**基于注解的授权实现**   
相关的注解：   
**@ RequiresAuthentication**   
可以用户类/属性/方法，用于表明当前用户需是经过认证的用户。

**Java代码  [收藏代码](javascript:void())**

1. @RequiresAuthentication
2. **public** **void** updateAccount(Account userAccount) {
3. //this method will only be invoked by a
4. //Subject that is guaranteed authenticated
5. ...
6. }

**@ RequiresGuest**   
表明该用户需为”guest”用户   
  
**@ RequiresPermissions**   
当前用户需拥有制定权限

**Java代码  [收藏代码](javascript:void())**

1. @RequiresPermissions("account:create")
2. **public** **void** createAccount(Account account) {
3. //this method will only be invoked by a Subject
4. //that is permitted to create an account
5. ...
6. }

**@RequiresRoles**   
当前用户需拥有制定角色   
  
**@ RequiresUser**   
当前用户需为已认证用户或已记住用户 

**基于JSP  TAG的授权实现**   
Shiro提供了一套JSP标签库来实现页面级的授权控制。   
在使用Shiro标签库前，首先需要在JSP引入shiro标签：

**Java代码  [收藏代码](javascript:void())**

1. <%@ taglib prefix="shiro" uri="http://shiro.apache.org/tags" %>

下面一一介绍Shiro的标签：   
guest标签   
验证当前用户是否为“访客”，即未认证（包含未记住）的用户

**Xml代码  [收藏代码](javascript:void())**

1. **<shiro:guest>**
2. Hi there!  Please **<a** href="login.jsp"**>**Login**</a>** or **<a** href="signup.jsp"**>**Signup**</a>** today!
3. **</shiro:guest>**

user标签   
认证通过或已记住的用户

**Xml代码  [收藏代码](javascript:void())**

1. **<shiro:user>**
2. Welcome back John!  Not John? Click **<a** href="login.jsp"**>**here**<a>** to login.
3. **</shiro:user>**

authenticated标签   
已认证通过的用户。不包含已记住的用户，这是与user标签的区别所在。

**Xml代码  [收藏代码](javascript:void())**

1. **<shiro:authenticated>**
2. **<a** href="updateAccount.jsp"**>**Update your contact information**</a>**.
3. **</shiro:authenticated>**

notAuthenticated标签   
未认证通过用户，与authenticated标签相对应。与guest标签的区别是，该标签包含已记住用户。

**Xml代码  [收藏代码](javascript:void())**

1. **<shiro:notAuthenticated>**
2. Please **<a** href="login.jsp"**>**login**</a>** in order to update your credit card information.
3. **</shiro:notAuthenticated>**

principal 标签   
输出当前用户信息，通常为登录帐号信息

**Xml代码  [收藏代码](javascript:void())**

1. Hello, **<shiro:principal/>**, how are you today?

hasRole标签   
验证当前用户是否属于该角色

**Xml代码  [收藏代码](javascript:void())**

1. **<shiro:hasRole** name="administrator"**>**
2. **<a** href="admin.jsp"**>**Administer the system**</a>**
3. **</shiro:hasRole>**

lacksRole标签   
与hasRole标签逻辑相反，当用户不属于该角色时验证通过

**Xml代码  [收藏代码](javascript:void())**

1. **<shiro:lacksRole** name="administrator"**>**
2. Sorry, you are not allowed to administer the system.
3. **</shiro:lacksRole>**

hasAnyRole标签   
验证当前用户是否属于以下任意一个角色。

**Xml代码  [收藏代码](javascript:void())**

1. **<shiro:hasAnyRoles** name="developer, project manager, administrator"**>**
2. You are either a developer, project manager, or administrator.
3. **</shiro:lacksRole>**

hasPermission标签   
验证当前用户是否拥有制定权限

**Xml代码  [收藏代码](javascript:void())**

1. **<shiro:hasPermission** name="user:create"**>**
2. **<a** href="createUser.jsp"**>**Create a new User**</a>**
3. **</shiro:hasPermission>**

lacksPermission标签   
与hasPermission标签逻辑相反，当前用户没有制定权限时，验证通过

**Xml代码  [收藏代码](javascript:void())**

1. **<shiro:lacksPermission** name="user:create"**>**
2. **<a** href="createUser.jsp"**>**Create a new User**</a>**
3. **</shiro:lacksPermission>**

**基于页面过滤器的配置**   
Shiro主要是通过URL过滤来进行安全管理，这里的配置便是指定具体授权规则定义。

**Xml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.jsp"**/>**
4. **<property** name="successUrl" value="/home.jsp"**/>**
5. **<property** name="unauthorizedUrl" value="/unauthorized.jsp"**/>** --**>**
6. **<property** name="filterChainDefinitions"**>**
7. **<value>**
8. # some example chain definitions:
9. /admin/\*\* = authc, roles[admin]
10. /docs/\*\* = authc, perms[document:read]
11. /\*\* = authc
12. # more URL-to-FilterChain definitions here
13. **</value>**
14. **</property>**
15. **</bean>**

URL表达式说明   
1、URL目录是基于HttpServletRequest.getContextPath()此目录设置   
2、URL可使用通配符，\*\*代表任意子目录   
3、Shiro验证URL时，URL匹配成功便不再继续匹配查找。所以要注意配置文件中的URL顺序，尤其在使用通配符时。

4、roles[admin]：具有角色admin

5、perms[document:read]：具有权限document:read