



链滴

# 单点登陆总结

作者: [wk1024](#)

原文链接: <https://ld246.com/article/1519784119087>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

## 一.非单点登陆模式(SSM环境)

总体思路：拦截Http请求，校验用户名密码是否正确；正确放行，否则跳转登录页。

实现方式：

1.使用servlet Filter实现。

2.使用spring mvc 拦截器实现。

下面介绍基于interceptor实现方式：

自定义类实现HandlerInterceptor接口。此接口有三个方法：

preHandle(): 预处理，Http请求被前端控制器DispatcherServlet拦截之后，到达具体controller之执行的动作。

postHandle():Http请求执行完成之后，渲染页面之前执行的动作。

afterCompletion():页面渲染完成后执行的动作。

在preHandle()方法里加入登录逻辑的判断：校验通过，放行，并存放用户信息(用户id, 角色code, 限code,部门等)到session里，同一个session无需再走是否登录逻辑。

spring-mvc.xml配置文件中配置自定义Interceptor。

多节点环境部署的话利用memcached做session共享,tomcate的context.xml配置如下：

```
<Manager className="de.javakaffee.web.msm.MemcachedBackupSessionManager"
sticky="false"
lockingMode="auto"
memcachedNodes="n1:172.0.0.1:8888"
requestUriIgnorePattern=".*\.(png|gif|jpg|css|js)$"
sessionBackupAsync="false"
sessionBackupTimeout="100"
transcoderFactoryClass="de.javakaffee.web.msm.serializer.javolution.JavolutionTranscoderFactory"
copyCollectionsForSerialization="false" />
```

## 二.单点登陆

随着业务系统的增多，每个系统单独维护用户信息这种方式显得异常繁琐，不便维护。在多业务系统境下，集中管理和维护用户，权限等信息。

原理：

这里写图片描述

拦截Http请求(Servlet Filter),校验是否登录；未登录则重定向到SSO认证中心，校验用户名和密码，功并生成token作为参数返回业务系统,业务系统并保存此session(供单点注销使用)。部署图如下：

这里写图片描述

涉及两个单独工程：1.sso-client.jar(单点登陆Filter, 单点注销 Filter) 。2.sso-server.war(sso认证we工程，校验用户名密码以及token)

注意点：多节点部署业务系统，保存session时需要集中保存。

## 三.单点注销

删除sso的session及业务系统session。（session.invalidate()）

流程如下图：

这里写图片描<https://b3logfile.com/e/bec7f6f311ef47448f99b23ddc68db82.pn><https://b3logfile>

com/e/e9a94bf65a384d80af5c4bb3d2e6946b.pnhttps://b3logfile.com/e/13ec18134cbb4f119  
9f27a8456668b1.png