深入理解和改进J _Servlet会话管理机制 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/267/2021_2022__E6_B7_B1_E5_85_A5_E7_90_86_E8_c104_267110.htm 在Web服务器端编程中，会话状态管理是一个经常必须考虑的重要问题。本文分析JSP/Servlet的会话管理机制及其所面临的问题，然后提出了一种改进的会话管理方法。 一、Servlet的会话管理机制 根据设计，HTTP是一种无状态的协议。它意味着Web应用并不了解有关同一用户以前请求的信息。维持会话状态信息的方法之一是使用Servlet或者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。简而言之，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HTTP协议中，当用户不再活动时不存在显式的终止信号。由于这个原因，我们不知道用户是否还要再次返回，如果不采取某种方法解决这个问题，内存中会积累起大量的HttpSession对象。 为此，Servlet采用“ 超时限制 ”的办法来判断用户是否还在访问：如果某个用户在一定的时间之内没有发出后继请求，则该用户的会话被作废，他的HttpSession对象被释放。会话的默认超时间隔由Servlet容器定义。这个值可以通过getMaxInactiveInterval方法获得，通过setMaxInactiveInterval方法修改，这些方法中的超

时时间以秒计。如果会话的超时时间值设置成-1，则会话永不超时。Servlet可以通过getLastAccessedTime方法获得当前请求之前的最后一次访问时间。 要获得HttpSession对象，我们可以调用HttpServletRequest对象的getSession方法。为了正确地维持会话状态，我们必须在发送任何应答内容之前调用getSession方法。 用户会话既可以用手工方法作废，也可以自动作废。作废会话意味着从内存中删除HttpSession对象以及它的数据。例如，如果一定时间之内（默认30分钟）用户不再发送请求，Java Web Server自动地作废他的会话。 Servlet/JSP会话跟踪机制有着一定的局限，比如： 会话对象保存在内存之中，占用了可观的资源。 会话跟踪依赖于Cookie。由于各种原因，特别是安全上的原因，一些用户关闭了Cookie。 会话跟踪要用到服务器创建的会话标识符。在多个Web服务器以及多个JVM的环境中，Web服务器不能识别其他服务器创建的会话标识符，会话跟踪机制无法发挥作用。 要深入理解会话跟踪机制，首先我们必须理解在Servlet/JSP容器中会话如何运作。 二、会话标识符 每当新用户请求一个使用了HttpSession对象的JSP页面，JSP容器除了发回应答页面之外，它还要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，等待同一用户返回时再次调用它的方法。 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。如果浏览器不支持Cookie又如何呢？此时，对服务器的后继请求将不会带有会话标识符。结果，JSP容器认为该请求来自一个新用户，它会再创建一个HttpSession对象，而以前创建的HttpSession对象仍旧驻留在内存中，但该用户以前的会话信息却丢失了。 另外，Servlet/JSP容器只认可它自己创建的会话标识符。如果同一Web应用在“ Web农场 ”（Web farm）的多台服务器上运行，则必须存在这样一种机制：保证来自同一用户的请求总是被定向到处理该用户第一次请求的服务器。 三、伪会话管理机制 如前所述，基于Cookie的会话管理技术面临着种种问题。下面我们要设计一种新的会话管理机制来解决这些问题。这种会话管理机制称为“ 伪会话 ”（Pseudo Session）机制，它具有如下特点： 1) 对象和数据不是保存在内存中，而是以文本文件形式保存。每一个文本文件与一个特定的用户关联，文件的名字就是会话的标识符。因此，文件名字必须是唯一的。 2) 文本文件保存在一个专用的目录中，所有Web服务器都可以访问这个目录。因此，伪会话可以用于Web农场。 3) 会话标识符不作为Cookie发送，而是直接编码到URL里面。因此，采用伪会话技术要求修改所有的超级链接，包括HTML表单的ACTION属性。 此外，实现伪会话管理机制时我们还要考虑到以下几点： 4) 它应该与应用无关，其他想要实现同样功能的开发者应该能够方便地重用它。 5) 考虑到安全原因，应该有一种为会话标识符生成随机数字的办法。 6) 为了作废过期的会话，应该设定一个超时值。同一个用户，如果他超过

一定的时间之后再次返回，他将获得一个新的会话标识符。此举能够防止未经授权的用户冒用其他人的会话。 7) 应该有一种收集过期会话并删除相应文本文件的机制。 8) 如果用户使用已经过期的会话标识符再次访问服务器，即使这个会话标识符的文本文件还没有删除，系统也不应该允许用户使用原来的会话。 9) 同时，应该存在一种更新会话文本文件最后改动时间的机制，使得用户在会话过期时限之前返回时会话总是保持最新且合法的状态数据。 