**实验七 访问权限过滤与在线用户列表**

## **实验目标**

本实验任务实现“Q-ITOffer”锐聘网站后台管理系统的访问权限过滤、在线用户列表功能：

* 【实验7-1】 访问权限过滤。
* 【实验7-2】 在线用户列表。

## **【实验7-1】访问权限过滤**

### 知识点回顾

* Filter接口
* FilterChain接口
* 过滤器声明配置

### 功能描述

实现“Q-ITOffer”锐聘网站后台管理系统用户在未登陆的情况下对系统资源的访问权限过滤功能。

### 实验分析

1. 对于锐聘网站后台管理系统，除登陆请求处理外，其它的请求均需要在用户成功登陆后才可访问，如果单独在每个请求中进行登陆状态的判断将非常繁琐，因此可以使用过滤器统一拦截验证。在本项目中，对manage文件夹下的所有资源的请求以及对除用户登陆功能的Servlet请求以外的所有Servlet请求均需要通过过滤器进行拦截验证。

### 实验步骤

1. 在Eclipse工具中创建Java Web项目“Q\_ITOffer\_Experiment07”，导入【实验六】中的所有资源，使每个后续实验都能够保证前期实验的所有功能的实现。
2. 在项目下首先创建包com.qst.itofferbacker.filter，然后创建过滤器AuthorityFilter.java，声明配置上需要过滤拦截的资源，随后对在登陆成功时才会保存的会话域属性进行判断，从而确定该访问者是否有此请求的访问权限，过滤器具体实现代码如下所示。

**【代码7- 1】AuthorityFilter.java**

package com.qst.itofferbacker.filter;

/\*\*

\* 用户访问权限过滤器

\* @公司 青软实训

\* @作者 fengjj

\*/

@WebFilter(

urlPatterns = { "/manage/\*" },

servletNames = {"com.qst.itofferbacker.servlet.CompanyAddServlet",

"com.qst.itofferbacker.servlet.CompanyServlet",

"com.qst.itofferbacker.servlet.ResumeServlet",

"com.qst.itofferbacker.servlet.UserAddServlet",

"com.qst.itofferbacker.servlet.UserServlet"},

initParams = { @WebInitParam(name = "loginPage", value = "login.html")})

public class AuthorityFilter implements Filter {

private String loginPage;

public AuthorityFilter() {

}

public void init(FilterConfig fConfig) throws ServletException {

// 获取当请求被拦截时转向的页面

loginPage = fConfig.getInitParameter("loginPage");

if (null == loginPage) {

loginPage = "login.html";

}

}

public void destroy() {

this.loginPage = null;

}

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

HttpServletRequest req = (HttpServletRequest) request;

HttpServletResponse resp = (HttpServletResponse) response;

HttpSession session = req.getSession();

// 判断如果是登录请求则放行

if("/UserServlet".equals(req.getServletPath()) &&

"type=login".equals(req.getQueryString())){

chain.doFilter(request, response);

return;

}

// 判断被拦截的请求用户是否处于登录状态，若未登录，返回登录页面

if (session.getAttribute("SESSION\_USER") == null) {

resp.sendRedirect(req.getContextPath() + "/" + loginPage);

} else {

chain.doFilter(request, response);

}

}

}

## **【实验7-2】在线用户列表**

### 知识点回顾

* 监听器的作用
* HttpSessionListener监听器

### 功能描述

实现“Q-ITOffer”锐聘网站后台管理系统当前在线用户的列表查询功能。

### 实验分析

1. 在线用户是指在进行在线用户查询时已成功登陆系统、正在使用系统、还未退出系统的用户，通过实现HttpSessionAttributeListener监听器，在当一个用户登陆成功、向会话域添加属性时将用户的信息存入应用域属性中；通过实现HttpSessionListener监听器，在用户退出、会话对象失效时将用户对象从应用域属性中移除。
2. 通过在线用户列表显示页面userOnline.jsp从应用域属性中将获取、显示用户的信息。

### 实验步骤

1. 在Q\_ITOffer\_Experiment07项目中创建包com.qst.itofferbacker.listener用于存放过滤器，创建实现HttpSessionAttributeListener的监听器OnlineUsersListener.java，具体实现代码如下所示。

**【代码7- 2】OnlineUsersListener.java**

package com.qst.itofferbacker.listener;

/\*\*

\* 在线用户监听器

\* @公司 青软实训

\* @作者 fengjj

\*/

@WebListener

public class OnlineUsersListener implements HttpSessionAttributeListener {

public void attributeAdded(HttpSessionBindingEvent event) {

// 判断监听到的是否为记录用户信息的会话域属性

if("SESSION\_USER".equals(event.getName())){

// 获取会话域属性值：当前登录的用户对象

User user = (User)event.getValue();

// 获取应用上下文对象

ServletContext application = event.getSession().getServletContext();

// 获取存放在应用域属性中的在线用户信息列表

Map<String,User> onlineUserMap =

(Map<String,User>)application.getAttribute("ONLINE\_USER");

if(onlineUserMap == null)

onlineUserMap = new Hashtable<String,User>();

// 将当前登录的用户对象存入应用域属性：在线用户信息列表

onlineUserMap.put(user.getUserLogname(), user);

application.setAttribute("ONLINE\_USER", onlineUserMap);

}

}

}

1. 在包com.qst.itofferbacker.listener下创建实现HttpSessionListener的监听器LogoutUserListener.java，具体实现代码如下所示。

**【代码7- 3】OnlineUsersListener.java**

package com.qst.itofferbacker.listener;

import com.qst.itofferbacker.javabean.User;

/\*\*

\* 在线用户退出监听器

\* @公司 青软实训

\* @作者 fengjj

\*/

@WebListener

public class LogoutUserListener implements HttpSessionListener {

public void sessionDestroyed(HttpSessionEvent event) {

// 判断监听到的是否为记录用户信息的会话域属性

Enumeration<String> attrnames = event.getSession().getAttributeNames();

while(attrnames.hasMoreElements()){

System.out.println(attrnames.nextElement());

if("SESSION\_USER".equals(attrnames.nextElement())){

// 获取会话域属性值：当前登录的用户对象

User user = (User) event.getSession().getAttribute("SESSION\_USER");

// 获取应用上下文对象

ServletContext application =

event.getSession().getServletContext();

// 获取存放在应用域属性中的在线用户信息列表

Map<String,User> onlineUserMap =

(Map<String,User>)application.getAttribute("ONLINE\_USER");

if(onlineUserMap != null){

// 将当前登录的用户对象移出应用域属性：在线用户信息列表

onlineUserMap.remove(user.getUserLogname());

application.setAttribute("ONLINE\_USER", onlineUserMap);

}

}

}

}

}

1. 修改目录菜单页面中对在线用户列表查询的请求地址，实现代码如下所示。

**【代码7- 4】left.jsp**

<a href="**userOnline.jsp**" target="rightFrame">在线用户</a>

1. 在manage目录下通过模板userOnline.html创建在线用户列表显示JSP页面userOnline.jsp，从应用域中获取显示当前在线用户的信息，具体实现代码如下所示，页面运行效果如图7- 1所示。

**【代码7- 5】userOnline.jsp**

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

pageEncoding="UTF-8"%>

<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>用户在线列表</title>

<link href="../css/manageadmin.css" rel="stylesheet" type="text/css" />

</head>

<body>

<div class="place"> <span>位置：</span>

<ul class="placeul">

<li><a href="index.html">首页</a></li><li>用户在线列表</li>

</ul>

</div>

<div class="rightinfo">

<table class="imgtable">

<thead>

<tr>

<th ><input name="" type="checkbox" value="" checked="checked"/></th>

<th>用户登录名</th>

<th>用户真实姓名</th>

<th>用户角色</th>

<th>操作</th>

</tr>

</thead>

<tbody>

<c:forEach items="${applicationScope.ONLINE\_USER}" var="usermap">

<tr height="50px">

<td>${usermap.key}</td>

<td>${usermap.value.userRealname}</td>

<td><c:if test="${usermap.value.userRole==3}">普通用户</c:if>

<c:if test="${usermap.value.userRole==2}"> 企业管理员</c:if>

<c:if test="${usermap.value.userRole==1}">系统管理员</c:if></td>

<td><a

href="../UserServlet?type=forceLogout&userId=${usermap.value.userId }"

class="tablelink">强制下线</a></td>

</tr>

</c:forEach>

</tbody>

</table>

</div>

</body>

</html>

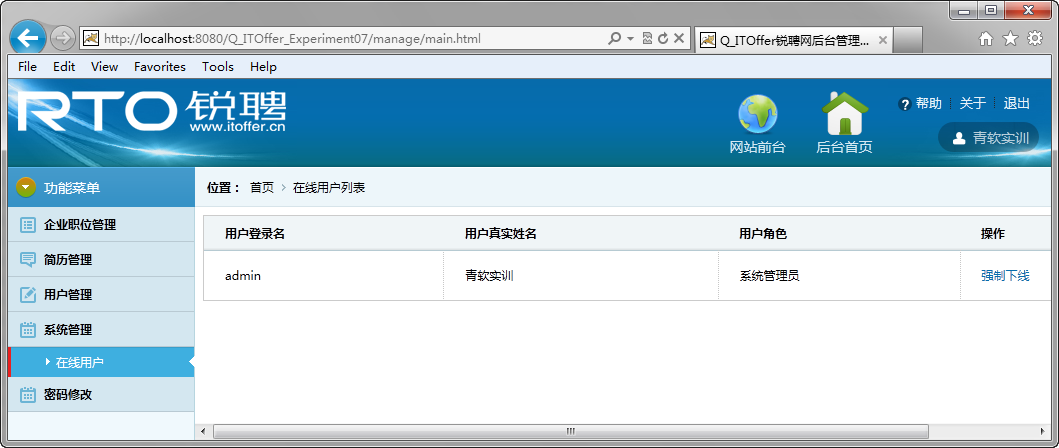


图7- 1 在线用户列表页面运行效果图