## 拦截器应用——验证是否登录

### 处理流程

1. 有一个登录页面，需要写一个Controller访问登录页面
2. 登录页面有一提交表单的动作。需要在Controller中处理。
   1. 判断用户名密码是否正确（在控制台打印）
   2. 如果正确,向session中写入用户信息（写入用户名username）
   3. 跳转到商品列表
3. 拦截器。
   1. 拦截用户请求，判断用户是否登录（登录请求不能拦截）
   2. 如果用户已经登录。放行
   3. 如果用户未登录，跳转到登录页面。

### 编写登录jsp

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

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

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

<title>Insert title here</title>

</head>

<body>

<form action=*"*${pageContext.request.contextPath }*/user/login.action"*>

<label>用户名：</label>

<br>

<input type=*"text"* name=*"username"*>

<br>

<label>密码：</label>

<br>

<input type=*"password"* name=*"password"*>

<br>

<input type=*"submit"*>

</form>

</body>

</html>

### 用户登陆Controller

@Controller

@RequestMapping("user")

**public** **class** UserController {

/\*\*

\* 跳转到登录页面

\*

\* **@return**

\*/

@RequestMapping("toLogin")

**public** String toLogin() {

**return** "login";

}

/\*\*

\* 用户登录

\*

\* **@param** username

\* **@param** password

\* **@param** session

\* **@return**

\*/

@RequestMapping("login")

**public** String login(String username, String password, HttpSession session) {

// 校验用户登录

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

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

// 把用户名放到session中

session.setAttribute("username", username);

**return** "redirect:/item/itemList.action";

}

}

### 编写拦截器

public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object arg2) throws Exception {

// TODO Auto-generated method stub

System.out.println("方法前 1");

//判断用户是否登陆 如果没有登陆 重定向到登陆页面 不放行 如果登陆了 就放行了

// URL http://localhost:8080/springmvc-mybatis/login.action

//URI /login.action

String requestURI = request.getRequestURI();

if(!requestURI.contains("/login")){

String username = (String) request.getSession().getAttribute("USER\_SESSION");

if(null == username){

response.sendRedirect(request.getContextPath() + "/login.action");

return false;

}

}

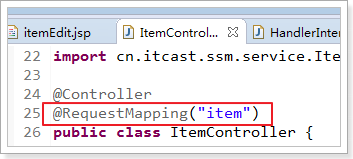
return true;

}

}

### 配置拦截器

只能拦截商品的url，所以需要修改ItemController，让所有的请求都必须以item开头，如下图：



在springmvc.xml配置拦截器

<mvc:interceptor>

<!-- 配置商品被拦截器拦截 -->

<mvc:mapping path=*"/item/\*\*"* />

<!-- 配置具体的拦截器 -->

<bean class=*"cn.itcast.ssm.interceptor.LoginHandlerInterceptor"* />

</mvc:interceptor>