

# 微信中同时支持公开和未公开页面

## 1 概述

此示例同样适用于 v7

### 业务场景：

某公司在微信公众平台上集成的时候会出现session冲突的问题。情况是这样，在平台上集成了我们smartbi里做的两块功能，微招聘与投诉。

1. 微招聘：使用公共的用户（可以理解为系统内置匿名用户）自动登录到我们smartbi
2. 投诉：需要用户登录后才能打开该页面

两个功能独立运行正常。

### 产生的问题：

当我打开了微招聘，会使用默认用户（chakanjob）自动登录。

在不关闭微信的情况下，再去打开投诉页面，会保留chakanjob用户的session，导致无法打开（因chkan.job用户没有看投诉页面的权限）。

### 解决的思路：

编写一个filter，拦截openresource.jsp，当没有传递用户名密码时，如果判断到用户没有资源权限，直接重定向到登陆界面。从而让微信中同时支持匿名用户登陆资源和需要用户登陆的资源。

## 2 详细步骤

### 2.1 在extension.xml声明过滤器：

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE extension SYSTEM "extension.dtd">
<extension name="Test" alias="Test" desc="Test" priority="100" version="1.0">
    <enable-jsp-processor>1</enable-jsp-processor>
    <filter>
        <filter-name>LoginFilter</filter-name>
        <filter-class>djjt.rfc.LoginFilter</filter-class>
    </filter>
    <filter-mapping>
        <filter-name>LoginFilter</filter-name>
        <url-pattern>/vision/openresource.jsp</url-pattern>
    </filter-mapping>
</extension>
```

### 2.2 编写filter

```

package djjt.rfc;
import java.io.IOException;
import javax.servlet.Filter;
import javax.servlet.FilterChain;
import javax.servlet.FilterConfig;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.http.HttpServletRequest;
import smartbi.catalogtree.CatalogTreeModule;
import smartbi.usermanager.UserManagerModule;
import smartbi.util.StringUtil;
public class LoginFilter implements Filter{
    @Override
    public void destroy() {
    }
    @Override
    public void doFilter(ServletRequest req, ServletResponse resp, FilterChain chain) throws IOException,
        ServletException {
        HttpServletRequest request = (HttpServletRequest) req;
        String resId = request.getParameter("resid");
        String user = request.getParameter("user");
        UserManagerModule userManagerModule = UserManagerModule.getInstance();
        //
        //
        if (!StringUtil.isNullOrEmpty(resId) && StringUtil.isNullOrEmpty(user)
            && userManagerModule.isLogged()) {
            boolean isCatalogElementAccessible = CatalogTreeModule.getInstance()
                .isCatalogElementAccessible(resId, "REF");
            if (!isCatalogElementAccessible) {
                userManagerModule.logout();
            }
        }
        chain.doFilter(req, resp);
    }
    @Override
    public void init(FilterConfig config) throws ServletException {
    }
}

```