



# 《JavaEE 实验手册》

Java 教研室

版本 1.0

文档提供：Java 教研室 孙丽萍

## 修 改 记 录

修改时间	修改人	修改内容
2012. 8. 29	张立飞	文档创建
2016. 3. 7	孙丽萍	修改

## 目录

一、内容概述.....	4
二、实验一 实现权限认证过滤功能.....	4
2.1 实验目的.....	4
2.2 准备.....	4
2.3 实验步骤.....	4
2.4 实验结论.....	12

## 第六章 过滤器

### 一、内容概述

本章的教学内容是介绍 JavaWeb 开发中 Filter 过滤器相关知识，包括编写，配置，部署，运行以及 Filter 相关接口、类知识介绍。

通过本章实验应该达到的目的：掌握如何编写 Filter，以及 Filter 认证特定角色访问不同资源。

### 二、实验一 实现权限认证过滤功能

#### 2.1 实验目的

编写 Filter 过滤器

实现功能:实现权限认证过滤功能

#### 2.2 准备

保证 Eclipse 已经正确配置 Tomcat 服务器，可以进行 JavaWeb 项目编写、编译与调试。（如果以上设置不正确，请参考《Eclipse 部署配置 Tomcat 手册》进行正确设置）

#### 2.3 实验步骤

步骤一：

在 Eclipse 中新建 Dynamic project “02\_authority”，在 “02\_authority/src” 目录下新建 org.edu.filter.AdminFilter.java，此类实现 Filter 接口，实现最高权限的访问控制，具有管理员权限就转到管理员页面，没有就跳转到公共页面，关键代码如下

```
public void doFilter(ServletRequest request, ServletResponse response,
    FilterChain chain) throws IOException, ServletException {
    if("a".equals(((HttpServletRequest)request).getSession().getAttribute("admin"))){
        ((HttpServletRequest)request).getRequestDispatcher("../admin/admin.html").forward(request, response);
    }else{
        ((HttpServletResponse)response).sendRedirect("../public.html");
    }
}
```

图 2-1

步骤二：

在 “02\_authority/WebContent/WEB-INF/” 的 web.xml 文件中配置 Filter

节点, 过滤对于 admin 文件夹的访问权限, 关键代码如下

```
<filter>
  <filter-name>admin</filter-name>
  <filter-class>org.edu.filter.AdminFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>admin</filter-name>
  <url-pattern>/admin/*</url-pattern>
</filter-mapping>
```

图 2-2

### 步骤三:

在“02\_ authority/src”目录下新建 org.edu.filter.UserFilter.java, 此类实现 Filter 接口, 实现用户的访问控制, 具有管理员权限或用户权限就转到用户页面, 没有就跳转到公共页面, 关键代码如下

```
public void doFilter(ServletRequest request, ServletResponse response,
    FilterChain chain) throws IOException, ServletException {
    if("a".equals(((HttpServletRequest)request).getSession().getAttribute("admin"))
        || "u".equals(((HttpServletRequest)request).getSession().getAttribute("user"))){
        ((HttpServletRequest)request).getRequestDispatcher("../user/user.html").forward(request, response);
    }else{
        ((HttpServletResponse)response).sendRedirect("../public.html");
    }
}
```

图 2-3

### 步骤四:

在“02\_ authority/WebContent/WEB-INF/”的 web.xml 文件中配置 Filter 节点, 过滤对于 user 文件夹的访问权限, 关键代码如下

```
<filter>
  <filter-name>user</filter-name>
  <filter-class>org.edu.filter.UserFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>user</filter-name>
  <url-pattern>/user/*</url-pattern>
</filter-mapping>
```

图 2-4

### 步骤五:

在“02\_filter/src”目录下新建 org.edu.servlet.RoleServlet.java 的 servlet 文件, doPost() 方法中处理客户端提交的权限信息, 如果是管理员, 将管理员信息放入 session 并跳转到 admin/admin.html 页面, 如果是用户将用户信息放入 session 并跳转到 user/user.html 页面, 否则清空 session 跳转到公共页面, 关键代码如下

```

protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    String role = request.getParameter("roles");
    if("a".equals(role)){
        request.getSession().setAttribute("admin", role);
        response.sendRedirect("admin/admin.jsp");
    }else if("u".equals(role)){
        request.getSession().setAttribute("user", role);
        response.sendRedirect("user/user.jsp");
    }else{
        request.getSession().removeAttribute("admin");
        request.getSession().removeAttribute("user");
        response.sendRedirect("public.html");
    }
}

```

图 2-5

在 web.xml 中配置 RoleServlet 节点

```

<servlet>
    <servlet-name>role</servlet-name>
    <servlet-class>org.edu.servlet.RoleServlet</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>role</servlet-name>
    <url-pattern>/role</url-pattern>
</servlet-mapping>

```

图 2-6

#### 步骤六:

编写设置权限页面 02\_authority/WebContent/index.html、公共页面 02\_authority/WebContent/public.html、用户页面 02\_authority/WebContent/user/user.html、管理员页面 02\_authority/WebContent/admin/admin.html, 其中公共页面、用户页面、管理员页面显示欢迎用户信息就可以了, 关键代码如下

```

7 <body>
8 <form action="role" method="post">
9 权限: <input type="radio" name="roles" value="e" />任意用户
0 <input type="radio" name="roles" value="u" />用户
1 <input type="radio" name="roles" value="a" />管理员<br/>
2 <input type="submit" value="提交" />
3 </form>
4 </body>

```

图 2-7 index.html

```

<body>
everyone access this page!
</body>

```

图 2-8 public.html

```
<body>  
admin or user access this page!  
</body>
```

图 2-9 user.html

```
<body>  
just only admin access this page!  
</body>
```

图 2-10 admin.html

**步骤七:**

运行项目“02\_authority”，在浏览器中输入  
http://localhost:8080/02\_authority/, 回车，效果图如下：

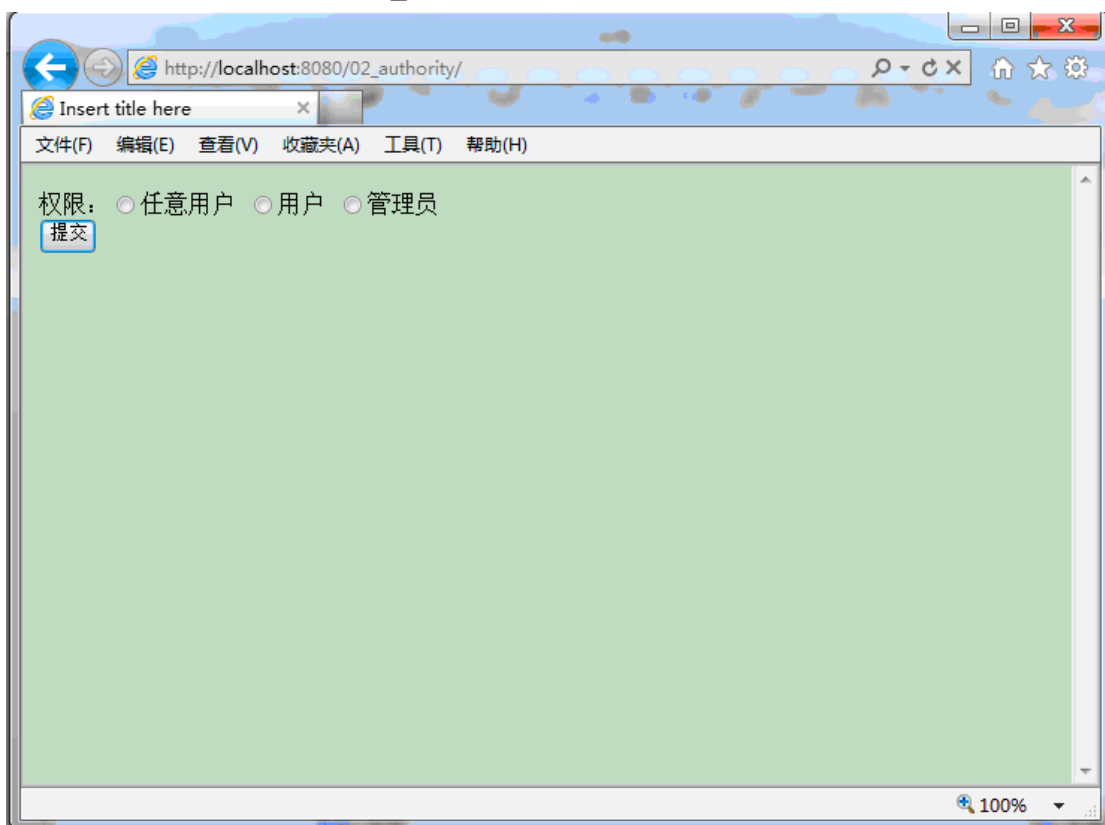


图 2-11

在权限表单上面选择任意用户，点击提交按钮，效果如下

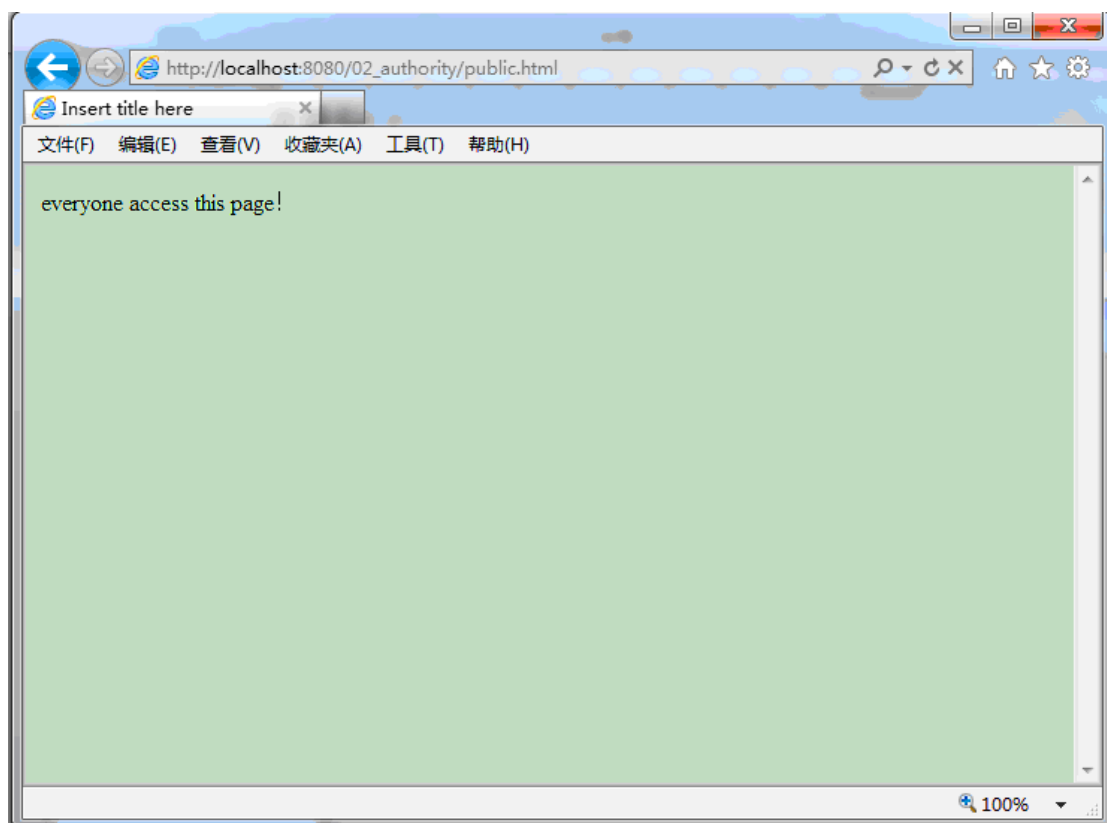


图 2-12

尝试在浏览器中将 url 改为  
[http://localhost:8080/02\\_authority/admin/admin.html](http://localhost:8080/02_authority/admin/admin.html) 或  
[http://localhost:8080/02\\_authority/user/user.html](http://localhost:8080/02_authority/user/user.html), 回车, 效果图如下:



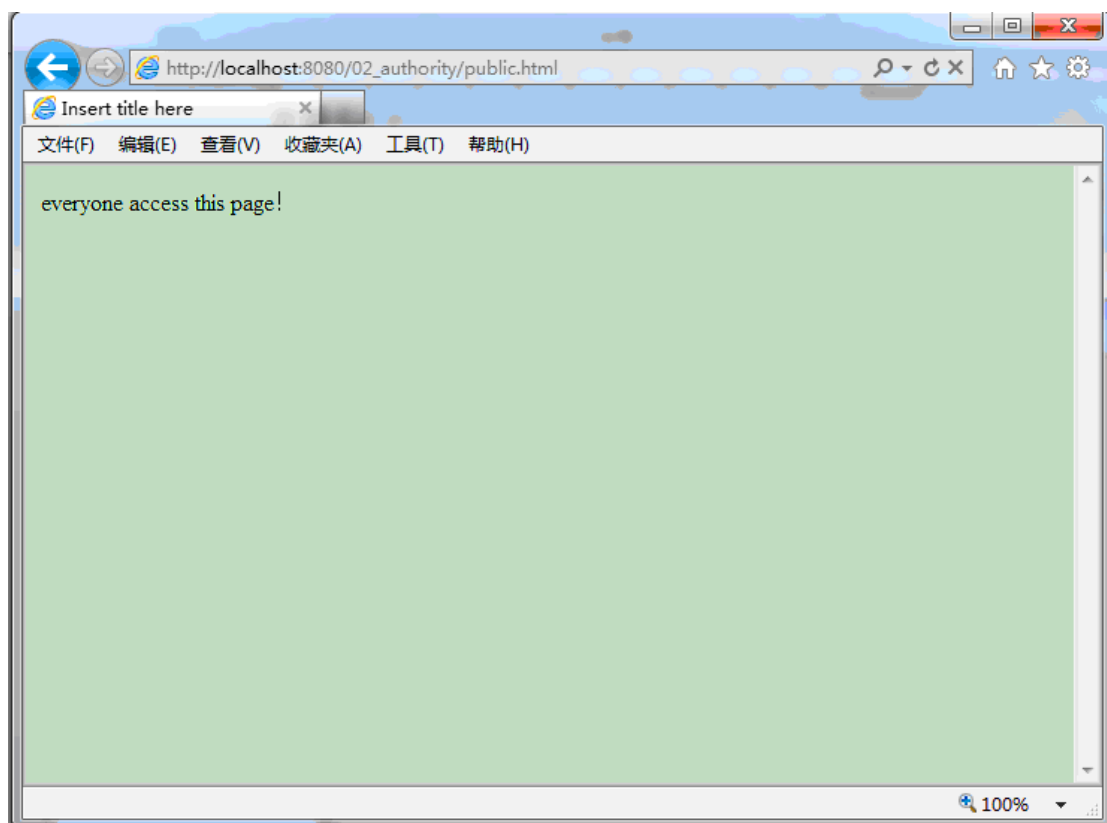


图 2-13

返回首页，在权限表单上面选择用户，点击提交按钮，，效果如下



图 2-14

尝试在浏览器中将 url 改为  
[http://localhost:8080/02\\_authority/admin/admin.html](http://localhost:8080/02_authority/admin/admin.html) 回车，效果图如下：

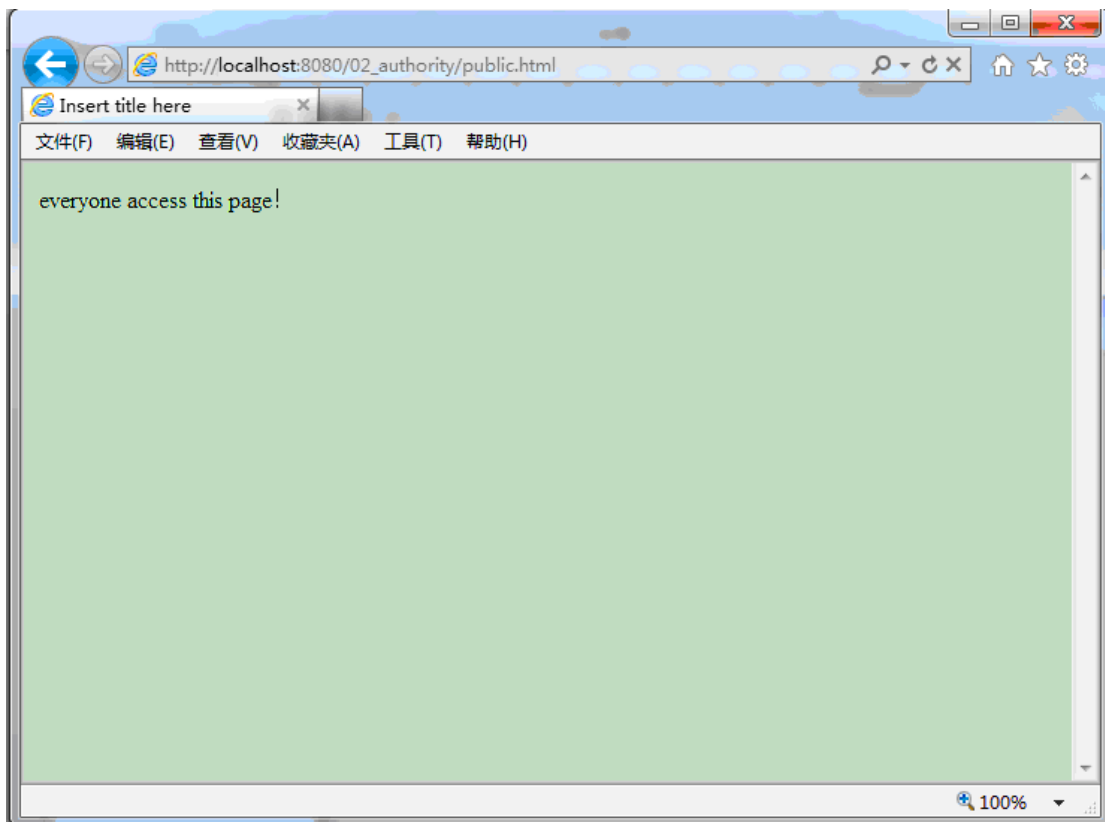


图 2-15

返回首页，在权限表单上面选择管理员，点击提交按钮，，效果如下

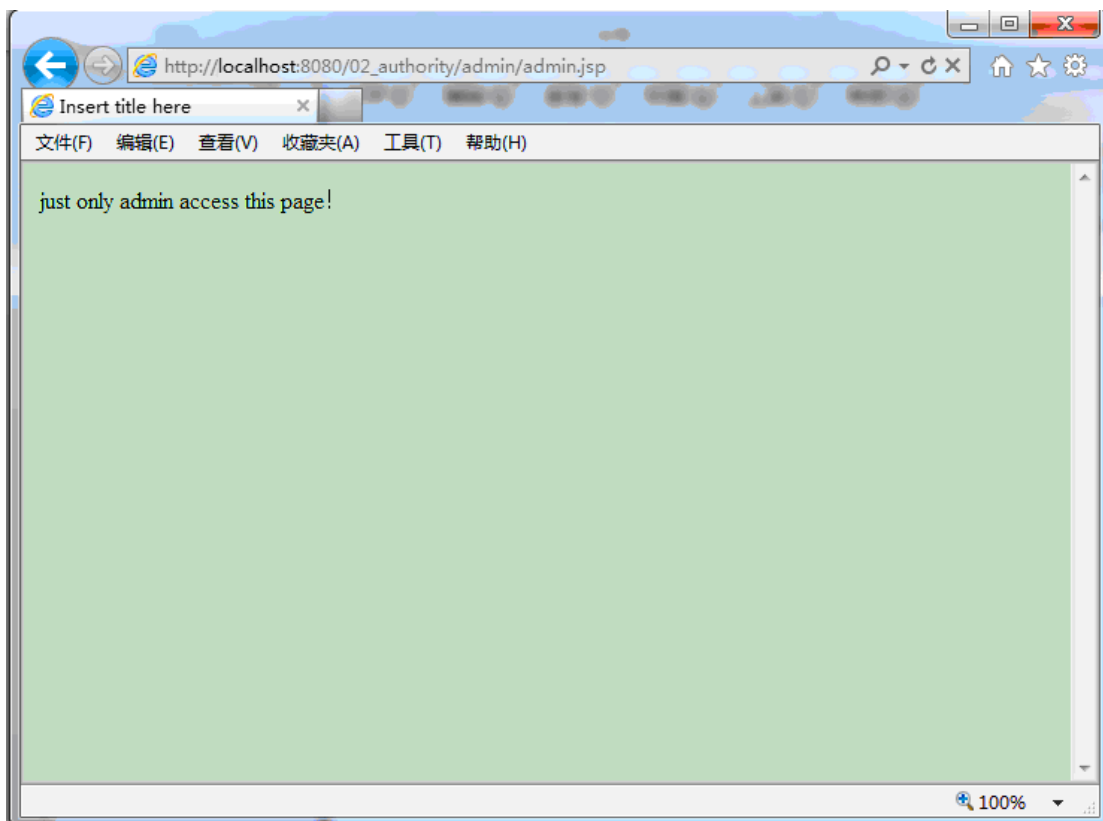


图 2-16

尝试在浏览器中将 url 改为

[http://localhost:8080/02\\_authority/user/user.html](http://localhost:8080/02_authority/user/user.html) 回车, 效果图如下:



图 2-17

## 2.4实验结论

通过以上实验可以观察出，使用过滤器使我们无代码污染的以组件的方式进行权限认证，可以控制特定的资源被特定的角色访问。