利用clickstream跟踪用户

|  |
| --- |
| 项目主页：<http://www.opensymphony.com/clickstream/> 具体的例子可以看clickstream-1.0.2.war  **Clickstream 概述**  一个用来跟踪正在你的站点上访问的用户所到达位置细节的工具。它允许你跟踪访问你的站点的“点击流”或者“传输路径”。请访问JIRA的[更新日志](http://jira.opensymphony.com/secure/BrowseProject.jspa?id=10012&report=changelog) 来了解Clickstream最近的发展。  **特性**  • 当用户会话产生时，开始跟踪点击流。(通过一个[监听器](http://java.sun.com/products/servlet/2.3/javadoc/javax/servlet/ServletContextListener.html)) • 跟踪用户产生的每次点击信息。(通过一个[过滤器](http://java.sun.com/products/servlet/2.3/javadoc/javax/servlet/Filter.html)) • 当用户会话终止时，将完整的点击流记录到文件或者PrintStream中。 • 设法发现用户是不是机器人，并进行适当的过滤（目前可以检测252种机器人）  **安装**  把[clickstream-1.0.2.jar and commons-logging.jar](http://www.opensymphony.com/clickstream/download.html)放到[web应用程序根目录]/WEB-INF/lib里，把下面的代码加到[web应用程序根目录]/WEB-INF/web.xml里面：  <filter>       <filter-name>clickstream</filter-name>       <filter-class>com.opensymphony.clickstream.ClickstreamFilter</filter-class> </filter>  <filter-mapping>       <filter-name>clickstream</filter-name>       <url-pattern>\*.jsp</url-pattern> </filter-mapping>  <filter-mapping>       <filter-name>clickstream</filter-name>       <url-pattern>\*.html</url-pattern> </filter-mapping>  <listener> <listener-class>**com.opensymphony.clickstream.ClickstreamListener**</listener-class> </listener>  根据上面的配置，Clickstream会跟踪所有对jsp或者html文件的请求.无论什么时候，只要一个HttpSession失效了，对那个会话的 完整点击流就会被记录。如果你想跟踪对其他类型的点击(例如.txt或者pdf文件)，只需要添加更多的filter-mapping就可以了。  下面是一个完整的web.xml的例子  <?xml version="1.0" encoding="ISO-8859-1" ?>      <!DOCTYPE web-app (View Source for full doctype...)>      <web-app>     <display-name>Opensymphony Clickstream Example</display-name>      <filter>       <filter-name>clickstream</filter-name>        <filter-class>com.opensymphony.clickstream.ClickstreamFilter</filter-class>      </filter>     <filter-mapping>       <filter-name>clickstream</filter-name>        <url-pattern>/\*</url-pattern>      </filter-mapping>      <listener>       <listener-class>**com.opensymphony.clickstream.ClickstreamListener**</listener-class>     </listener>     </web-app>  你可以利用clickstreams.jsp  <%@ page import="java.util.\*,                    com.opensymphony.clickstream.Clickstream" %>  <%       Map clickstreams = (Map) application.getAttribute("clickstreams");        String showbots = "false";       if ("true".equalsIgnoreCase(request.getParameter("showbots")))           showbots = "true";       else if ("both".equalsIgnoreCase(request.getParameter("showbots")))           showbots = "both"; %> <html>       <head>           <title>All Clickstreams</title>       </head>            <body>           <h1>All Clickstreams</h1>            <a href="?showbots=false">No Bots</a> |           <a href="?showbots=true">All Bots</a> |           <a href="?showbots=both">Both</a>                    <p>                   <% if (clickstreams.isEmpty()) { %>             No clickstreams in progress.           <% } else {               synchronized(clickstreams) {                   Iterator it = clickstreams.keySet().iterator();                   int count = 0;                   while (it.hasNext())                   {                       String key = (String)it.next();                       Clickstream stream = (Clickstream)clickstreams.get(key);                        if (showbots.equals("false") && stream.isBot())                       {                           continue;                       }                       else if (showbots.equals("true") && !stream.isBot())                       {                           continue;                       }                        count++;                       try {                   %>                   <%= count %>. <a href="viewstream.jsp?sid=<%= key %>"><b><%= (stream.getHostname() != null &&   !stream.getHostname().equals("") ? stream.getHostname() : "Stream") %></b></a> <font size="-1">[<%= stream.getStream().size()   %> reqs]</font><br>                   <%                       }                       catch (Exception e)                       {                   %>                       An error occurred - <%= e %><br>                   <%                       }                   }               }           }           %>           </p>       </body> </html>  和viewstream.jsp来显示系统当前的Clickstream活动。  <%@ page import="java.util.\*,                    com.opensymphony.clickstream.Clickstream,                    com.opensymphony.clickstream.ClickstreamRequest" %>  <% if (request.getParameter("sid") == null) {     response.sendRedirect("clickstreams.jsp");     return; }  Map clickstreams = (Map)application.getAttribute("clickstreams");  Clickstream stream = null;  if (clickstreams.get(request.getParameter("sid")) != null) {     stream = (Clickstream)clickstreams.get(request.getParameter("sid")); }  if (stream == null) {     response.sendRedirect("clickstreams.jsp");     return; } %>  <html>       <head>           <title>Clickstream for <%= stream.getHostname() %></title>       </head>            <body>           <div align="right"><a href="clickstreams.jsp">All streams</a></div>            <h1>Clickstream for <%= stream.getHostname() %></h1>            <b>Initial Referrer</b>: <a href="<%= stream.getInitialReferrer() %>"><%= stream.getInitialReferrer() %></a><br>           <b>Hostname</b>: <%= stream.getHostname() %><br>           <b>Session ID</b>: <%= request.getParameter("sid") %><br>           <b>Bot</b>: <%= stream.isBot() ? "Yes" : "No" %><br>           <b>Stream Start</b>: <%= stream.getStart() %><br>           <b>Last Request</b>: <%= stream.getLastRequest() %><br>                    <% long streamLength = stream.getLastRequest().getTime() - stream.getStart().getTime(); %>           <b>Session Length</b>:             <%= (streamLength > 3600000 ?               " " + (streamLength / 3600000) + " hours" : "") +             (streamLength > 60000 ?               " " + ((streamLength / 60000) % 60) + " minutes" : "") +             (streamLength > 1000 ?               " " + ((streamLength / 1000) % 60) + " seconds" : "") %><br>                    <b># of Requests</b>: <%= stream.getStream().size() %>                    <p><b>Click stream</b>:</p>                    <table border="0" cellpadding="2">           <%           synchronized(stream) {               Iterator clickstreamIt = stream.getStream().iterator();                int count = 0;               while (clickstreamIt.hasNext())               {                   count++;                   String click = ((ClickstreamRequest)clickstreamIt.next()).toString();               %>               <tr><td><%= count %>:</td><td><a href="http://<%= click %>"><%= click %></a></td></tr>               <%               }           }           %>           </table>       </body> </html>  **配置**  日志功能是非常基本的(你可以与[log4j集成](http://jakarta.apache.org/commons/logging/guide.html))。如果你需要更多更高级的日志功能，那就需要实现[com.opensymphony.clickstream.logger.ClickstreamLogger](http://www.opensymphony.com/clickstream/api/com/opensymphony/clickstream/logger/ClickstreamLogger.html)接口并且提供任何你所希望的持久化方法。  默认情况下，Clickstream使用[com.opensymphony.clickstream.logger.SimpleClickstreamLogger](http://www.opensymphony.com/clickstream/api/com/opensymphony/clickstream/logger/SimpleClickstreamLogger.html)这个类来处理日志。这个logger只能向Jakarta的[Commons Logging](http://jakarta.apache.org/commons/logging.html)工具发送一个INFO日志声明，Commons Logging可以依次通知多个日志provider，包括[log4j](http://logging.apache.org/log4j/docs/)。  可以通过创建一个clickstream.xml配置文件并且把它放在你应用程序的类路径中（通常是[web应用程序根目录]/WEB-INF/classes）来改变这种默认的行为。配置文件可能是如下的形式：  <clickstream>       <!-- there can only be one logger -->       <logger class="com.opensymphony.clickstream.logger.SimpleClickstreamLogger"/>        <!-- there can be zero or more bot-hosts -->       <bot-host name="someBotHost"/>        <!-- there can be zero or more bot-agents -->       <bot-agent name="someBotAgent"/> </clickstream>  如果你希望提供你自己的logger，请查看Clickstream 的[javadoc文档](http://www.opensymphony.com/clickstream/api/index.html)。  **和容器有关的特定问题**  当从控制台停止Clickstream应用程序时，BEA WebLogic 8.1 (SP2)会抛出NullPointerException异常(when setting the attribute NULL for the clickstreams in the ServletContext on contextDestroyed—**这句话不会翻译，谁能帮帮忙**)，需要应用CR133558这个补丁(已经包含在weblogic的sp3中)。  **外部文档** • [使用EAServer跟踪你的用户](http://www.opensymphony.com/clickstream/visitor_tracking.pdf)—Berndt Hamboeck所著的Sybase白皮书 • [Servlet2.3 模型的过滤器代码](http://www.javaworld.com/javaworld/jw-06-2001/jw-0622-filters-p2.html)—Jason Hunter所著的“谁在你的站点上，他们正在干什么”的一段。 |