[**第十八章 并发登录人数控制——《跟我学Shiro》**](http://jinnianshilongnian.iteye.com/blog/2039760)

**博客分类：**

* [跟我学Shiro](http://jinnianshilongnian.iteye.com/category/305053)

[跟我学Shiro](http://www.iteye.com/blogs/tag/%E8%B7%9F%E6%88%91%E5%AD%A6Shiro)

**目录贴：**[**跟我学Shiro目录贴**](http://jinnianshilongnian.iteye.com/blog/2018398)

在某些项目中可能会遇到如每个账户同时只能有一个人登录或几个人同时登录，如果同时有多人登录：要么不让后者登录；要么踢出前者登录（强制退出）。比如spring security就直接提供了相应的功能；Shiro的话没有提供默认实现，不过可以很容易的在Shiro中加入这个功能。

示例代码基于《第十六章 综合实例》完成，通过Shiro Filter机制扩展KickoutSessionControlFilter完成。

**首先来看看如何配置使用（spring-config-shiro.xml）**

kickoutSessionControlFilter用于控制并发登录人数的

**Java代码  [收藏代码](javascript:void())**

1. <bean id="kickoutSessionControlFilter"
2. **class**="com.github.zhangkaitao.shiro.chapter18.web.shiro.filter.KickoutSessionControlFilter">
3. <property name="cacheManager" ref="cacheManager"/>
4. <property name="sessionManager" ref="sessionManager"/>
6. <property name="kickoutAfter" value="false"/>
7. <property name="maxSession" value="2"/>
8. <property name="kickoutUrl" value="/login?kickout=1"/>
9. </bean>

cacheManager：使用cacheManager获取相应的cache来缓存用户登录的会话；用于保存用户—会话之间的关系的；

sessionManager：用于根据会话ID，获取会话进行踢出操作的；

kickoutAfter：是否踢出后来登录的，默认是false；即后者登录的用户踢出前者登录的用户；

maxSession：同一个用户最大的会话数，默认1；比如2的意思是同一个用户允许最多同时两个人登录；

kickoutUrl：被踢出后重定向到的地址；

shiroFilter配置

**Java代码  [收藏代码](javascript:void())**

1. <bean id="shiroFilter" **class**="org.apache.shiro.spring.web.ShiroFilterFactoryBean">
2. <property name="securityManager" ref="securityManager"/>
3. <property name="loginUrl" value="/login"/>
4. <property name="filters">
5. <util:map>
6. <entry key="authc" value-ref="formAuthenticationFilter"/>
7. <entry key="sysUser" value-ref="sysUserFilter"/>
8. <entry key="kickout" value-ref="kickoutSessionControlFilter"/>
9. </util:map>
10. </property>
11. <property name="filterChainDefinitions">
12. <value>
13. /login = authc
14. /logout = logout
15. /authenticated = authc
16. /\*\* = kickout,user,sysUser
17. </value>
18. </property>
19. </bean>

此处配置除了登录等之外的地址都走kickout拦截器进行并发登录控制。

**测试**

此处因为maxSession=2，所以需要打开3个浏览器（需要不同的浏览器，如IE、Chrome、Firefox），分别访问<http://localhost:8080/chapter18/>进行登录；然后刷新第一次打开的浏览器，将会被强制退出，如显示下图：



KickoutSessionControlFilter核心代码：

**Java代码  [收藏代码](javascript:void())**

1. **protected** **boolean** onAccessDenied(ServletRequest request, ServletResponse response) **throws** Exception {
2. Subject subject = getSubject(request, response);
3. **if**(!subject.isAuthenticated() && !subject.isRemembered()) {
4. //如果没有登录，直接进行之后的流程
5. **return** **true**;
6. }
8. Session session = subject.getSession();
9. String username = (String) subject.getPrincipal();
10. Serializable sessionId = session.getId();
12. //TODO 同步控制
13. Deque<Serializable> deque = cache.get(username);
14. **if**(deque == **null**) {
15. deque = **new** LinkedList<Serializable>();
16. cache.put(username, deque);
17. }
19. //如果队列里没有此sessionId，且用户没有被踢出；放入队列
20. **if**(!deque.contains(sessionId) && session.getAttribute("kickout") == **null**) {
21. deque.push(sessionId);
22. }
24. //如果队列里的sessionId数超出最大会话数，开始踢人
25. **while**(deque.size() > maxSession) {
26. Serializable kickoutSessionId = **null**;
27. **if**(kickoutAfter) { //如果踢出后者
28. kickoutSessionId = deque.removeFirst();
29. } **else** { //否则踢出前者
30. kickoutSessionId = deque.removeLast();
31. }
32. **try** {
33. Session kickoutSession =
34. sessionManager.getSession(**new** DefaultSessionKey(kickoutSessionId));
35. **if**(kickoutSession != **null**) {
36. //设置会话的kickout属性表示踢出了
37. kickoutSession.setAttribute("kickout", **true**);
38. }
39. } **catch** (Exception e) {//ignore exception
40. }
41. }
43. //如果被踢出了，直接退出，重定向到踢出后的地址
44. **if** (session.getAttribute("kickout") != **null**) {
45. //会话被踢出了
46. **try** {
47. subject.logout();
48. } **catch** (Exception e) { //ignore
49. }
50. saveRequest(request);
51. WebUtils.issueRedirect(request, response, kickoutUrl);
52. **return** **false**;
53. }
54. **return** **true**;
55. }

此处使用了Cache缓存用户名—会话id之间的关系；如果量比较大可以考虑如持久化到数据库/其他带持久化的Cache中；另外此处没有并发控制的同步实现，可以考虑根据用户名获取锁来控制，减少锁的粒度。

另外可参考JavaEE项目开发脚手架，其提供了后台踢出用户的功能：

<https://github.com/zhangkaitao/es/blob/master/web/src/main/java/com/sishuok/es/sys/user/web/controller/UserOnlineController.java>

示例源代码：<https://github.com/zhangkaitao/shiro-example>；可加群 231889722 探讨Spring/Shiro技术。