

# SpringMVC整合Shiro

官方文档: <http://shiro.apache.org/spring.html>

- 1.导入依赖包

```
<dependency>
    <groupId>org.apache.shiro</groupId>
    <artifactId>shiro-core</artifactId>
    <version>1.4.1</version>
</dependency>

<dependency>
    <groupId>org.apache.shiro</groupId>
    <artifactId>shiro-web</artifactId>
    <version>1.4.1</version>
</dependency>

<dependency>
    <groupId>org.apache.shiro</groupId>
    <artifactId>shiro-spring</artifactId>
    <version>1.4.1</version>
</dependency>
```

- 2.在web.xml中添加shrio对应的过滤器配置

- 拦截所有请求

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
        http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
    version="3.1">
    <context-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>classpath:spring-*.xml</param-value>
    </context-param>
    <listener>
        <listener-
class>org.springframework.web.context.ContextLoaderListener</listener-class>
        </listener>

    <servlet>
        <servlet-name>springMVC</servlet-name>
        <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-
class>
        <init-param>
            <param-name>contextConfigLocation</param-name>
            <param-value>classpath:spring-context*.xml</param-value>
        </init-param>
```

```

</servlet>
<servlet-mapping>
    <servlet-name>springMVC</servlet-name>
    <url-pattern>*.do</url-pattern>
</servlet-mapping>

<filter>
    <filter-name>shiroFilter</filter-name>
    <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-
class>
    <init-param>
        <param-name>targetFilterLifecycle</param-name>
        <param-value>true</param-value>
    </init-param>
</filter>
<filter-mapping>
    <filter-name>shiroFilter</filter-name>
    <url-pattern>*</url-pattern>
</filter-mapping>
</web-app>

```

### • 3.在spring配置文件中，配置shrio框架所需的内容

- 创建spring-context-shiro.xml
- 完成shiro配置

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="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.xsd">

    <!--shiro过滤器-->
    <bean id="shiroFilter"
class="org.apache.shiro.spring.web.ShiroFilterFactoryBean">
        <property name="securityManager" ref="securityManager"></property>
        <!-- 配置登录页面地址，非必须，默认寻找web项目根路径下的/Login.jsp-->
        <property name="loginUrl" value="/login.jsp"></property>
        <!-- 配置登录成功之后跳转的页面路径，此配置一般不使用，一般在LoginController中处理
逻辑
        <property name="successUrl" value="/index.jsp"></property>
        -->
        <property name="unauthorizedUrl" value="/"></property>
        <property name="filterChainDefinitions">
            <value>/**=anon</value>
        </property>
    </bean>

    <bean id="lifecycleBeanPostProcessor"
class="org.apache.shiro.spring.LifecycleBeanPostProcessor"></bean>

    <!--证书匹配器-->
    <bean id="credentialsMatcher"
class="org.apache.shiro.authc.credential.Md5CredentialsMatcher"></bean>

```

```

<!-- 使用自带的Realm-->
<bean id="jdbcRealm" class="org.apache.shiro.realm.jdbc.JdbcRealm">
    <property name="credentialsMatcher" ref="credentialsMatcher"></property>
    <property name="permissionsLookupEnabled" value="true"></property>
    <property name="dataSource" ref="dataSource"></property>
</bean>

<!-- 缓存管理-->
<bean id="cacheManager"
class="org.apache.shiro.cache.MemoryConstrainedCacheManager"></bean>

<!-- shiro安全管理器-->
<bean id="securityManager"
class="org.apache.shiro.web.mgt.DefaultWebSecurityManager">
    <property name="realm" ref="jdbcRealm"></property>
    <property name="cacheManager" ref="cacheManager"></property>
</bean>

</beans>

```

- **4.配置权限验证使用注解**

```

<!-- 配置权限验证使用注解-->
<aop:config proxy-target-class="true"></aop:config>
<bean
class="org.apache.shiro.spring.security.interceptor.AuthorizationAttributeSourceAdvisor">
    <property name="securityManager" ref="securityManager"></property>
</bean>

```