_实现ServletDemo04,获取各个商品的PID,向客户端输出商品详情信息,效果如下



代码如下:

response.setContentType("text/html;charset=utf-8");

PrintWriter out = response.getWriter();

String pid = request.getParameter("pid");

Product p = DB.getProductByPid(pid);

out.println("<form action='/day38\_homework/ServletDemo05' method='post'>");

out.println("编号:<input type='text' name='pid' value='"+p.getPid()+"'/><br/>");

out.println("名称:"+p.getPname()+"<br/>");

out.println("简介:"+p.getPdesc()+"<br/>");

out.println("价格:"+p.getPrice()+"<br/>");

out.println("数量:<input type='text' name='num' value='1'/><br/>");

out.println("<input type='submit' value='加入购物车'/>");

out.println("</form>");

5\_实现ServletDemo05,接收商品编号和数量,将其放入到MAP中(这个MAP代表购物车),将

Map放入session域对象内,并向客户端生成相应信息

response.setContentType("text/html;charset=utf-8");

PrintWriter out = response.getWriter();

String pid = request.getParameter("pid");

int num = Integer.parseInt(request.getParameter("num"));

Map<String, Product> map =

(Map<String, Product>) request.getSession().getAttribute("cart");

if (null == map) {

map = new HashMap<String, Product>();

request.getSession().setAttribute("cart", map);

}

if (!map.containsKey(pid)) {

Product pro = DB.getProductByPid(pid);

pro.setNum(1);

map.put(pid, pro);

} else {

Product pro = map.get(pid);

pro.setNum(pro.getNum() + num);

}

out.println("商品加入购物车成功<br/>");

out.println("<a href='/day38\_homework/demo01.html'>请继续浏览商品</a><br/>");

out.println("<a href='/day38\_homework/ServletDemo06'>浏览购物车</a><br/>");

6\_实现ServletDemo06代码,实现点击浏览购物车的链接时,实现的功能

response.setContentType("text/html;charset=utf-8");

PrintWriter out = response.getWriter();

Map<String,Product> map=(Map<String,Product>)request.getSession().getAttribute("cart");

if(null==map||map.size()==0){

String str=”<a href='/day38\_homework/demo01.html'>浏览商品</a>”;

out.println("购物车中暂无商品信息"+str);

}else{

for(Product pro:map.values()){

out.println("商品名称: "+pro.getPname()+"<br/>");

out.println("商品价格: "+pro.getPrice()+"<br/>");

out.println("商品数量"+pro.getNum()+"<br/><br/>");

}

}

out.println("<a href='/day38\_homework/demo01.html'>继续浏览商品</a>");

# 关卡3

## 训练案例1 使用session防止表单重复提交

### 训练考核知识点

1\_获取session对象

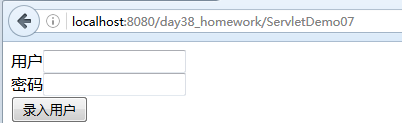
2\_获取session中键值对数据

3\_向session存入键值对数据

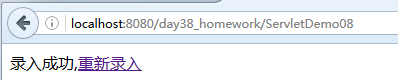
4\_移除session域对象范围内数据

### 训练描述

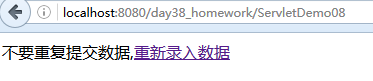
1\_用户访问ServletDemo07,获取到录入用户数据的表单界面



2\_用户录入用户和密码信息,点击录入用户按钮,看到如下效果



3\_此时刷新浏览器,用户不能再次重复录入数据

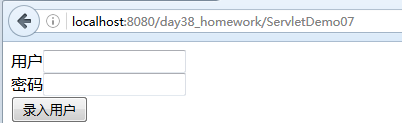


### 操作步骤描述

1\_实现ServletDemo07,向客户端输出一个录入用户数据的表单,而且为了防止用户

重复提交数据,生成了一个token的随机字符串,而且已经在session中存储了一份

,在表单的隐藏域中输出了一份



代码如下:

response.setContentType("text/html;charset=utf-8");

PrintWriter out = response.getWriter();

String token=UUIDUtils.getId();

request.getSession().setAttribute("token", token);

out.print("<form action='/day38\_homework/ServletDemo08' method='post'>");

out.print("<input type='hidden' name='token' value='"+token+"'/>");

out.print("用户<input type='text' name='username'/><br/>");

out.print("密码<input type='password' name='password'/><br/>");

out.print("<input type='submit' value='录入用户'/>");

out.print("</form>");

2\_实现ServletDemo08,接收用户表单数据,比较获取到的token和session中的token是否是

同一份,如果不是,向客户端响应账户密码不能重复的提示,如果是同一份,清除session中

的token之后,调用UserService中的录入学生的功能

try {

response.setContentType("text/html;charset=utf-8");

PrintWriter out = response.getWriter();

String token01 = request.getParameter("token");

String token02 = (String) request.getSession().getAttribute("token");

request.getSession().removeAttribute("token");;

if (!token01.equals(token02)) {

String ss=”<a href='/day38\_homework/ServletDemo07'>重新录入数据</a>”

out.println("不要重复提交数据,"+ss);

return;

}

String um = request.getParameter("username");

String up = request.getParameter("password");

User user = new User();

user.setUid(UUIDUtils.getId());

user.setUsername(um);

user.setPassword(up);

UserService UserService = new UserService();

UserService.addUser(user);

String ss01=”录入成功,<a href='/day38\_homework/ServletDemo07'>重新录入</a>”;

out.println(ss01);

} catch (Exception e) {

e.printStackTrace();

}

3\_实现UserServcie中录入用户功能

public void addUser(User user) throws SQLException {

UserDao.addUser(user);

}

4\_实现UserDao中代码

public void addUser(User user) throws SQLException {

String sql="insert into users values (? , ? , ? )";

QueryRunner qr=new QueryRunner(JDBCUtils.getDataSource());

qr.update(sql,user.getUid(),user.getUsername(),user.getPassword());

}