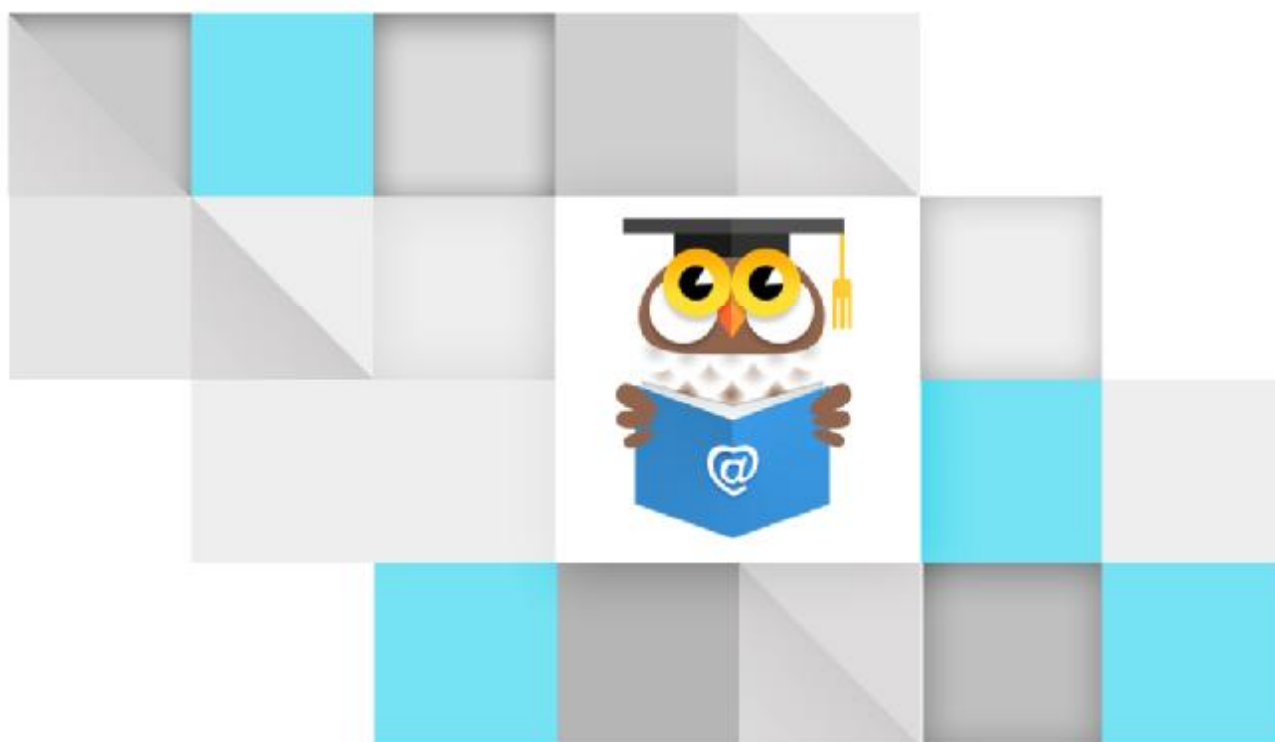


# 去哪玩旅游网（动态 WEB）

## 教学指导手册

### PRJ-WTP-JEE-017 – 登录状态过滤器



**Campus** Solution Group

# 目 录

一、场景说明 .....	1
1、完成效果 .....	1
2、业务描述 .....	1
3、实现思路 .....	2
4、核心组件 .....	3
二、实训技能 .....	4
1、重点演练 .....	4
2、相关技能 .....	4
3、相关知识点 .....	4
4、前置条件 .....	4
5、搭建环境 .....	5
三、场景任务 .....	6
任务 1、创建并配置 Filter 组件 .....	6
任务 2、控制隐私资源的有效性访问 .....	9
任务 3、优化有效性访问的跳转 URL .....	13

# 一、场景说明

## 1、完成效果

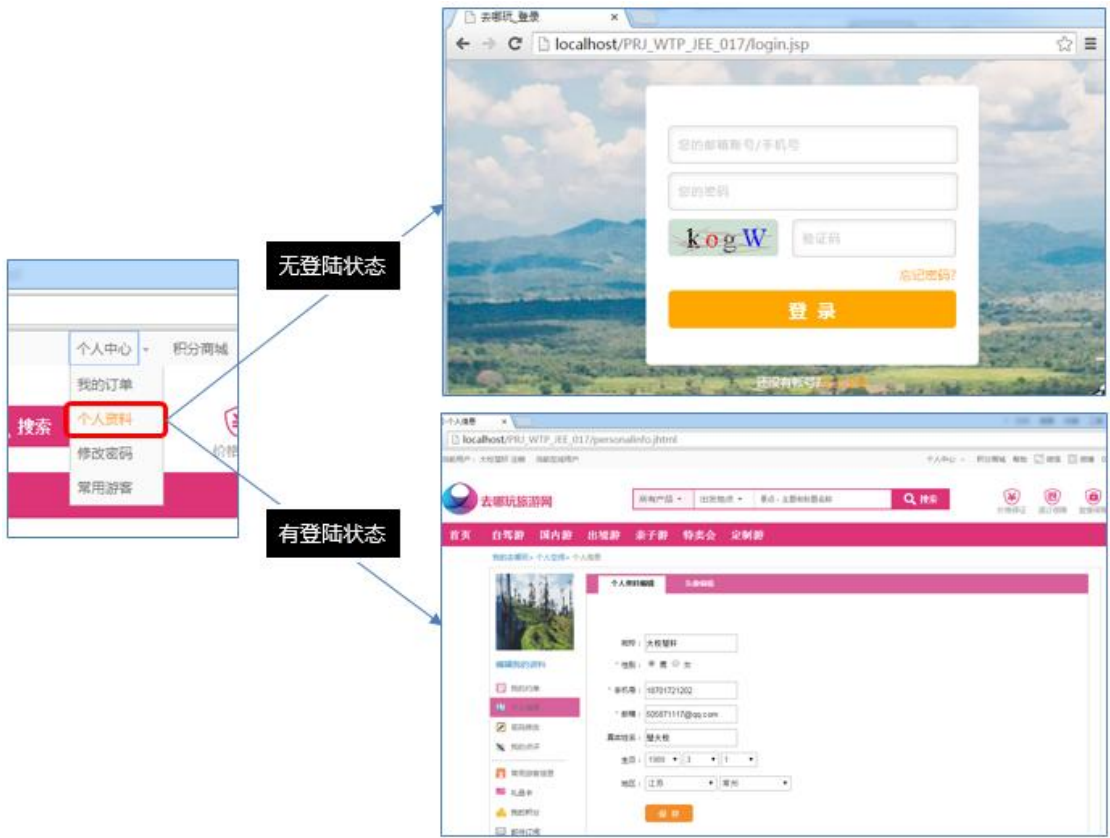


图 1-1-1

## 2、业务描述

本场景主要用于实现，《去哪玩》旅游网页面的**有效性访问**功能。

2-1. 什么是**有效性访问**？

2-1.1. 通过技术手段控制某些页面只有在登录后才能访问，某些页面则无需登录。

2-2. 为何需要设置**有效性访问**（一个常见的电子商务业务功能）？

2-2.1. 站点为方便用户浏览商品、查询商品、了解商品的详情，一般这些页面都允许用

户在不登录的情况下直接访问，例如：《去哪玩》的首页、产品详情页等。

2-2.2. 当用户需要购买某个商品或进入站点的个人空间时，系统就会强制要求用户登录

后才能访问，例如：《去哪玩》的个人信息页。

### 3、实现思路

3-1. 如何实现页面的有效性访问？

3-1.1. 拦截站点的所有URL，依次判断该URL是否需要执行有效性访问验证。

3-1.2. 如果需要，则进一步判断用户是否已经登录。

3-1.3. 如果没有登录，则直接跳转到登录页面，要求用户登录后再访问。

3-1.4. 实现思路技术结构如下图：

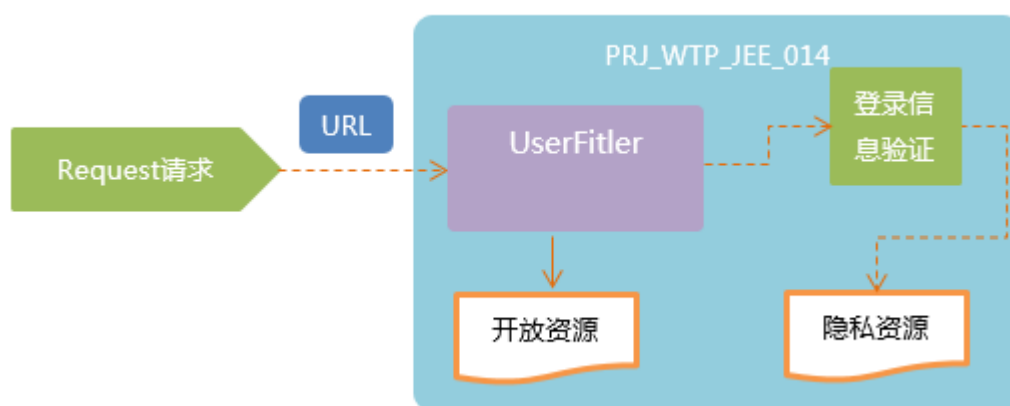


图 1-3-1

3-2. 本场景建议将“有效性访问”业务分为三个任务依次实现：

3-2.1. 任务1. 创建有效性访问过滤器，拦截站点所有URL。

3-2.2. 任务2. 对隐私资源分类的URL，执行有效性访问验证。

1) 隐私资源：只有在登录后才能访问的站点资源（页面）。

2) 开放资源：无需登录，即可访问的站点资源（页面）。

3-2.3. 任务3. 基于任务2，优化跳转连接，确保重新登录后直接访问隐私资源。

- 1) 因访问隐私资源而被要求登录时，登录成功后应跳转到该隐私资源。

## 4、核心组件

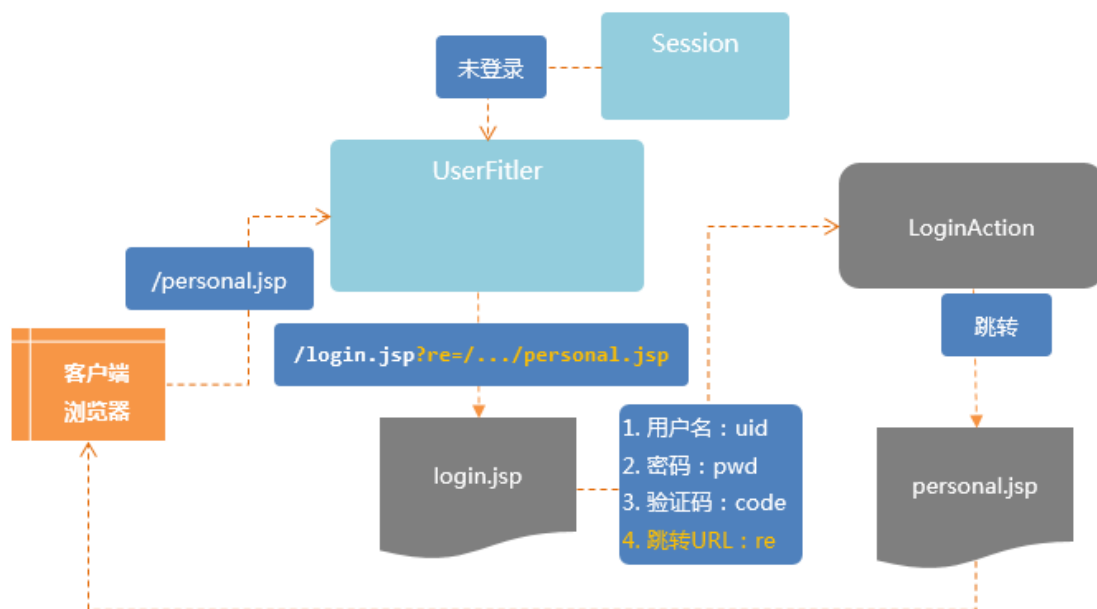


图 1-4-1

### 4-1. login.jsp ( 已提供 )

4-1.1. 《去哪玩》旅游网登录页，场景PRJ-WTP-JEE-007已开发完成。

### 4-2. personal.jsp ( 已提供 )

4-2.1. 《去哪玩》旅游网的个人信息页。

### 4-3. LoginAction ( 已提供 )

4-3.1. 《去哪玩》旅游网登录页请求处理Servlet。

### 4-4. UserFilter ( 需实现 )

4-4.1. 有效性访问过滤器，负责拦截所有客户端请求。

4-4.2. 当前场景，该过滤器负责判断URL是否允许在用户不登录的情况下直接访问。

## 二、实训技能

### 1、重点演练

- 1-1. 创建、配置、使用Filter组件。
- 1-2. 通过FilterConfig对象，读取web.xml文件中的配置信息。
- 1-3. 通过Filter实现页面跳转。

### 2、相关技能

- I Filter 技术应用

### 3、相关知识点

- I Filter 接口
- I Filter 配置
- I Filter 与 Servlet
- I 权限验证 Filter

### 4、前置条件

- 4-1. 前置场景：PRJ-WTP-JEE-016 乱码过滤器
- 4-2. 已学技能：
  - 4-2.1. Java开发工具（Eclipse）。
  - 4-2.2. Web容器（Tomcat）。
  - 4-2.3. Servlet应用技术。

4-2.4. JSP应用技术。

4-2.5. EL表达式应用技术。

4-2.6. Filter应用技术。

## 5、搭建环境



## 三、场景任务

### 任务 1、创建并配置 Filter 组件

#### 1. 任务说明：

##### 1-1. 完成效果：

UserFilter被创建

信息: Server startup in 506 ms

UserFilter获得请求: /PRJ\_WTP\_JEE\_017/  
UserFilter获得请求: /PRJ\_WTP\_JEE\_017/image\_cache/6.jpg  
UserFilter获得请求: /PRJ\_WTP\_JEE\_017/image\_cache/17.jpg  
UserFilter获得请求: /PRJ\_WTP\_JEE\_017/image\_cache/26.jpg  
UserFilter获得请求: /PRJ\_WTP\_JEE\_017/image\_cache/36.jpg  
UserFilter获得请求: /PRJ\_WTP\_JEE\_017/image\_cache/29.jpg  
UserFilter获得请求: /PRJ\_WTP\_JEE\_017/image\_cache/49.jpg  
UserFilter获得请求: /PRJ\_WTP\_JEE\_017/image\_cache/20.jpg  
UserFilter获得请求: /PRJ\_WTP\_JEE\_017/image\_cache/37.jpg  
UserFilter获得请求: /PRJ\_WTP\_JEE\_017/image\_cache/41.jpg  
UserFilter获得请求: /PRJ\_WTP\_JEE\_017/image\_cache/1.jpg  
UserFilter获得请求: /PRJ\_WTP\_JEE\_017/image\_cache/14.jpg  
UserFilter获得请求: /PRJ\_WTP\_JEE\_017/image\_cache/34.jpg  
UserFilter获得请求: /PRJ\_WTP\_JEE\_017/image\_cache/23.jpg  
UserFilter获得请求: /PRJ\_WTP\_JEE\_017/image\_cache/39.jpg  
UserFilter获得请求: /PRJ\_WTP\_JEE\_017/image\_cache/45.jpg  
UserFilter获得请求: /PRJ\_WTP\_JEE\_017/image\_cache/47.jpg  
UserFilter获得请求: /PRJ\_WTP\_JEE\_017/image\_cache/10.jpg  
UserFilter获得请求: /PRJ\_WTP\_JEE\_017/image\_cache/43.jpg

图 3-1-1

##### 1-2. 任务目标：

1-2.1. 创建有效性访问过滤器，拦截站点所有URL。

##### 1-3. 任务要求：

1-3.1. 创建负责处理有效性访问功能的过滤器（Filter）组件：

1）限制1. Filter取名：UserFilter。



2) 限制2. `UserFilter`创建于`campsg.qunawan.filter`包中。

3) 限制3. 允许通过任意类型URL ( `/*` ) 访问该Filter。

1-3.2. 当`UserFilter`获取请求后，在控制台输出：

1) Filter创建时，输出 “`UserFilter被创建`” 。

2) Filter接收到请求时，输出：“`UserFilter获得请求：xxxxxx`” 。

## 2. 实现思路：

2-1. 创建负责处理有效性访问功能的过滤器：`UserFilter`：

2-1.1. 在`campsg.qunawan.filter`包中创建Class类，命名：`UserFilter`。

2-1.2. 使`UserFilter`实现`javax.servlet.Filter`接口。

2-1.3. 重写`javax.servlet.Filter`接口中的`init`、`doFilter`、`destory`方法。

2-2. 配置`UserFilter`：

2-2.1. 在`web.xml`中配置`UserFilter`。

2-2.2. 将`UserFilter`的url-pattern设置为：`/*`

2-3. 使用`UserFilter`：

2-3.1. 在`UserFilter`的`init`方法中，打印：“`UserFilter被创建`” 。

2-3.2. 在`UserFilter`的`doFilter`方法中，获取请求URL连接。

2-3.3. 随后打印：“`UserFilter获得请求：`” 拼接获取的URL连接。

## 3. 推荐步骤：

3-1. 按实现思路，创建名为`UserFilter`的过滤器Filter。

3-2. 按实现思路，配置名为`UserFilter`的过滤器Filter。

3-3. 使用`UserFilter`：

3-3.1. 在`init`方法中，打印：“`UserFilter被创建`” 。

3-3.2.在doFilter方法中，获取请求URL并输出：

+ 提示：

1) 在Filter类的doFilter方法获取请求URL，需依次执行以下步骤：

- 1-1. 将ServletRequest对象强制类型转换成：HttpServletRequest。
- 1-2. 通过HttpServletRequest类的getRequestURI方法，获取请求URL
- 1-3. 代码如下所示：

```
HttpServletRequest req = (HttpServletRequest) request;  
System.out.println("UserFilter获得请求: " + req.getRequestURI());
```

3-3.3. 最后，执行过滤链（FilterChain）的doFilter方法，运行下一Web组件。

4. 验证与测试：

- 4-1. 运行项目工程。
- 4-2. 使用Chrome浏览器访问该项目（必须）。
- 4-3. 在浏览器地址栏输入 [http://localhost/PRJ\\_WTP\\_JEE\\_017/](http://localhost/PRJ_WTP_JEE_017/) 访问首页。
- 4-4. 观察控制台，应见【任务完成效果截图】。

## 任务 2、控制隐私资源的有效性访问

### 1. 任务说明：

#### 1-1. 完成效果：

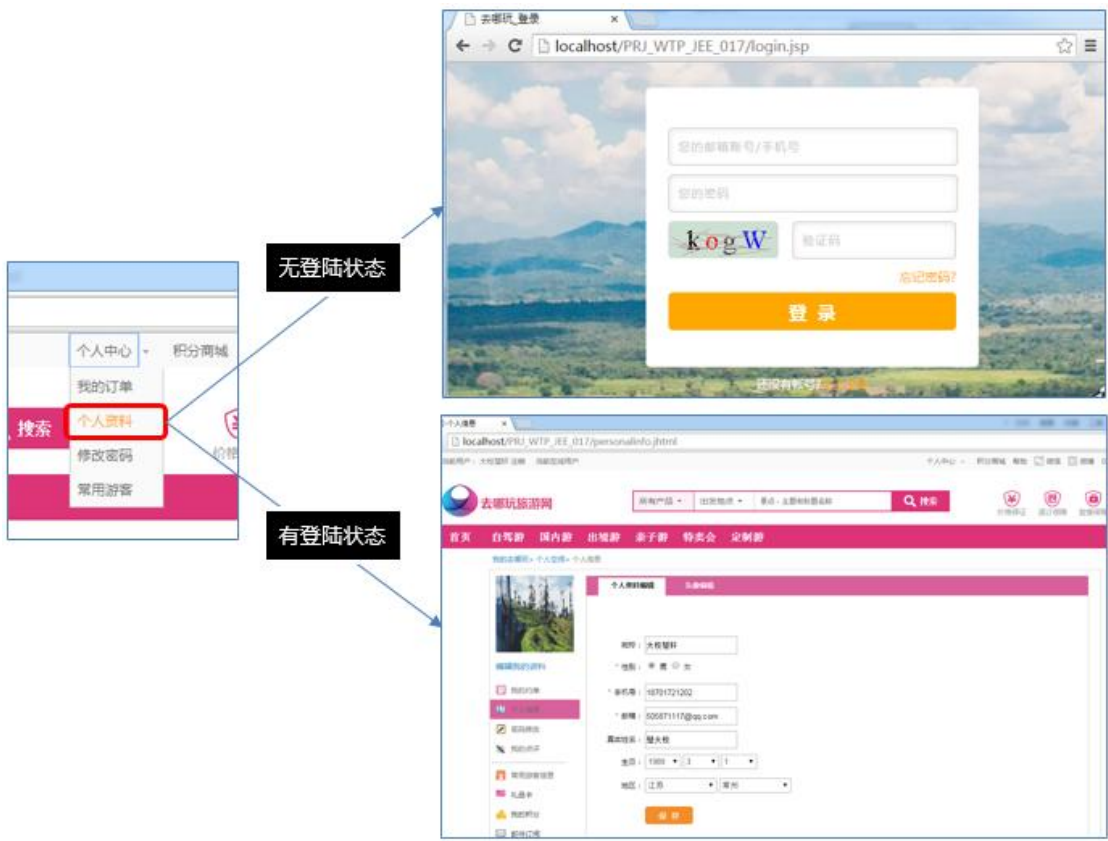


图 3-2-1

#### 1-2. 任务目标：

##### 1-2.1. 对隐私资源分类的URL，执行有效性访问验证。

- 1) 隐私资源：只有在登录后才能访问的站点资源（页面）。
- 2) 开放资源：无需登录，即可访问的站点资源（页面）。

#### 1-3. 任务要求：

##### 1-3.1. 本场景将以下内容定义为隐私资源：

- 1) 《去哪玩》旅游网个人主页：</personal.jsp>

2) 个人页请求处理Servlet：**PersonalAction**的访问URL：**/personalinfo.jhtml**

1-3.2. 其他资源，如：首页、登录页、图片资源等均为**开放资源**。

1-3.3. 为保证系统的可维护性，将**隐私资源**保存于文件中，方便维护：

1) 限制1. **隐私资源**需保存于web.xml文件的Filter配置中。

2) 限制2. 多个**隐私资源**之间用 “;” 间隔，方便读取。

1-3.4. **UserFilter**处理**有效性访问**业务：

1) UserFilter获取（读取）隐私资源。

2) UserFilter根据拦截的URL，判断其是否隶属于隐私资源。

3) 若是隐私资源，则判断当前用户是否登录。

4) 没有登录则直接跳转到登录页面。

## 2. 实现思路：

2-1. 将**隐私资源**保存于文件中，方便维护：

2-1.1. 在web.xml文件中，创建**UserFilter**的**init-param**节点。

1) param-name 节点建议设置为：**checked**

2) param-value 节点设置为：多个**隐私资源**之间用 “;” 间隔。

2-2. **UserFilter**处理**有效性访问**业务：

2-2.1. 在**UserFilter**的**init**方法中，通过**FilterConfig**读取**隐私资源**。

2-2.2. 在**UserFilter**的**doFilter**方法中，以 “;” 分割多个隐私资源，形成数组。

2-2.3. 根据获取的请求URL，判断其是否为**隐私资源**。

2-2.4. 如果当前URL是隐私资源，那么做如下处理：

1) 从 Session 作用域中获取当前登录的用户对象：User，**key = user**。

2) 如果 User 为 Null，则表示用户没有登录，客户端重定向到：**login.jsp**

3) 否则认为用户登录成功, 执行过滤链 ( FilterChain ) 的 doFilter 方法, 运行下一 Web 组件。

2-2.5. 如果当前URL是开放资源, 那么做如下处理:

1) 执行过滤链 ( FilterChain ) 的 doFilter 方法, 运行下一 Web 组件。

2-3. 任务开发流程见下图:

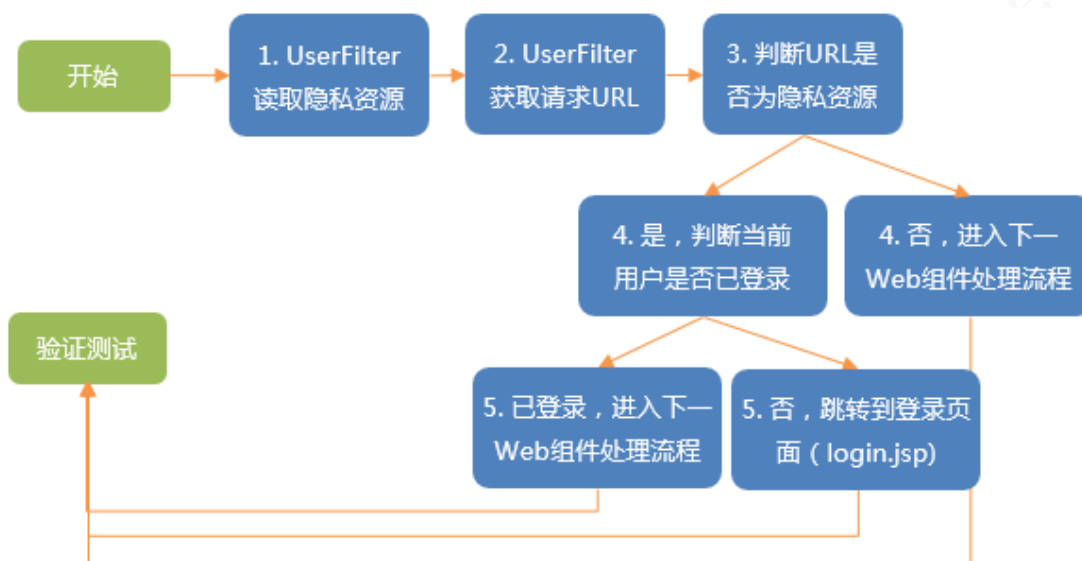


图 3-2-2

### 3. 推荐步骤:

3-1. 将隐私资源保存于文件中, 方便维护:

#### +提示:

1) UserFilter的初始化参数配置, 建议定义成如下形式:

```
<init-param>
  <param-name>checked</param-name>
  <param-value>/personal info.jhtml; personal.jsp</param-value>
</init-param>
```

3-2. 在UserFilter的init方法中, 通过FilterConfig读取隐私资源

3-3. 以 ";" 分割多个隐私资源, 形成数组, 建议使用String类的split函数。

3-4. 按实现思路, 依次实现UserFilter处理有效性访问业务

#### 4. 验证与测试：

4-1. 运行项目工程。

4-2. 使用Chrome浏览器访问该项目（必须）。

4-3. 在浏览器地址栏输入 [http://localhost/PRJ\\_WTP\\_JEE\\_017/](http://localhost/PRJ_WTP_JEE_017/) 访问首页。

4-4. 测试项1，用户不登录直接访问隐私资源：

4-4.1. 点击首页顶部工具条，右侧【个人中心】下拉菜单中的：个人资料。

4-4.2. 此时应见登录页。

4-5. 测试项2，用户不登录，通过URL访问隐私资源：

4-5.1. 输入 [http://localhost/PRJ\\_WTP\\_JEE\\_017/personal.jsp](http://localhost/PRJ_WTP_JEE_017/personal.jsp) 访问个人信息页。

4-5.2. 此时应见登录页。

4-6. 测试项3，用户登录后访问隐私资源：

4-6.1. 在登录页输入正确的验证码。

4-6.2. 输入正确的用户名、密码【用户名：18701721202，密码：123456】。

4-6.3. 进入首页，单击【个人中心】下拉菜单中的：个人资料。

4-6.4. 此时应见个人信息页。

### 任务 3、优化有效性访问的跳转 URL

#### 1. 任务说明：

##### 1-1. 完成效果：



图 3-3-1

##### 1-2. 任务目标：

1-2.1. 基于任务2，优化跳转连接，确保重新登录后直接访问隐私资源：

##### 1-3. 任务要求：

1-3.1. 因访问隐私资源而被要求登录时，登录成功后应跳转到该隐私资源，例如：

- 1) 未登录直接访问personal.jsp，因UserFilter的处理，直接跳转到login.jsp。
- 2) 登录成功后，应该直接访问personal.jsp，而不应该显示首页。

#### 2. 实现思路：

2-1. 优化有效性访问的跳转URL的技术实现思路如下图：

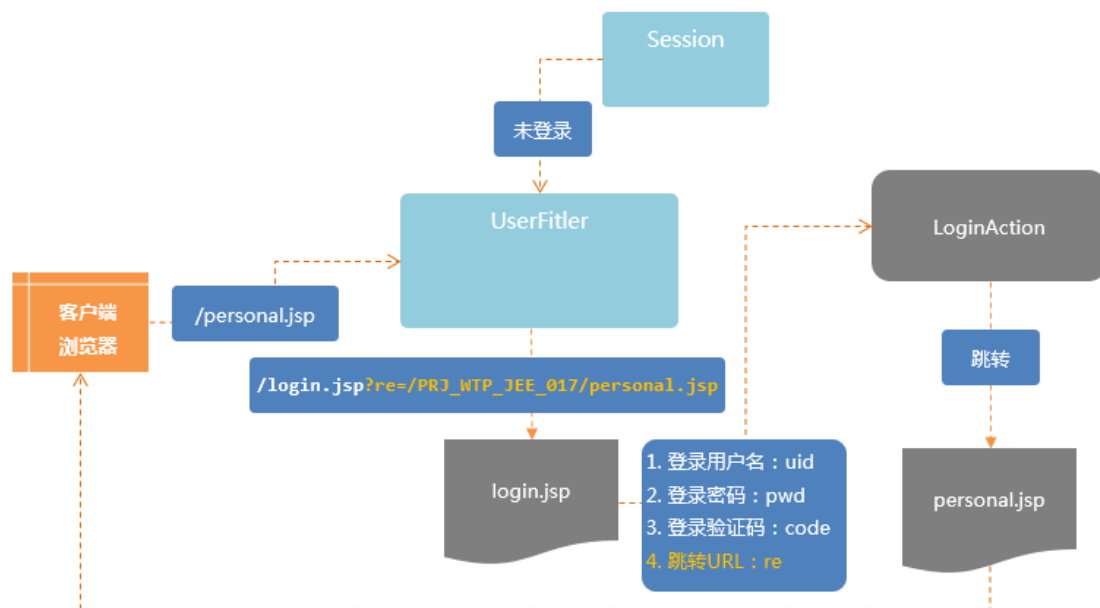


图 3-3-2

2-2. 步骤1. 在UserFilter内有效性验证失败后，跳转到login.jsp时，需追加隐私URL：

例如：/login.jsp?re=/PRJ\_WTP\_JEE\_017/personal.jsp

2-3. 步骤2. login.jsp将隐私资源URL保存在隐藏域（hidden）。

2-4. 步骤3. 当用户登录时，隐藏域（hidden）内的隐私URL会被提交给LoginAction。

2-5. 步骤4. LoginAction在验证登录信息有效后，验证请求中是否包含隐私URL。

2-6. 步骤5. 如果存在隐私URL，那么LoginAction直接跳转隐私URL。

2-7. 步骤6. 如果没有隐私URL，那么LoginAction跳转到index.jhtml（首页）。

### 3. 推荐步骤：

3-1. 修改UserFilter，确保UserFilter在跳转到login.jsp时，需追加隐私URL。

3-2. login.jsp将隐私资源URL保存在隐藏域（hidden）：

3-2.1. 找到form表单标签，增加name=re的hidden标签。

3-2.2. 通过EL表达式将UserFilter传递过来的隐私URL，保存在隐藏域中。

+ 提示：



1) 设置保存隐私资源URL的隐藏域 ( hidden ) :

```
<form method="post" action="/PRJ_WTP_JEE_017/login.jhtml">  
  <input type="hidden" name="re" value="${param.re}" />  
</form>
```

3-3. LoginAction请求中是否包含隐私URL:

3-3.1. 如果存在隐私URL , 那么LoginAction直接跳转隐私URL。

3-3.2. 如果没有隐私URL , 那么LoginAction跳转到index.jhtml ( 首页 )。

+ 提示 :

1) LoginAction请求中是否包含隐私URL :

```
String re = request.getParameter("re");  
if(re != null) {  
    response.sendRedirect(re);  
    return;  
}  
response.sendRedirect("index.jhtml");
```

4. 验证与测试 :

4-1. 运行项目工程。

4-2. 使用Chrome浏览器访问该项目 ( 必须 )。

4-3. 在浏览器地址栏输入 [http://localhost/PRJ\\_WTP\\_JEE\\_017/](http://localhost/PRJ_WTP_JEE_017/) 访问首页。

4-4. 点击首页顶部工具条, 右侧【个人中心】下拉菜单中的: 个人资料。

4-5. 因UserFilter的作用, 显示登录页。

4-6. 在登录页输入正确的验证码。

4-7. 输入正确的用户名、密码【用户名: 18701721202, 密码: 123456】。

4-8. 登录后, 应见个人信息页。

作者: Fox.Sun