

Web应用开发实验报告

**(2018级)**



实验题目 实验8 Servlet高级应用

姓名

学号

专业 软件工程 班

学院 计算机科学与技术学院

提交日期 2020年6月21日

**目录**

第1题 [3](#_Toc32707)

[1.1运行结果 3](#_Toc26873)

[1.2问题及解决方案 3](#_Toc29751)

[1.3源代码 3](#_Toc17543)

[第2题 5](#_Toc26985)

[2.1运行结果 5](#_Toc13696)

[2.2问题及解决方案 6](#_Toc21387)

[2.3源代码 7](#_Toc8522)

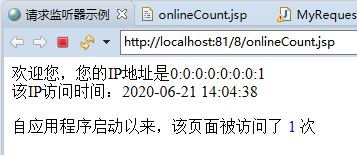
**实验8 Servlet高级应用**

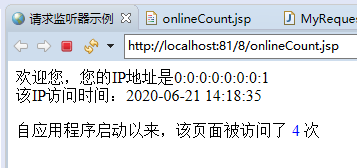
1、修改教材P254的例子程序8.3和程序8.4，实现客户端访问onlineCount.jsp的次数，并将客户端的访问记录（包括ip、访问时间）保存至数据库中，要求同一个IP只能记录一次。

提示：用户访问的IP可存到List或数据库表中，若有用户访问onlineCount.jsp，则在监听器MyRequestListener中得到客户端IP，若该IP已在List对象或数据库中，则不增加访问次数，否则访问次数加1，并将该IP加到List对象或数据库中。测试时，用两台电脑或者手机与电脑组成一个网络，通过IP访问页面。

**1.1运行结果**

(1)请求监听器实例页面





**1.2问题及解决方案**

不知道怎么用两台电脑或者手机与电脑组成一个网络，通过IP访问页面。

**1.3源代码**

1. MyRequestListener.java

**package** com.listener;

**import** java.util.Date;

**import** java.text.SimpleDateFormat;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.ServletRequestEvent;

**import** javax.servlet.ServletRequestListener;

**import** javax.servlet.annotation.WebListener;

@WebListener

**public** **class** MyRequestListener **implements** ServletRequestListener{

**private** **int** count = 0;

**public** **void** requestInitialized(ServletRequestEvent re){

HttpServletRequest request= (HttpServletRequest)re.getServletRequest();

**if**(request.getRequestURI().endsWith("onlineCount.jsp")){

//将客户端的访问记录（包括IP、访问时间）保存至数据库中，要求同一个IP只能记录一次

//得到客户端IP，若该IP已在List对象或数据库中，则不增加访问次数，否则访问次数加1，并将该IP加到List对象或数据库中

count++;

String requestUrlIP = request.getServerName();

String userIpAddr = request.getRemoteAddr();

System.***out***.println("\*\*\*访问的Url中的服务器IP："+requestUrlIP);

System.***out***.println("\*\*\*用户客户端的IP地址："+userIpAddr);

Date t = **new** Date();

SimpleDateFormat df = **new** SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

String time=df.format(t);

System.***out***.println(time);

System.***out***.println(t);

re.getServletContext().setAttribute("time",time);

re.getServletContext().setAttribute("count",count);

}

}

**public** **void** requestDestroyed(ServletRequestEvent re){

}

}

1. OnlineCount.jsp

<%@ page contentType=*"text/html;charset=UTF-8"* %>

<html>

<head><title>请求监听器示例</title></head>

<body>

欢迎您，您的IP地址是${pageContext.request.remoteAddr}<br>

该IP访问时间：${applicationScope.time}<br>

<p>自应用程序启动以来，该页面被访问了

<font color=*"blue"*>${applicationScope.count}

</font>次<br>

</body>

</html>

2、编写测试的例子，实现教材P279第8题的过滤器，其中过滤器FilterOne 、FilterTwo和FilterThree类中的doFilter方法中分别输出“这是过滤器FilterOne ”、“这是过滤器FilterTwo ”、“这是过滤器FilterThree ”。另，如何能运行过滤器FilterOne，并写程序进行测试。

提示：

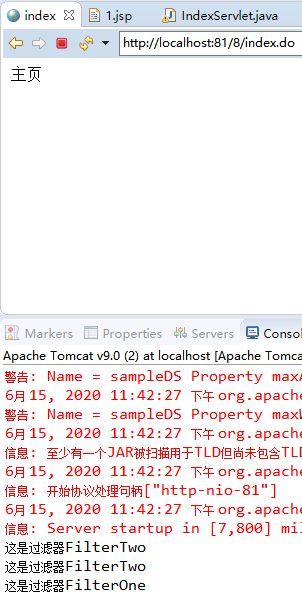
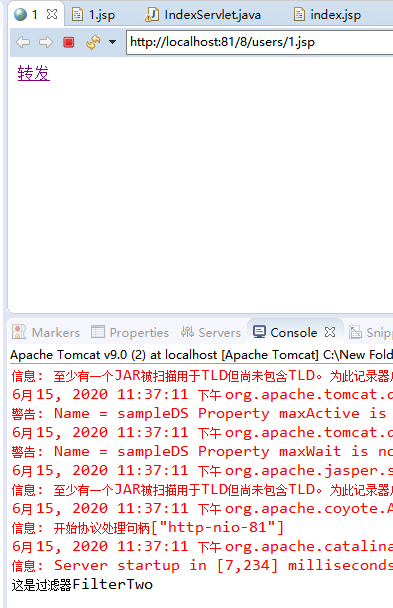
(1)编写3个过滤器类，doFilter中打印该过滤器名，并调用过滤器类

(2)在web.xml或者用@WebFilter中写这几个过滤器的映射

(3)编写/admin/index.jsp并允许测试

**2.1运行结果**

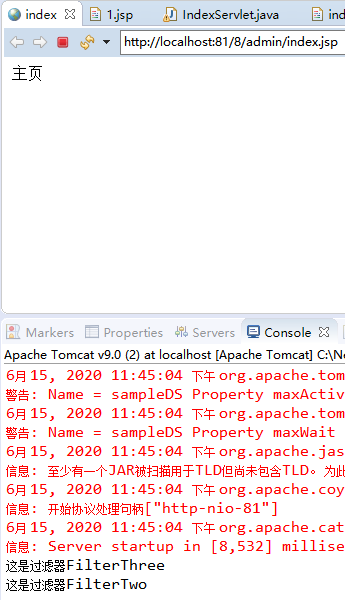
(1)运行/users/1.jsp的结果 (2)转发到/admin/index.jsp的结果



浏览器直接请求/users/1.jsp资源， 浏览器转发访问/admin/index.jsp，过滤过滤器FilterTwo会被执行。 器FilterOne会被执行,访问所有资源 时都会被执行的FilterTwo也会被执 行。

1. 运行/admin/index.jsp的结果

浏览器直接请求/admin/index.jsp资源，过滤器FilterThree、FilterTwo会被执行。



**2.2 问题及解决方案**

**问题**：转发时用超链接直接转发到/admin/index.jsp页面没有运行出FilterOne的结果；

**原因**：dispatcherTypes={DispatcherType.FORWARD}中的FORWARD表示过滤器应用在与调用RequestDispatcher的forward()匹配的请求，而不是直接用超链接转发；

**解决方案**：改成超链接先转发到Servlet处理，在Servlet的doGet()方法中用

RequestDispatcher rd = getServletContext().getRequestDispatcher("/admin/index.jsp");

rd.forward(request, response);

实现转发。

**2.3源代码**

(1)FilterOne.java(FilterTwo.java、FilterThree.java)

**package** com.Filter;

**import** java.io.IOException;

**import** javax.servlet.\*;

**import** javax.servlet.annotation.WebFilter;

@WebFilter(filterName="FilterOne",urlPatterns="/admin/\*",dispatcherTypes={DispatcherType.***FORWARD***})//只有转发访问index.jsp时，该过滤器才会被执行

//@WebFilter(filterName="FilterTwo",urlPatterns= {"/users/\*","/\*"})

//@WebFilter(filterName="FilterThree",urlPatterns="/admin/\*")

**public** **class** FilterOne **implements** Filter {

**public** **void** destroy() {

//销毁时调用

}

**public** **void** doFilter(ServletRequest request, ServletResponse response, FilterChain chain) **throws** IOException, ServletException {

//过滤方法 主要是对request和response进行一些处理,然后交给下一个过滤器或Servlet处理

System.***out***.println("这是过滤器FilterOne");

// System.out.println("这是过滤器FilterTwo");

// System.out.println("这是过滤器FilterThree");

chain.doFilter(request, response);//交给下一个过滤器或servlet处理

}

**public** **void** init(FilterConfig fConfig) **throws** ServletException {

//初始化方法 接收一个FilterConfig类型的参数 该参数是对Filter的一些配置

}

}

1. /admin/index.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html><head>

<meta charset=*"UTF-8"*>

<title>index</title>

</head><body>

主页

</body></html>

1. /users/1.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html><head>

<meta charset=*"UTF-8"*>

<title>1</title>

</head><body>

<a href=*"../index.do"*>转发</a>

</body></html>

1. IndexServlet.java

@WebServlet("/index.do")

**public** **class** IndexServlet **extends** HttpServlet {

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

RequestDispatcher rd = getServletContext().getRequestDispatcher("/admin/index.jsp");

rd.forward(request, response);

}

}