## 1.配置缓存

# 1.在pom.xml文件中加入依赖：

*<!-- shiro ehcache -->*<**dependency**>  
 <**groupId**>org.apache.shiro</**groupId**>  
 <**artifactId**>shiro-ehcache</**artifactId**>  
 <**version**>1.2.2</**version**>  
</**dependency**>  
  
*<!--   
 包含支持UI模版（Velocity，FreeMarker，JasperReports），  
 邮件服务，  
 脚本服务(JRuby)，  
 缓存Cache（EHCache），  
 任务计划Scheduling（uartz）。  
 -->*<**dependency**>  
 <**groupId**>org.springframework</**groupId**>  
 <**artifactId**>spring-context-support</**artifactId**>  
 </**dependency**>

# 2. shiro缓存管理器; 大型配置文件，记得加入安全管理器

*/\*\*  
 \* shiro缓存管理器;  
 \* 需要注入对应的其它的实体类中：  
 \* 1、安全管理器：securityManager  
 \* 可见securityManager是整个shiro的核心；  
 \** ***@return*** *\*/*@Bean  
**public** EhCacheManager ehCacheManager(){  
 System.out.println(**"ShiroConfiguration.getEhCacheManager()"**);  
 EhCacheManager cacheManager = **new** EhCacheManager();  
 cacheManager.setCacheManagerConfigFile(**"classpath:config/ehcache-shiro.xml"**);  
 **return** cacheManager;  
}

# 3.缓存的配置文件xml

*<?***xml version="1.0" encoding="UTF-8"***?>*<**ehcache name="es"**>  
  
 <**diskStore path="java.io.tmpdir"**/>  
   
 *<!--   
 name:缓存名称。   
 maxElementsInMemory:缓存最大数目   
 maxElementsOnDisk：硬盘最大缓存个数。   
 eternal:对象是否永久有效，一但设置了，timeout将不起作用。   
 overflowToDisk:是否保存到磁盘，当系统当机时  
 timeToIdleSeconds:设置对象在失效前的允许闲置时间（单位：秒）。仅当eternal=false对象不是永久有效时使用，可选属性，默认值是0，也就是可闲置时间无穷大。   
 timeToLiveSeconds:设置对象在失效前允许存活时间（单位：秒）。最大时间介于创建时间和失效时间之间。仅当eternal=false对象不是永久有效时使用，默认是0.，也就是对象存活时间无穷大。  
 diskPersistent：是否缓存虚拟机重启期数据 Whether the disk store persists between restarts of the Virtual Machine. The default value is false.   
 diskSpoolBufferSizeMB：这个参数设置DiskStore（磁盘缓存）的缓存区大小。默认是30MB。每个Cache都应该有自己的一个缓冲区。   
 diskExpiryThreadIntervalSeconds：磁盘失效线程运行时间间隔，默认是120秒。   
 memoryStoreEvictionPolicy：当达到maxElementsInMemory限制时，Ehcache将会根据指定的策略去清理内存。默认策略是LRU（最近最少使用）。你可以设置为FIFO（先进先出）或是LFU（较少使用）。   
 clearOnFlush：内存数量最大时是否清除。   
 memoryStoreEvictionPolicy:  
 Ehcache的三种清空策略;  
 FIFO，first in first out，这个是大家最熟的，先进先出。   
 LFU， Less Frequently Used，就是上面例子中使用的策略，直白一点就是讲一直以来最少被使用的。如上面所讲，缓存的元素有一个hit属性，hit值最小的将会被清出缓存。  
 LRU，Least Recently Used，最近最少使用的，缓存的元素有一个时间戳，当缓存容量满了，而又需要腾出地方来缓存新的元素的时候，那么现有缓存元素中时间戳离当前时间最远的元素将被清出缓存。  
 -->*

<**defaultCache  
 maxElementsInMemory="10000"  
 eternal="false"  
 timeToIdleSeconds="120"  
 timeToLiveSeconds="120"  
 overflowToDisk="false"  
 diskPersistent="false"  
 diskExpiryThreadIntervalSeconds="120"** />  
   
   
 *<!-- 登录记录缓存 锁定10分钟 -->* <**cache name="passwordRetryCache"  
 maxEntriesLocalHeap="2000"  
 eternal="false"  
 timeToIdleSeconds="3600"  
 timeToLiveSeconds="0"  
 overflowToDisk="false"  
 statistics="true"**>  
 </**cache**>  
   
</**ehcache**>

# 记住密码

# 1. ShiroConfiguration大型配置文件

*/\*\*  
 \* cookie对象;  
 \** ***@return*** *\*/*@Bean  
**public** SimpleCookie rememberMeCookie(){  
 System.out.println(**"ShiroConfiguration.rememberMeCookie()"**);  
 *//这个参数是cookie的名称，对应前端的checkbox 的name = rememberMe* SimpleCookie simpleCookie = **new** SimpleCookie(**"rememberMe"**);  
 *//<!-- 记住我cookie生效时间30天 ,单位秒;-->* simpleCookie.setMaxAge(259200);  
 **return** simpleCookie;  
}  
  
*/\*\*  
 \* cookie管理对象;  
 \** ***@return*** *\*/*@Bean  
**public** CookieRememberMeManager rememberMeManager(){  
 System.out.println(**"ShiroConfiguration.rememberMeManager()"**);  
 CookieRememberMeManager cookieRememberMeManager = **new** CookieRememberMeManager();  
 cookieRememberMeManager.setCookie(rememberMeCookie());  
 **return** cookieRememberMeManager;  
}

# 2. 将rememberMeManager注入到SecurityManager中

主要是加入了：

//配置记住我或认证通过可以访问的地址

 filterChainDefinitionMap.put("/index", "user");

 filterChainDefinitionMap.put("/", "user");

# 3. 修改登录界面加入rememberMe复选框：

在login.html中加入：

<P><input type="checkbox" name="rememberMe" />记住我</P>

# 4.测试

这时候运行程序，登录之后跳转到/index页面，然后我们关闭浏览器，然后直接访问/index还是可以访问的，说明我们写的记住密码已经生效了，如果访问<http://127.0.0.1:8080/userInfo/userAdd> 的

话还是需要重新登录的。