

servlet中session的使用 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/646/2021_2022_servlet_E4_B8_AD_c97_646663.htm 还是由于HTTP协议连接的无状态性，才使得session的不得已而产生。既然Web应用并不了解有关同一用户以前请求的信息，那么解决这个问题的一个办法是使用Servlet/JSP容器提供的会话跟踪功能，Servlet API规范定义了一个简单的HttpSession接口，通过它我们可以方便地实现会话跟踪。HttpSession接口提供了存储和返回标准会话属性的方法。标准会话属性如会话标识符、应用数据等，都以“名字-值”对的形式保存在服务器端。也就是说，HttpSession接口提供了一种把对象保存到内存、在同一用户的后继请求中提取这些对象的标准办法。在会话中保存数据的方法是setAttribute(String s, Object o)，从会话提取原来所保存对象的方法是getAttribute(String s)。在服务器端，每当新用户请求一个使用了HttpSession对象的JSP页面，Servlet/JSP容器除了发回应答页面之外，它还要以cookie的形式向浏览器发送一个特殊的数字。这个特殊的数字称为“会话标识符”，它是一个唯一的用户标识符。此后，HttpSession对象就驻留在内存之中(这当然是在服务器端)，浏览器再请求session时，服务器会先得到它的sessionid,再作处理。在客户端，浏览器保存会话标识符，并在每一个后继请求中把这个会话标识符发送给服务器。会话标识符告诉JSP容器当前请求不是用户发出的第一个请求，服务器以前已经为该用户创建了HttpSession对象。此时，JSP容器不再为用户创建新的HttpSession对象，而是寻找具有相同会话标识符的HttpSession对象，然后建立该HttpSession

对象和当前请求的关联。会话标识符以Cookie的形式在服务器和浏览器之间传送，标准会话属性在服务器端也是以会话的形式存在的，并且这个Cookie的生命周期只是临时的，即会话结束后就自动消失，没有为它指定固定的生命周期，因此有人说session是基于Cookie的技术。另外，如果客户端不支持Cookie，运用url重写机制来保证会话标识符传回服务器。还有一点，session不像Cookie那样拥有路径访问的问题。session对应的是窗口，只要是同一个客户端或者是存在父子关系的多个客户端，同一个application下的servlet/JSP可以共享同一个session。当然，session和窗口的对应关系也是受时间限制的，至于多长时间，可以在服务器的conf/web.xml中配置

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com