用Servlet控制会话

1.

会话状态的维持是开发Web应用所必须面对的问题，有多种方法可以来解决这个问题，如使用Cookies，或直接把状态信息加到URL中等，还有Servlet本身提供了一个HttpSession接口来支持会话状态的维持 。

Session的发明是为了填补HTTP协议的局限。

从服务器这端来看，每一个请求都是独立的，因此HTTP协议被认为是**无状态**协议，当用户在多个主页间切换时，服务器无法知道他的身份。Session的出现就是为了弥补这个局限。利用Session，您就可以当一个用户在多个主页间切换的时候也能保存他的信息。这样很多以前根本无法去做的事情就变得简单多了。  
    在访问者从到达某个特定的主页到离开为止的那段时间，**每个访问者都会单独获得一个Session，服务端根据请求的Cookie字段中的JSESSIONID找到对应的session（调用request.getSession的时候）**

**2.**

Java Servlet定义了一个**HttpSession**接口，实现的Session的功能，在Servlet中使用Session的过程如下：  
　　（1） 使用HttpServletRequest的getSession方法得到当前**存在的**session，如果当前没有定义session，则创建一个新的session，还可以使用方法getSession（true）  
　　（2） **写session变量**。可以使用方法HttpSession.**setAttribute**（name，value）来向Session中存储一个信息。  
　　（3） 读Session变量。可以使用方法HttpSession.getAttribute（name）来读取Session中的一个变量值，如果name是一个没有定义的变量，那么返回的是null。需要注意的是，**从getAttribute读出的变量类型是Object，必须使用强制类型转换**，比如：String uid = (String) session.getAttribute("uid");  
　　（4） **关闭session，**当时用完session后，可以使用**session.invalidate()**方法关闭session。**但是这并不是严格要求的。因为，Servlet引擎在一段时间之后，自动关闭seesion。**  
      HttpSession session = request.getSession(true); //**参数true是指在没有session时创建一个新的**  
　　　Date created = new Date(session.getCreationTime()); //得到session对象创建的时间  
　　　out.println("ID " + session.getId()+"<br>"); //得到该session的id，并打印  
　　　out.println("Created: " + created+"<br>");//打印session创建时间  
　　　session.setAttribute("UID","12345678"); //在session中添加变量UID=12345678  
　　　session.setAttribute("Name","Tom"); //在session中添加变量Name=Tom 　10.2.4