Java ee Servlet session

 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，hidden类型的表单域，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，在这里我们主要介绍基于这个接口的会话状态的管理。

　　Session的发明是为了填补HTTP协议的局限。请注意HTTP协议是如何工作的--用户发出请求，服务器作出响应，这种用户端和服务器端的联系就是离散的，非连续的。HTTP协议不能提供允许服务器跟踪用户请求的功能。在服务器端完成响应用户的请求之后，服务器不能继续与该浏览器继续保持连接。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。

[Java](http://java.chinaitlab.com/) 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。也可以使用HttpSession.putValue（name，value），但这个方法已经过时了。

　　（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");

　　也可以使用HttpSession.getValue（name），但是这个方法也已经过时了。

　　（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。

　　下面举一个简单的例子说明session的使用

　　//97色色 SessionExample.java

　　import java.io.\*;

　　import java.util.\*;

　　import javax.servlet.\*;

　　import javax.servlet.http.\*;

　　//导入必要的软件包

　　public class SessionExample extends HttpServlet

　　{

　　public void doGet(HttpServletRequest request, HttpServletResponse response)

　　throws IOException, ServletException //实现doGet方法

　　{

　　response.setContentType("text/html"); //设置HTTP头

　　PrintWriter out = response.getWriter(); //得到输出97gan

　　HttpSession session = request.getSession(true);

　　//得到session对象

　　//打印HTML标记

　　out.println("<html>");

　　out.println("<head>");

　　out.println("<meta http-equiv=\"Content-Type\" content=\"text/html; charset=gb2312\">");

　　out.println("</head>");

　　out.println("<body>");

　　Date created = new Date(session.getCreationTime());

　　//得到session对象创建的时间

　　Date accessed = new Date(session.getLastAccessedTime());

　　//得到最后访问该session对象的时间

　　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

　　out.println("Last Accessed: " + accessed+"<br>");

　　//打印最后访问时间

　　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

　　Enumeration e = session.getAttributeNames();

　　//得到session中变量名的枚举对象

　　while (e.hasMoreElements()) { //遍历每一个变量

　　String name = (String)e.nextElement(); //首先得到名字

　　String value = session.getAttribute(name).toString();

　　//由名字从97gan中得到值

　　out.println(name + " = " + value+"<br>"); //打印

　　}

　　out.println("</body>"); //打印HTML标记

　　out.println("</html>");

　　}

　　}

　　}