

实验报告

|  |  |
| --- | --- |
| 学 院： | 数计学院 |
| 专业班级： | 网络1701 |
| 学 号： | 1718064009 |
| 姓 名： | 王敏娟 |
|  |  |
| 实验课程： | Javaweb |
| 实验项目： | 基于过滤器的登录访问控制实现 |
| 实验类型： | 验证 |
| 实验学时： | 2 |
|  |  |
| 实验地点： |  |
| 实验日期： | 2020年4月26日 |

实验课程：

# 实验项目：基于过滤器的登录访问控制实现

# 1、实验目的

通过过滤器实现对与非登录用户访问资源的限制

# 2、实验环境

Idea + tomcat + jdk

# 3、实验内容与步骤

## 3.0 实验原理或方案

通过配置相应的过滤器实现在用户访问指定资源时，对用户的非法进行判断。从而限制非法用户的访问

## 3.1 步骤1

（1）编写静态登录页面

（2）代码

<1> index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>用户登录</title>

</head>

<body>

<form action="/login" method="post">

账户：<input type="text" name="account" placeholder="请输入账户" onblur="accountTest()">

<br>

密码：<input type="password" name="password" placeholder="请输入密码" onblur="passwordTest()">

<br>

&nbsp;&nbsp;<input type="submit" value="登录">

</form>

</body>

</html>

<2>success.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>登录成功</title>

</head>

<body>

<h1>恭喜你登录成功！</h1>

</body>

</html>

<3>error.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>错误页面</title>

</head>

<body>

<h1>用户名或密码错误！</h1>

</body>

</html>

## 3.2 步骤2

（1）编写处理登录的servlet

（2）代码

package com.danger.servlet;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import java.io.IOException;

/\*\*

\* 用户登录

\*/

@WebServlet("/login")

public class Login extends HttpServlet {

@Override

protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

req.setCharacterEncoding("utf-8");

String account = req.getParameter("account");

String password = req.getParameter("password");

if ("admin".equals(account) && "admin".equals(password)){

//密码正确

resp.sendRedirect(req.getContextPath() + "success.html");

//加入用户到会话

req.getSession().setAttribute("user",account);

}else{

//密码错误

resp.sendRedirect(req.getContextPath() + "error.html");

}

}

@Override

protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

this.doGet(req, resp);

}

}

## 3.3 步骤3

（1）编写过滤器

（2）代码

package com.danger.filter;

import javax.servlet.\*;

import javax.servlet.annotation.WebFilter;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import java.io.IOException;

@WebFilter("/\*")

public class LoginFilter implements Filter {

@Override

public void init(FilterConfig filterConfig) throws ServletException {

}

@Override

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

//判断用户是否登录

HttpServletRequest req = (HttpServletRequest) request;

HttpServletResponse resp = (HttpServletResponse) response;

response.setContentType("text/html;charset=utf-8");

//获取请求路径

String uri = req.getRequestURI();

if (uri.contains("index.html") || uri.contains("login")) {//正常登录

chain.doFilter(request, response);

} else {

Object user = req.getSession().getAttribute("user");//判断是否以登录

if (user != null) {

//已登录

chain.doFilter(request, response);

} else {

resp.getWriter().write("<font color='green' size='25'>用户未登录 5秒之后跳转到主页....</font>");

resp.setHeader("Refresh", "5;URL=" + req.getContextPath() + "index.html");

}

}

}

@Override

public void destroy() {

}

}

# 4、实验结果及分析

如果用户未登录而访问非登录页面，则过滤器会进行拦截。

如果用户正常的访问登录则会进入相应的页面

# 附录



