关于Cookie的知识

1.

<%

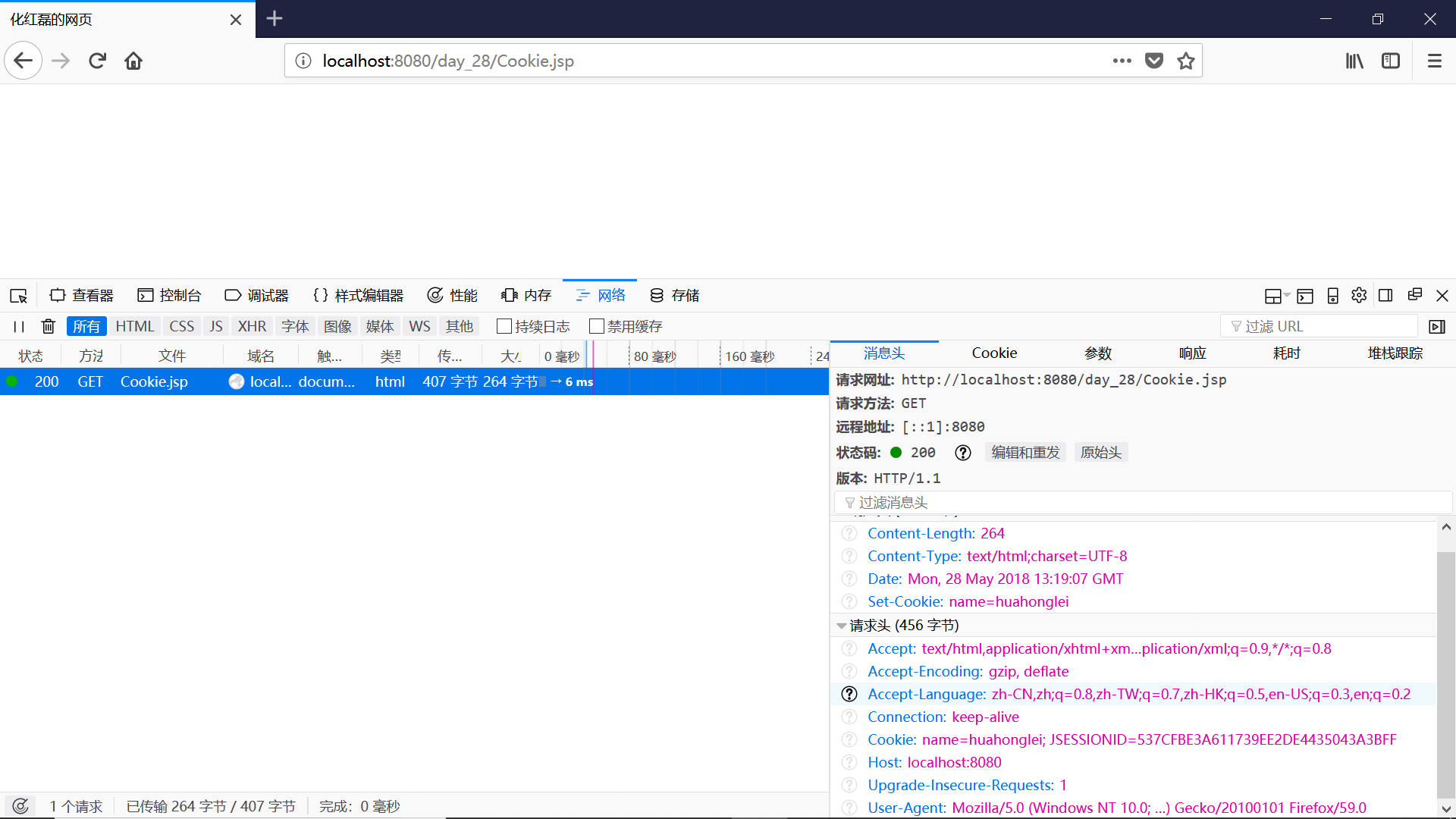
//1.创建一个Cookie对象

Cookie cookie=**new** Cookie("name","huahonglei");

//2.调用response中的方法把把这个对象发送到客户端

response.addCookie(cookie);

%>



可以看出右下角有设置的对应的cookie的值，注意，第一次请求的时候，，，请求里面没有cookie，然后响应传递给浏览器一个cookie，然后第二次请求时就有cookie了

2.

<%

//获取Cookie

Cookie[] cookies= request.getCookies();

**if**(cookies!=**null** && cookies.length>0){

**for**(Cookie cookie:cookies){

//2.获取Cookie的name和value

out.println(cookie.getName()+":"+cookie.getValue());

}

}**else**{

out.print("没有一个Cookie，正在创建并返回");

//1.创建一个Cookie对象

Cookie cookie =**new** Cookie("name","huahonglei");

//setMaxAge:设置Cookie的最大失效，以秒为单位，若为0，表示立即删除该Cookie

//若为负数，表示不存储该Cookie，若为正数，表示Cookie的存储时间

cookie.setMaxAge(30);

//2.调用response的一个方法把Cookie传给客户端

response.addCookie(cookie);

}

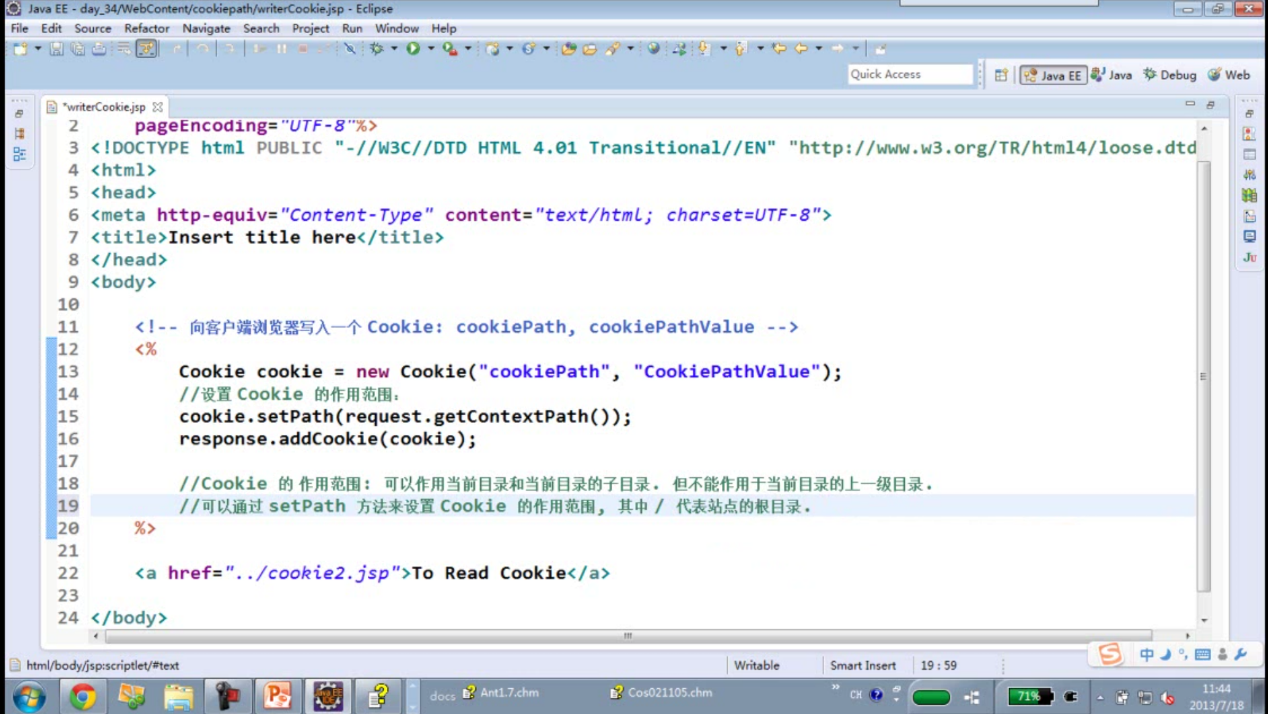
%>

3.关于自动登录

若可以获取到请求参数name，则打印出欢迎信息，把登录信息存储到Cookie中，并设置Cookie的最大时效为30s

从Cookie中读取用户信息，若存在则打印欢迎信息

若既没有请求参数，也没有cookie，则重定向到login.jsp

****

设置Cookie的作用范围。

Cookie的作用范围：可以作用当前目录和当前目录的子目录，但不能作用当前目录的上一句目录

可以通过setPath方法来设置Cookie的作用范围，其中/代表站点的根目录。

<%

Cookie cookie=**new** Cookie("name","value");

//设置Cookie的作用范围

cookie.setPath(request.getContextPath());

response.addCookie(cookie);

//Cookie的作用范围：可以作用当前目录的子目录，但是不能作用域当前目录的上一级目录.

//可以通过setPath方法设置Cookie的作用范围，其中/代表站点的根目录

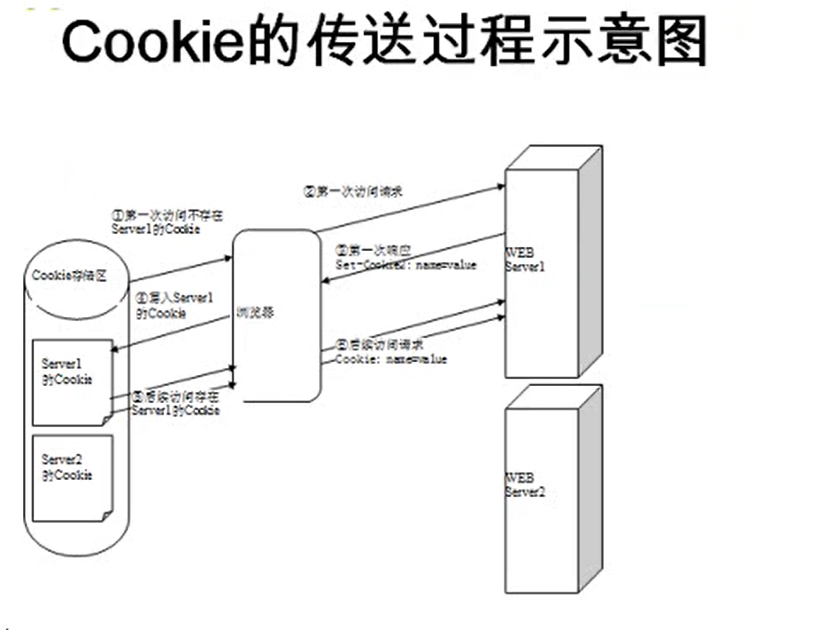
%>

1. **Cookie**

**1）.完成会话跟踪的一种机制：采用的是在客户端保持HTTP状态信息的方案**

**2）.Cookie是在浏览器访问WEB服务器的某个资源时，由WEB服务器在HTTP响应消息头中附带传送给浏览器的一个小文本文件。**

**3).一旦WEB浏览器保存了某个Cookie，那么它在以后每次访问该WEB服务器时，都会在HTTP请求头中将这个Cookie回传给WEB服务器**



**4）.底层原理:WEB服务器通过在HTTP响应消息中增加Set-Cookie响应头字段Cookie信息发送给浏览器，浏览器则通过在HTTP请求消息中增加Cookie2请求字段将Cookie回传给WEB服务器。**

1. **Cookie相关的API**

**1）.创建一个Cookie对象**

**Cookie cookie=new Cookie(“name”,”cookie”);**

**2).setMaxAge:设置Cookie的最大时效，以秒为单位，若为0，表示立即删除该Cookie**

**若为负数，表示不存储该Cookie，若为正数，表示该Cookie的存储时间**

**cookie.setMaxAge(30);**

**3).调用response的一个方法把Cookie传给客户端**

**response.addCookie(cookie);**

**4).设置Cookie的作用范围：可以作用当前目录和当前子目录，但不能作用域当前目录的上一句目录**

**cookie.setPath(request.getContextPath);**

**5)获取Cookie**

Cookie[] cookies=request.getCookies();

**if**(cookies!=**null**&&cookies.length>0){

**for**(Cookie cookie:cookies){

out.println(cookie.getName()+":"+cookie.getValue());

}

}

1. **会话cookie和持久cookie**

**1).如果不设置过期时间，则表示这个cookie生命周期为浏览器会话期间，只要关闭浏览器窗口，cookie就消失了。这种声明周期为浏览器会话期的cookie被称为会话cookie，会话cookie一般不保存在硬盘上而是保存在内存里**

**2）.如果设置了过期时间，浏览器就会把cookie保存到硬盘上，关闭后再次打开浏览器，这些cookie依然后效直到超过设定的过期时间**

1. **两个应用**

**1).自动登录**

**2).显示浏览过的商品的信息**