## [实现Web程序的自动登录](http://www.cnblogs.com/nokiaguy/archive/2008/06/05/1214695.html)

**本文为原创，如需转载，请注明作者和出处，谢谢！**

有很多Web程序中第一次登录后，在一定时间内（如2个小时）再次访问同一个Web程序时就无需再次登录，而是直接进入程序的主界面（仅限于本机）。实现这个功能关键就是服务端要识别客户的身份。而用Cookie是最简单的身从验证。

如果用户第一次登录，可以将用户名作为Cookie写到本地，代码如下：

Cookie cookie = new Cookie("user", user);  
cookie.setMaxAge(365 \* 24 \* 3600);  
cookie.setPath("/");  
response.addCookie(cookie);

    当用户再次访问程序时，服务端程序应该检测这个Cookie是否存在，代码如下：

Cookie[] cookies=request.getCookies();  
for(Cookie cookie: cookies)  
{  
    if(cookie.getName().equals(user))  
    {  
        // 如果user Cookie存在，进行处理  
        break;  
    }  
}

    尽管从客户端可以获得User Cookie，但这上Cookie可能存在很长时间，而且仅凭这个Cookie就自动登录并不安全，因此，可以在服务端使用一个Session来管理用户。也就是当第一次登录成功后，就创建一个Session，并将用户的某些信息保存在Session顺。代码如下：

HttpSession session =request.getSession();  
session.setAttribute(user, user);  
session.setMaxInactiveInterval(2 \* 3600);  // Session保存两小时

     当再次访问程序时，确定了cookie存在后，就会继续验证User Session的存在，代码如下：

Cookie[] cookies=request.getCookies();  
for(Cookie cookie: cookies)  
{  
    if(cookie.getName().equals(user))  
    {  
        if(session.getAttribute(user) != null)  
        {  
            // 直接forward到主界面  
           break;  
        }  
        else  
        {    
            // forward到登录界面  
         }  
     }  
}

    虽然上面的代码可以很好地实现自动登录功能，但是当浏览器关闭，再次启动后，由于Servlet用于保存Session ID的JSESSIONID Cookie是临时的（也就是说不是持久Cookie，当浏览器关闭后，这个Cookie就会被删除），因此，需要将JSESSIONID进行持久化。代码如下：

HttpSession session = request.getSession();  
session.setAttribute(user, user);  
session.setMaxInactiveInterval(2 \* 3600);  // Session保存两小时  
Cookie cookie = new Cookie("JSESSIONID", session.getId());  
cookie.setMaxAge(2 \* 3600);  // 客户端的JSESSIONID也保存两小时  
session.setMaxInactiveInterval(interval)  
cookie.setPath("/");          
response.addCookie(cookie);

如果使用上面的代码，即使浏览器关闭，在两小时之内，Web程序仍然可以自动登录。

如果我们自已加一个JSESSIONID Cookie，在第一次访问Web程序时，HTTP响应头有两个JSESSIONID，但由于这两个JSESSIONID的值完全一样，因此，并没有任何影响。如果在响应头的Set-Cookie字段中有多个相同的Cookie，则按着path和name进行比较，如果这两个值相同，则认为是同一个Cookie，最后一个出现的Cookie将覆盖前面相同的Cookie，如下面的两个Cookie，最后一个将覆盖前一个：

Set-Cookie: JSESSIONID=DDB2274CAC6739E1D6747B0043D5D90E; Path=/web

Set-Cookie: JSESSIONID=mysession; Expires=Thu, 05-Jun-2008 05:02:50 GMT; Path=/web

    由于下面两个Cookie的path不同，因此，它们是完全不同的两个Cookie：

Set-Cookie: JSESSIONID=DDB2274CAC6739E1D6747B0043D5D90E; Path=/web1

Set-Cookie: JSESSIONID=mysession; Expires=Thu, 05-Jun-2008 05:02:50 GMT; Path=/web2