

使用一个拦截器实现登录验证

- 1、新建maven项目
- 2、使用原来wen.xml文件中的中央调度器
- 3、修改index.jsp发起请求
- 4、是用原来的MyController类处理请求
- 5、修改结果显示页面show.jsp
- 6、创建一个login.jsp，模拟登录（把用户信息放入session对象中）
创建一个logout.jsp，模拟登出（将session中的信息清除）
- 7、创建拦截器，从session中获取用户的登录数据，验证能否访问该资源
- 8、修改SpringMVC的配置文件

创建项目模块g-springmvc-login-interceptor-07，代码和配置文件使用06项目的，具体修改如下

- 1)、Controller类不变
- 2)、拦截器类只留下一个，内容如下

```
package com.studymyself.handler;

import org.springframework.web.servlet.HandlerInterceptor;
import org.springframework.web.servlet.ModelAndView;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.util.Date;

//拦截器类：拦截用户的请求
public class MyInterceptor implements HandlerInterceptor {

    @Override
    public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler) throws Exception {

        Object attr = request.getSession().getAttribute("name");

        //判断是否是登录状态
        if (attr==null){

            request.getRequestDispatcher("/view/tip.jsp").forward(request,response);
            return false;

        }
        return true;

    }

}
```

5、修改index页面和修改结果显示页面show.jsp

show.jsp

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<%
    String basePath = request.getScheme()+ "://" +
        request.getServerName()+ ":" + request.getServerPort()+
        request.getContextPath()+ "/";
%>
<html>
<head>
    <title>正确结果展示</title>
    <base href="<%=basePath%>">
</head>
<body>
    show.jsp结果展示页面<br>
    myName数据: ${myName}<br>
    myAge数据: ${myAge}<br>
    <a href="view/logout.jsp">点击登出</a>
</body>
</html>
```

index.jsp

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<%
    String basePath = request.getScheme()+ "://" +
        request.getServerName()+ ":" + request.getServerPort()+
        request.getContextPath()+ "/";
%>
<html>
<head>
    <title>欢迎页面</title>
    <base href="<%=basePath%>">
</head>
<body>
<P>提交参数给Controller，接收请求参数</P>
<form action="myInterceptor/some.do" method="post">
    姓名:<input type="text" name="name"><br>
    年龄:<input type="text" name="age"><br>
    <input type="submit" value="提交参数">
</form><br>
<a href="view/login.jsp">去登录</a>
</body>
</html>
```

6、创建一个login.jsp，模拟登录（把用户信息放入session对象中）

创建一个logout.jsp，模拟登出（将session中的信息清除）

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<%
    String basePath = request.getScheme()+ "://" +
```

```

        request.getServerName()+":"+request.getServerPort()+
        request.getContextPath()+"/";
    %>
<html>
<head>
    <title>登录</title>
    <base href="<%=basePath%>">
</head>
<body>
    模拟登录页面：已登录<br>
    <%
        session.setAttribute("name","deng");
    %>
    <a href="view/index.jsp">点击返回</a>
</body>
</html>

```

```

<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<%
    String basePath = request.getScheme()+"://"+
        request.getServerName()+":"+request.getServerPort()+
        request.getContextPath()+"/";
    %>
<html>
<head>
    <title>登出</title>
    <base href="<%=basePath%>">
</head>
<body>
    登出模拟：已登出!!!
    <%
        session.removeAttribute("name");
    %>
    <a href="index.jsp">点击返回</a>
</body>
</html>

```

提示页面tip.jsp

```

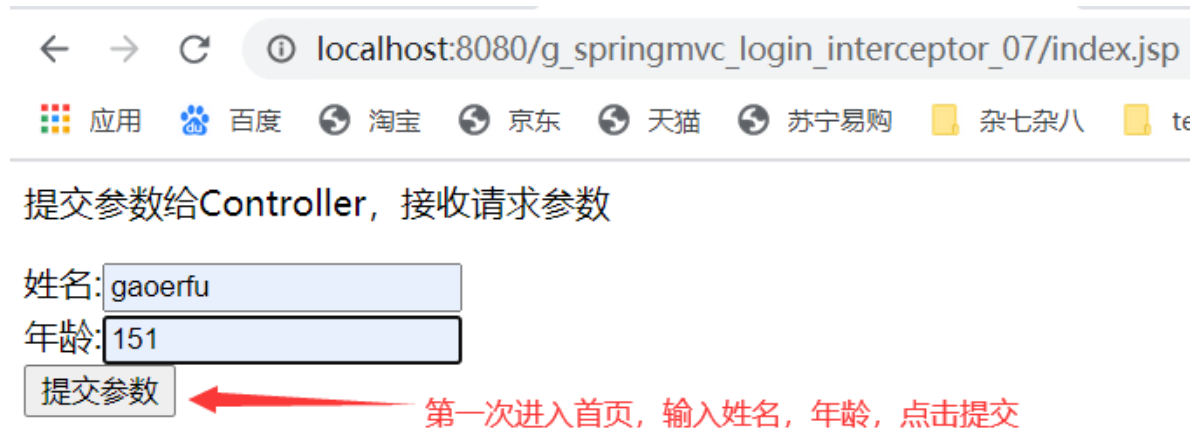
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<%
    String basePath = request.getScheme()+"://"+
        request.getServerName()+":"+request.getServerPort()+
        request.getContextPath()+"/";
    %>
<html>
<head>
    <title>提示</title>
    <base href="<%=basePath%>">
</head>
<body>
    tip.jsp提示展示页面<br>
    该用户无法访问该请求，该请求被拦截!!!<br>
    <a href="view/login.jsp">去登录</a>
</body>

```

```
</html>
```

部署项目测试

1、第一次进入首页，直接输入姓名密码



提交参数给Controller，接收请求参数

姓名: gaoerfu


年龄: 151

提交参数

第一次进入首页，输入姓名，年龄，点击提交

[去登录](#)

结果如下:



localhost:8080/g_springmvc_login_interceptor_07/myInterceptor/some.do

tip.jsp提示展示页面

该用户无法访问该请求，该请求被拦截!!!

[去登录](#)

点击 去登录，如下页面



localhost:8080/g_springmvc_login_interceptor_07/view/login.jsp

模拟登录页面：已登录

[点击返回](#)

然后点击返回，如下页面

← → ↻

localhost:8080/g_springmvc_login_interceptor_07/index.jsp

应用 百度 淘宝 京东 天猫 苏宁易购 杂七杂八 tensorflow

提交参数给Controller，接收请求参数

姓名: 交互稿

年龄: 212

提交参数

登录后在返回该页面，输入姓名年龄，提交

[去登录](#)

重新输入姓名密码，点击提交，结果如下

← → ↻

localhost:8080/g_springmvc_login_interceptor_07/myInterceptor/some.do

应用 百度 淘宝 京东 天猫 苏宁易购 杂七杂八 tensorflow 工具

show.jsp结果展示页面
myName数据：交互稿
myAge数据：212
[点击登出](#)

可以访问这个资源了

然后点击登出，如下结果

← → ↻

localhost:8080/g_springmvc_login_interceptor_07/view/logout.jsp

应用 百度 淘宝 京东 天猫 苏宁易购 杂七杂八 tensorflow

登出模拟：已登出!!! [点击返回](#)

点击返回，到首页，再输入姓名密码，提交，可以发现无法访问doSome方法了(即some.do资源)

← → ↻

localhost:8080/g_springmvc_login_interceptor_07/index.jsp

应用 百度 淘宝 京东 天猫 苏宁易购 杂七杂八 tensorflow

提交参数给Controller，接收请求参数

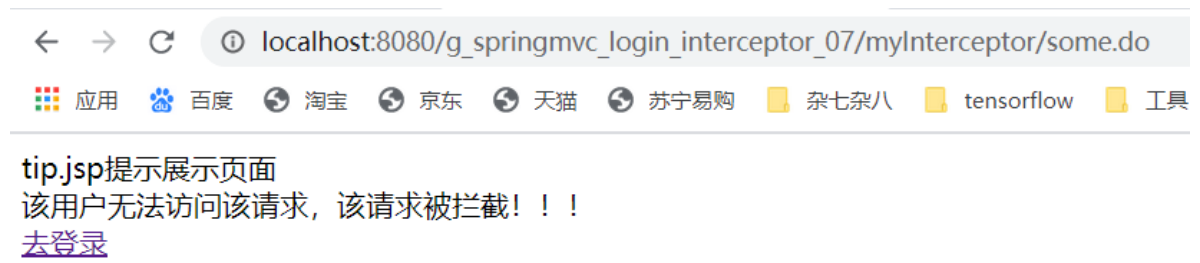
姓名: 交互稿

年龄: 212

提交参数

当点击登出后返回该页面，重新提交，转到提示页

[去登录](#)



完美，也可以点击首页的去登录，再回来提交数据