Security3说明

最简单配置样例：

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans:beans xmlns=*"http://www.springframework.org/schema/security"*

xmlns:beans=*"http://www.springframework.org/schema/beans"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.1.xsd*

*http://www.springframework.org/schema/security http://www.springframework.org/schema/security/spring-security-3.1.xsd"*>

<!-- 不要过滤图片等静态资源，其中\*\*代表可以跨越目录，\*不可以跨越目录。 -->

<http pattern=*"/css/\*\*"* security=*"none"* />

<http pattern=*"/img/\*\*"* security=*"none"* />

<http pattern=*"/login.jsp\*"* security=*"none"* />

<http pattern=*"/forgotPassword.jsp"* security=*"none"* />

<http auto-config=*'true'* access-denied-page=*"/accessDenied.jsp"*>

<form-login login-page=*"/login.jsp"*

authentication-failure-url=*"/login.jsp?error=true"*

default-target-url=*"/index.jsp"* always-use-default-target=*'true'*/>

<logout logout-success-url=*"/login.jsp"* />

<intercept-url pattern=*"/hr/\*\*"* access=*"ROLE\_ADMIN"* />

<intercept-url pattern=*"/\*\*"* access=*"ROLE\_USER"* />

</http>

<authentication-manager>

<authentication-provider>

<user-service>

<user name="user" password="123456" authorities="ROLE\_USER" />

<user name="admin" password="123456" authorities="ROLE\_ADMIN" />

</user-service>

</authentication-provider>

</authentication-manager>

</beans:beans>

设置无需保护的资源：

<!-- 不要过滤图片等静态资源，其中\*\*代表可以跨越目录，\*不可以跨越目录。 -->

<http pattern=*"/css/\*\*"* security=*"none"* />

<http pattern=*"/ext/\*\*"* security=*"none"* />

<http pattern=*"/img/\*\*"* security=*"none"* />

<http pattern=*"/jsComm/\*\*"* security=*"none"* />

<http pattern=*"/login.jsp"* security=*"none"* />

<http pattern=*"/forgotPassword.jsp"* security=*"none"* />

。。。。。

1、静态角色控制资源：

<http auto-config='true'>

<intercept-url pattern="/\*\*" access="ROLE\_USER" />

<intercept-url pattern="/system/\*\*" access="ROLE\_ADMIN" />

。。。

</http>

<authentication-manager>

<authentication-provider>

<user-service>

<user name="user" password="user" authorities="ROLE\_USER" />

<user name="admin" password="admin" authorities="ROLE\_ADMIN" />

</user-service>

</authentication-provider>

</authentication-manager>

控制的顺序：从上到下匹配，上面例子中的/system控制无效，因为首页匹配/\*\*，因此角色ROLE\_ADMIN也不能进入/system，只有ROLE\_USER才能进入。另外由于第一行为匹配所有资源所以只有拥有ROLE\_USER角色的用户可以进入系统所有模块，而其它角色则无法进入系统，/system的配置没有任何意义，前面一行就已经匹配完了，根本不会运行到这一行。

2、系统默认数据库配置

Security通过缺省数据库配置也相当简单，只需要创建两张表，一张是users,另一张是authorities， users表中一定要有username(varchar50)\password(varchar50)\enabled(boolean)三个字段，authorities表中一定要有username(varchar50)\authority(varchar50)两个字段，表中可以增加自己想要的其它字段。Authorites表中对应用户的角色，如张三-ROLE\_USER，李四-ROLE\_ADMIN。建好表后，无需写其它任何代码，只需将Authentication-manager配置中原有的user-service改为jdbc-user-service：

<authentication-manager>

<authentication-provider>

<jdbc-user-service data-source-ref=*"dataSource"*/>

</authentication-provider>

</authentication-manager>

配置好后即可如1中的静态角色一样进行控制。