

【济南中心】JavaEE就业班同步笔记第一阶段：JavaWeb之Cookie...

小鲁哥哥 • 2017-1-19 20:36:50

点击查看济南黑马校区最新开班计划**【济南中心】JavaEE就业班同步笔记第一阶段：
JavaWeb之核心技术--Cookie与Session****1 案例一：记录用户的上次登陆访问时间.****1.1 需求:**

用户登录完成后,显示您是第x位访问的用户,您的上次访问时间是:yyyy-MM-dd.

* 如果第一次访问的话, 只显示您是第x位用户.

* 如果不是第一次访问的话,显示您是第x位访问的用户,您的上次访问时间是:yyyy-MM-dd.

1.2 分析:**1.2.1 技术分析:****【会话技术】**

什么是会话:用户打开一个浏览器访问页面,访问网站的很多页面,访问完成后将浏览器关闭的过程称为是一次会话.

常见的会话技术:

* Cookie :将数据保存到客户端浏览器.

* Session :将数据保存到服务器端.

回复帖子...

回帖

参见图一和图二

使用会话技术:

【Cookie技术的使用】

向浏览器保存数据:

HttpServletResponse有一个方法:

- * void addCookie(Cookie cookie);

获得浏览器带过来的Cookie:

HttpServletRequest有一个方法:

- * Cookie[] getCookies();

创建一个Cookie对象:

- * Cookie(String name,String value);

【JSP的简单概述】

什么是JSP :Java Server Pages(Java服务器端页面).JSP = Java代码 + HTML的元素 + JSP内置东西

SUN公司为什么推出JSP动态网页开发技术:

- * SUN公司推出的Servlet技术进行动态网页开发.发现Servlet自身有不足没有办法与ASP,PHP技术竞争.想在动态网页中输出表单.在Servlet中获得PrintWriter out = response.getWriter();

- * out.println("<form action=" method=">");

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

- * SUN又推出了动态的网页开发技术就是JSP.

JSP的执行过程:

- * JSP会被翻译成Servlet,编译成class进行执行的.

JSP的嵌入Java代码:JSP的脚本元素

- * <%! %>:翻译成类中的成员部分. 定义变量,定义方法,定义类.Servlet是线程不安全的,尽量少在类中定义成员属性!!

- * <% %>:翻译成类的service方法内部的内容. 定义变量,定义类,直接写代码块.

- * <%= %>:翻译成service方法内部的out.print();

1.2.2 步骤分析:

【步骤一】: 准备登陆的案例.

【步骤二】: 在统计人数的Servlet中.判断是否是第一次访问.

【步骤三】: 根据是否是第一次显示不同的信息, 同时将当前的时候保存到Cookie中.

1.3 代码实现

```
[mw_shl_code=java,true]public class CountServlet extends HttpServlet {  
    private static final long serialVersionUID = 1L;  
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
        response.setContentType("text/html;charset=UTF-8");
```

```

Integer count = (Integer) this.getContext().getAttribute("count");
// response.getWriter().println("<h1>现在网站被访问的次数为:"+count+"</h1>");

/**
 * 获得浏览器中带过来的所有的Cookie信息,从数组中查找有没有指定名称的Cookie
 * 判断用户是否是第一次访问:(从数组中没有找到指定名称的Cookie)
 * * 如果是第一次:显示欢迎,记录当前访问的时间存入到Cookie中.
 * * 如果不是第一次:显示欢迎,上一次访问时间,同时记录当前访问的时间存入到Cookie中。
 */
// 获得浏览器带过来的所有的Cookie:
Cookie[] cookies = request.getCookies();
// 从数组中查找指定名称的Cookie:
Cookie cookie = CookieUtils.findCookie(cookies, "lastVisit");
// 判断是否是第一次:
if(cookie == null){
    // 第一次访问
    response.getWriter().println("您是第"+count+"位访客!");
}else{
    // 不是第一次
    Long l = Long.parseLong(cookie.getValue());
    Date d = new Date(l);
    response.getWriter().println("您是第"+count+"位访客! 上次访问时间是:"+d.toLocaleString());
;

}
// 创建一个Cookie对象:
Cookie c = new Cookie("lastVisit", ""+System.currentTimeMillis());
// 保存到浏览器端:
response.addCookie(c);
}

```

```

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    doGet(request, response);
}
}[/mw_shl_code]

```

1.4 总结:

1.4.1 Cookie的常用API:

Cookie的常用的API:

- * getName();
- * getValue();
- * setDomain(String domain); -- 设置Cookie的有效域名. // www.baidu.com music.baidu.com
- * setPath(String path); -- 设置Cookie的有效路径.
- * setMaxAge(int maxAge); -- 设置Cookie的有效时间.

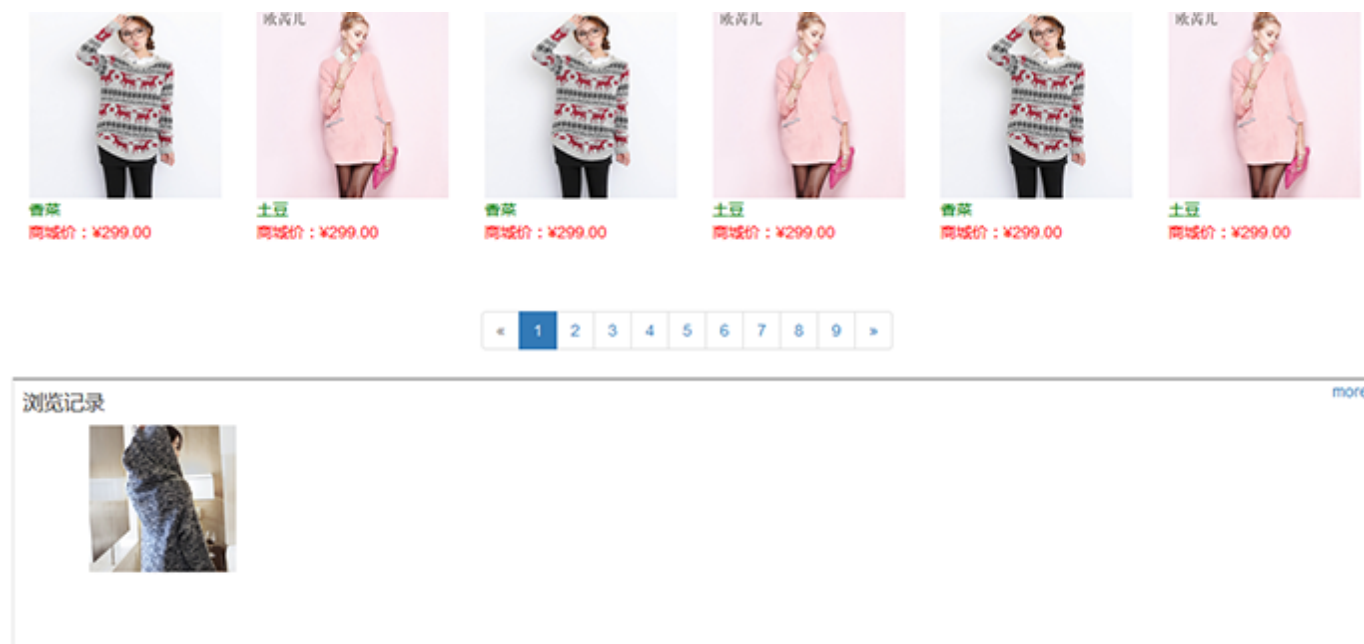
Cookie的分类有关:

- * 会话级别的Cookie:默认的Cookie.关闭浏览器Cookie就会销毁.
- * 持久级别的Cookie:可以设置Cookie的有效时间.那么关闭浏览器Cookie还会存在. 手动销毁持久性Cookie. setMaxAge(0)---前提是有效路径必须一致.

2 案例二：记录用户的商品浏览记录：

2.1 需求：

在购物网站上浏览商品的信息,商家为了留住用户,记住之前浏览的一些商品.



2.2 分析：

2.2.1 步骤分析：

- 【步骤一】：在登录完成后, 显示商品列表页面.
- 【步骤二】：为商品列表页面做一些准备工作.
- 【步骤三】：点击某个商品,将商品ID传递一个Servlet.

- 【步骤四】：在Servlet中:判断是否是第一次浏览商品
- 【步骤五】：如果是第一次：将商品的ID存入到Cookie中即可.
- 【步骤六】：如果不是第一次：判断该商品是否已经浏览了.
- 【步骤七】：如果浏览器过.删除之前元素,将该元素添加到最前面.
- 【步骤八】：如果没有浏览过该商品.判断最大长度，没有超过限制,直接加到最前,如果已经超过限制,删除最后一个，将其插入到最前.

2.3 代码实现:

```
[mw_shl_code=java,true]public class ProductServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        /**
         * * 接收商品id.
         * * 接收从客户端带过来的所有Cookie.
         * * 从Cookie的数组中查找指定名称的Cookie.
         * * 判断是否是第一次浏览商品:
         *     * 第一次浏览商品
         *         * 直接将商品的ID存入到Cookie.
         *         * 将Cookie回写到浏览器.
         *     * 不是第一次浏览商品 1-2
         *         * 判断当前的商品是否已经在浏览记录.
         *             * 已经存在: 2-1 移除当前元素，将当前元素添加到最开始.
         *             * 没在浏览记录中:
         *                 * 判断是否已经超过了最大长度:如果超过 2-1-3:删除最后一个 将当前元素添加到最前面.
         *                 * 没有超过:直接将该元素添加到最前位置.
         *             * 将转换的id的值存入到Cookie,回写浏览器.
         */
        // 接收id:
        String id = request.getParameter("id");
        // 获得所有的Cookie的信息:
        Cookie[] cookies = request.getCookies();
        // 判断是否是第一次:
        Cookie cookie = CookieUtils.findCookie(cookies, "history");
        if(cookie == null){
            // 第一次浏览商品
```

```

Cookie c = new Cookie("history",id);
c.setPath("/day11");
c.setMaxAge(60*60*24*7);
response.addCookie(c);
}else{
    // 不是第一次浏览
    // 判断选择的商品是否已经在浏览记录中 2-1
    String value = cookie.getValue();
    String[] ids = value.split("-");
    // 将数组变为集合:
    LinkedList<String> list = new LinkedList<String>(Arrays.asList(ids));
    if(list.contains(id)){
        // 之前浏览过该商品
        list.remove(id); // 1-2-3
        list.addFirst(id);
    }else{
        // 没有浏览过该商品.
        if(list.size() >=3 ){
            // 超过3个
            list.removeLast();
            list.addFirst(id);
        }else{
            // 没到3个.
            list.addFirst(id);
        }
    }
    // 将list中的元素取出,使用-连接上保存到Cookie,写回浏览器.
    StringBuffer sb = new StringBuffer();
    for(String s:list){
        sb.append(s).append("-");
    }
    String sValue = sb.toString().substring(0,sb.length()-1);
    System.out.println(sValue);
    // 存入到Cookie中
    Cookie c = new Cookie("history",sValue);
    c.setPath("/day11");
    c.setMaxAge(60*60*24*7);

```

```

        response.addCookie(c);
    }

    request.getRequestDispatcher("/demo2/product_info.htm").forward(request, response);
}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    doGet(request, response);
}
}[/mw_shl_code]

```

2.4 总结:

2.4.1 清空浏览记录:

删除持久性的Cookie:

```

[mw_shl_code=java,true]public class ClearServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        Cookie cookie = new Cookie("history", null);
        cookie.setPath("/day11");
        cookie.setMaxAge(0);
        response.addCookie(cookie);

        response.sendRedirect("/day11/demo2/product_list.jsp");
    }

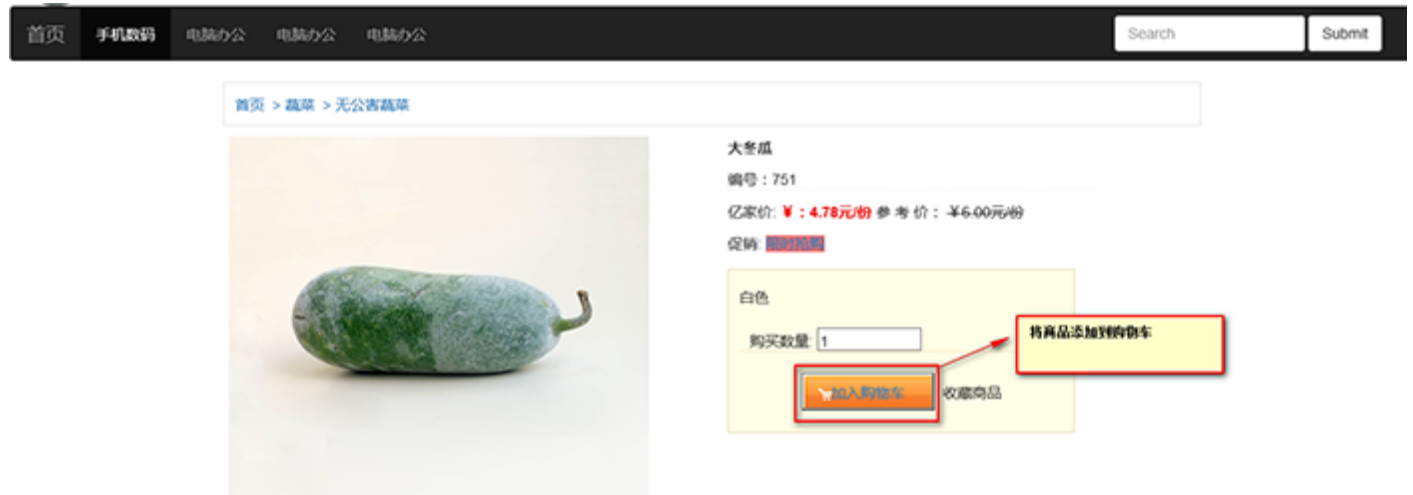
    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
    }
}[/mw_shl_code]

```

3 案例三：将商品添加到购物车：

3.1 需求：

查看某个商品详情的时候,添加到购物车.需要将商品添加到购物车.



3.2 分析：

1.4.2.1 技术分析：

【Session的概述】

Cookie本身是有大小和个数的限制.Session没有限制.Cookie的数据保存在客户端,Session数据保存在服务器端.

Session的执行原理：基于Cookie的.

使用Session：

* 获得Session：

* `request.getSession();`

3.2.2 步骤分析：

【步骤一】：点击加入购物车提交到Servlet

【步骤二】：在Servlet将购物的商品存入到Session中.

【步骤三】：可以创建一个Map集合用于保存购物信息Map的key可以是商品的名称,Map的value是数量.

【步骤四】：在购物车页面中显示Map中的数据就可以.

3.3 代码实现：

```
[mw_shl_code=java,true]public class CartServlet extends HttpServlet {  
    private static final long serialVersionUID = 1L;  
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException
```



```

ception, IOException {
    // 接收商品名称:
    String name = new String(request.getParameter("name").getBytes("ISO-8859-1"), "UTF-8");
    // 创建Map集合用于保存购物信息.Map<String,Integer> Map的key是商品的名称 value是购买的数量.
    Map<String,Integer> map = (Map<String, Integer>) request.getSession().getAttribute("cart");
    if(map == null){
        map = new LinkedHashMap<String,Integer>();
    }
    // 判断购物车中是否已经买了该商品.
    if(map.containsKey(name)){
        // map中已经有该商品:// * 如果购物车中已经有该商品: 获得到Map中该商品的数量+1。 存回到Map集合中.
        Integer count = map.get(name);
        count++;
        map.put(name, count);
    }else{
        // map中没有该商品.// * 如果购物车中没有改商品: 将商品添加到Map集合中 数量1.
        map.put(name, 1);
    }

    // * 将Map集合保存到session中.
    request.getSession().setAttribute("cart", map);
    response.setContentType("text/html;charset=UTF-8");
    response.getWriter().println("<h3><a href='/day11/demo2/product_list.jsp'>继续购物</a> | <a href='/day11/demo2/cart.jsp'>去结算</a></h3>");
    }[/mw_shl_code]

```

3.4 总结:

3.4.1 Session是域对象:

session何时创建和销毁?作用范围:

- * 创建:服务器端第一次调用getSession()创建session.
- * 销毁:三种情况销毁session:
 - * 1.session过期. 默认过期时间为30分钟.
 - * 2.非正常关闭服务器.如果正常关闭session序列化到硬盘.
 - * 3.手动调用session.invalidate();
- * 作用范围:多次请求.(一次会话)

4 案例四：进行一次验证码的校验

4.1 需求：

在登录的页面中，需要有一个验证码的校验。

The screenshot shows a web page with a black navigation bar at the top containing links: 首页, 手机数码, 电脑办公, 电脑办公, 电脑办公. To the right of the navigation bar is a search bar with the text 'Search' and a 'Submit' button. Below the navigation bar is a large red banner. On the left side of the banner is a large white text graphic that reads '11月11天 同是低价 爽购11天'. On the right side of the banner is a white login form titled '会员登录 USER LOGIN'. The form contains the following fields: '用户名' (Username) with the value 'aaa', '密码' (Password) with the placeholder '请输入密码', and '验证码' (Captcha) with the placeholder '请输入验证码'. To the right of the captcha input is a green box containing the characters 'C K X'. Below the input fields are two checkboxes: '自动登录' (Auto login) and '记住用户名' (Remember username). At the bottom of the form is a red '登录' (Login) button.

4.2 分析：

4.2.1 技术分析：

使用session保存生产的验证码.

4.2.2 步骤分析：

- 【步骤一】：生成验证码,将生成验证码的随机的4个字母或数字保存到session中.
- 【步骤二】：在页面中输入验证码值,点提交.
- 【步骤三】：在Servlet中从表单中获得验证码从session中获得一个验证码.
- 【步骤四】：比对两个验证码值是否一致.
- 【步骤五】：将session中保存的验证码清空.

4.3 代码实现：

```
[mw_shl_code=java,true]protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
    // 校验验证码程序:  
    String code1 = request.getParameter("code");  
    String code2 = (String) request.getSession().getAttribute("code");  
    request.getSession().removeAttribute("code");  
    if(!code1.equalsIgnoreCase(code2)){  
        request.setAttribute("msg", "验证码输入错误! ");  
        request.getRequestDispatcher("/demo2/login.jsp").forward(request, response);  
    }  
}
```

```
        return ;  
    }  
    ...  
}[/mw_shl_code]4.4 总结:
```

4.4.1 使用JS控制图片切换:

```
[mw_shl_code=javascript,true]<script type="text/javascript">  
    function changelmg(){  
        document.getElementById("img1").src="/day11/ChecklmgServlet?time="+new Date().getTime()  
    ;  
    }  
</script>[/mw_shl_code]
```



回帖

— 25条回帖 —



济南王昭珽

还有比这更详细的么？没有。

沙发 • 2017-2-4 10:35:40

回帖



小虎同学

来看看，学习学习

藤椅 • 2017-2-4 11:10:56

回帖



来自宇宙超级黑马专属苹果客户端



守护星空

学习学习!!!!

板凳 • 2017-2-4 13:51:55

回帖



zmanx

这笔记很详细啊,收藏一下以后看看

回帖



爱吃橘子的小泽

顶一下,祝黑马越办越好

地板 • 2017-2-12 22:10:07

回帖



a690223483

顶顶顶 早晚会用到 马上就去学习了

7# • 2017-2-22 21:03:14

回帖



zhangkaitong

谢谢分享 赞赞赞!!!!

8# • 2017-2-24 13:30:28

回帖



xulinml

谢谢老师~! ~感谢感谢

9# • 2017-2-28 08:57:28

回帖



为过去的我买单

谢谢分享!

10# • 2017-3-2 10:06:53

回帖



zhouxiaoyang









11# • 2017-3-26 23:25:50

回帖



ys5214

23333333333333

12# • 2017-4-1 13:27:44

回帖



Cerol

力挺楼主，支持你 666

13# • 2017-4-12 17:18:14

回帖



hrywzn

学习一下,真不错

14# • 2017-5-22 22:26:41

回帖



hrywzn

谢谢分享!

15# • 2017-5-22 22:27:39

回帖



zsldeyou

力挺楼主，支持你 666

16# • 2017-5-29 10:46:58

回帖



liujinlong666

谢谢分享！

17# • 2017-5-31 00:13:17

回帖



liujinlong666

谢谢分享！！

18# • 2017-6-1 00:06:28

回帖



dhj

学习学习，感谢分享！！

19# • 2017-6-11 16:45:39

回帖



dhj

感谢无私分享！！！！

20# • 2017-6-11 17:01:43

回帖

上一页

1/2

下一页