一.Cookie介绍

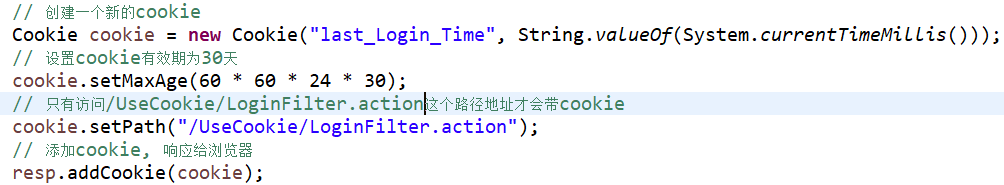
1.Cookie其实是一份小数据, 是服务器给客户端, 并且存储在客户端上的一份小数据。

2.应用场景: 自动登录、浏览记录、购物车。

3.为什么要有这个Cookie: http的请求是无状态。 客户端与服务器在通讯的时候, 是无状态的, 其实就是客户端在第二次来访的时候, 服务器根本就不知道这个客户端以前有没有来访问过。 Cooke就是为了更好的用户体验, 更好的交互[自动登录], 其实从公司层面讲, 就是为了更好的收集用户习惯[大数据]。

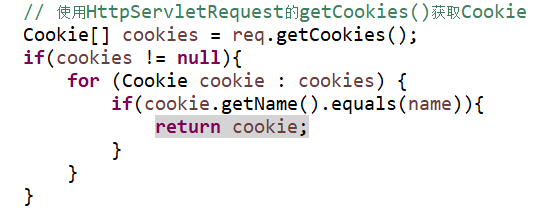
4.创建Cookie并发送给浏览器

(1).创建Cookie并发送给浏览器

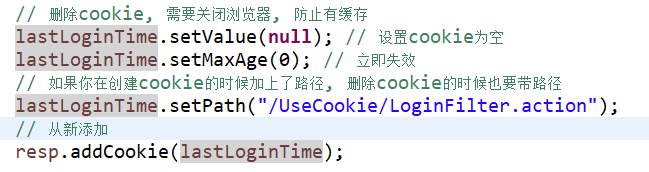


(2).客户端收到的信息里面, 响应头中多了一个字段Set-Cookie。

5.获取Cookie



6.清除Cookie



7.Cookie的常用方法

(1).创建Cookie: new Cookie(“name”, “value”);

(2).设置Cookie有效期: cookie.setMaxAge(60 \* 60 \* 24 \* 30);

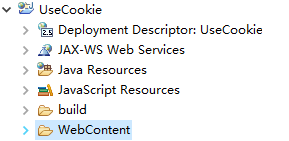
(3).设置路径, 只有访问/UseCookie/LoginFilter.action这个路径地址才会带cookie: cookie.setPath("/UseCookie/LoginFilter.action");

(4).使用HttpServletResponse的addCookie()方法, 服务器响应浏览器的时候把cookie发送给浏览器: resp.addCookie(cookie);

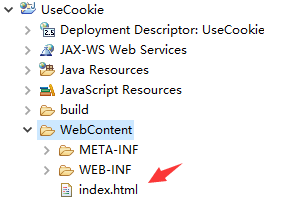
(5).使用HttpServletRequest的getCookies()获取请求中携带的Cookie数组: Cookie[] cookies = req.getCookies();

二.使用Cookie获取用户上次登录时间

1.创建一个UseCookie的Web工程



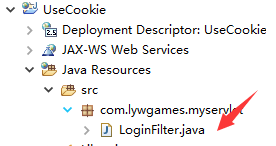
2.在WebContent下创建一个index.html



3.编写index.html

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <title>登录界面</title>  </head>  <body>  <form action="LoginFilter.action" method="post">  账号:<input type="text" name="username"/><br>  密码:<input type="text" name="password"/><br>  <input type="submit" value="登录"/>  </form>  </body>  </html> |

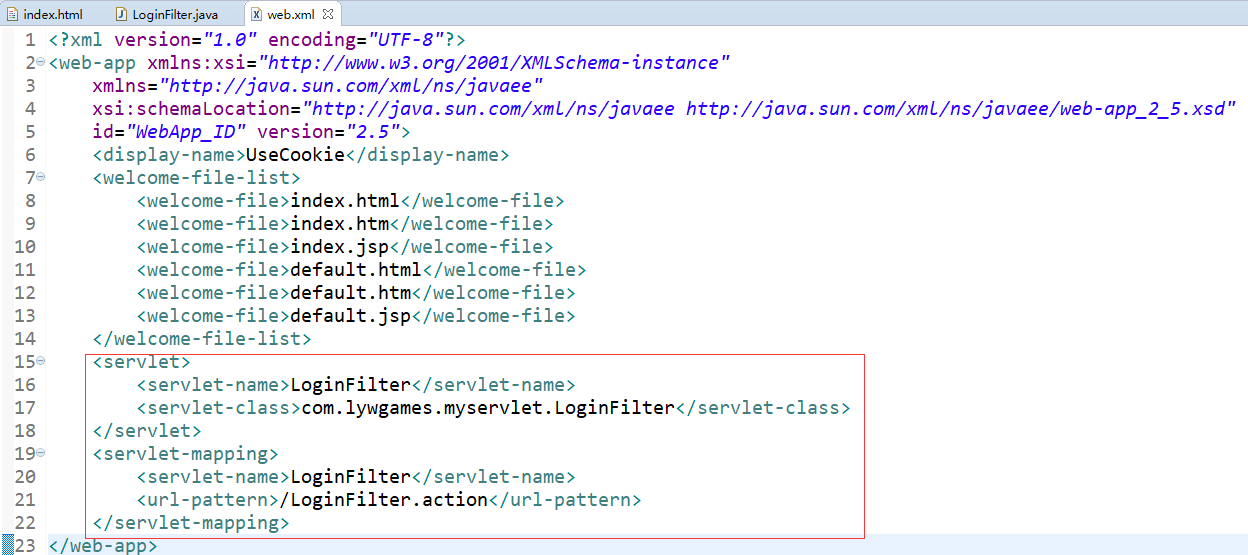
4.创建LoginFilter.java



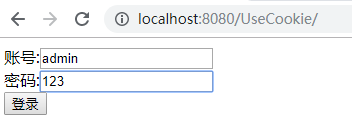
5.编写LoginFilter.java

|  |
| --- |
| package com.lywgames.myservlet;  import java.io.IOException;  import java.util.Date;  import javax.servlet.ServletException;  import javax.servlet.http.Cookie;  import javax.servlet.http.HttpServlet;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;  public class LoginFilter extends HttpServlet {  private static final long serialVersionUID = 1L;  public Cookie getCookie(HttpServletRequest req, String name) {  // 使用HttpServletRequest的getCookies()获取Cookie  Cookie[] cookies = req.getCookies();  if(cookies != null){  for (Cookie cookie : cookies) {  if(cookie.getName().equals(name)){  return cookie;  }  }  }    return null;  }    @Override  protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  resp.setContentType("text/html;charset=utf-8");    String userName = req.getParameter("username");  String password = req.getParameter("password");    if("admin".equals(userName) && "123".equals(password)){  // 获取lastLoginTime的Cookie  Cookie lastLoginTime = getCookie(req, "last\_Login\_Time");  // 如果lastLoginTime为空, 说明是第一次登录  if(lastLoginTime == null){  // 创建一个新的cookie  Cookie cookie = new Cookie("last\_Login\_Time", String.valueOf(System.currentTimeMillis()));  // 设置cookie有效期为30天  cookie.setMaxAge(60 \* 60 \* 24 \* 30);  // 只有访问/UseCookie/LoginFilter.action这个路径地址才会带cookie  cookie.setPath("/UseCookie/LoginFilter.action");  // 添加cookie, 响应给浏览器  resp.addCookie(cookie);    // 向客户端输出登录信息  resp.getWriter().println("欢迎您, " + userName);  }else{  Date date = new Date(Long.parseLong(lastLoginTime.getValue()));  // 更新cookie  lastLoginTime.setValue(String.valueOf(System.currentTimeMillis()));  resp.addCookie(lastLoginTime);    resp.getWriter().println("欢迎您, " + userName + ",上次登录时间是:" + date);  }  }else{  // 向客户端输出登录失败  resp.getWriter().println("登录失败");  }  }    @Override  protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  doGet(req, resp);  }  } |

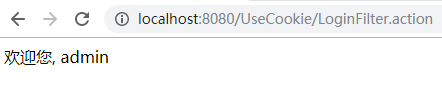
6.编写web.xml

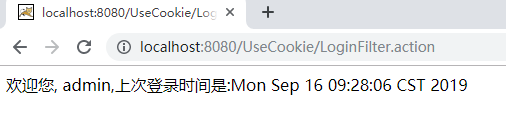


7.部署运行



8.点击登录, 首次登录没有上次登录时间, 之后登录有上次登录时间





1. Cookie的安全问题

1.由于Cookie会保存在客户端上, 所以有安全隐患问题。还有一个问题, Cookie的大小与个数有限制。为了解决这个问题, 就出现了Session 。