# 佟刚Shiro知识讲解：

## 一、Shiro概述：

### 1、Shiro简介：

1）Apache Shiro是Java的一个安全（权限）框架

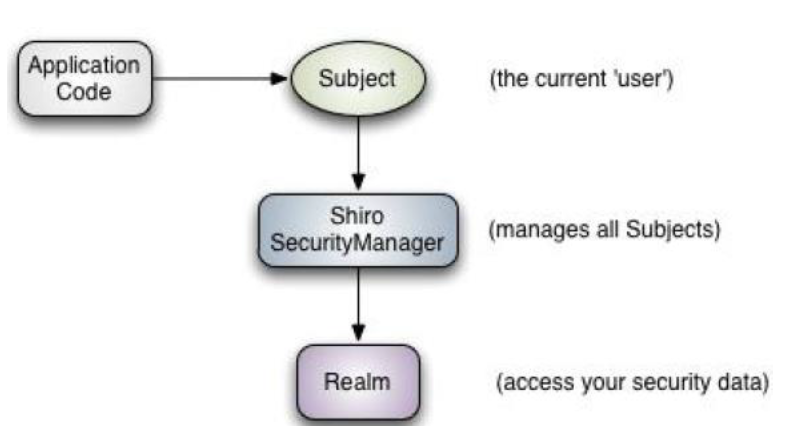
2）Shiro可以非常容易的开发出足够好用的应用，其不仅可以用在JavaSE环境，也可以用在JavaEES环境。

3）Shiro可以完成：认证、授权、加密、会话管理、与Web集成、缓存等

4）其他的安全框架还有Spring Security等（Shiro更简单好用）

### 2、Shiro架构：

1）从外部来看Shiro，即从应用程序角度来观察如何使用Shiro完成工作。



1. Subject：

【1】应用代码直接交互的对象是Subject，也就是说Shiro的对外API核心就是Subject。

【2】Subject代表了当前“用户”，这个用户不一定是一个具体的人，与当前应用交互的任何东西都是Subject，如网络爬虫、机器人等；

【3】与Subject的所有交互都会委托给SecurityManager；

【4】Subject其实是一个门面，SecurityManager才是实际的执行者。

3）Security：

安全管理器，即所有与安全相关的操作都会与SecurityManager交互；且其管理者所有Subject；可以看出它是Shiro的核心，他负责与Shiro的其他组件进行交互，它相当于SpringMVC中DispatcherServlet的角色。

1. Realm：

Shiro从Realm获取安全数据（如：用户、安全、权限），就是说SecurityManager要验证用户身份，那么它需要从Realm获取相应的用户进行比较以确定用户身份是否合法；也需要从Realm得到用户相应的角色/权限进行验证用户是否能进行操作；可以把Realm看成DataSource。

## 二、Shiro集成Spring：【shiroDemo2】

### 1、步骤：

1）加入Spring和Shiro的Jar包

2）配置Spring及SpringMVC

3）配置Shiro：参照Shiro源码\samples\spring下的web.xml和Spring配置文件

【1】在web.xml中配置：

|  |
| --- |
| <!-- Shiro Filter is defined in the spring application context: -->  <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> |

【2】在applicationContext.xml中配置：具体看项目applicationContext.xml

### 2、简单流程：

1）所有的请求都会经过 web.xml 中配置的shiroFilter

|  |
| --- |
| <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> |

2）shiroFilter工作原理：

|  |
| --- |
|  |

在applicationContext.xml 文件中的配置：

|  |
| --- |
|  |

## 三、知识点：

### 1、URL匹配模式：

1）URL模式使用Ant风格模式

2）Ant路径通配符支持 ?、\*、\*\*，注意：通配符匹配不包括目录分割符”/”

【1】 ？：匹配一个字符，如 “/admin？” 将匹配 “/admin1”,但不匹配 “/admin”或 “/admin/”

【2】 \*：匹配零个或多个字符串，如“/admin\*”将匹配 “/admin”、“/admin123”，但不匹配“/admin/1”

【3】 \*\*：匹配路径中的零个或多个路径，如“/admin/\*\*”将匹配 “/admin/a”或 “/admin/a/b”

3）URL匹配顺序：URL权限采取第一次匹配优先的方式，即从头开始使用第一个匹配的URL模式对应的拦截器链。（注意：不是准确原则，而是声明顺序原则）

如：

/bb/\*\*=filter1

/bb/aa=filter2

/\*\*=filter3

如果请求的URL是 “/bb/aa”,因为按照声明顺序进行匹配，那么将使用filter1进行拦截

### 2、Shiro认证思路：

1）获取当前的Subject：调用SecurityUtils.getSubject()方法

2）测试当前的用户是否已经被认证，即是否已经登录：调用；Subject的 isAuthenticated()方法

3）若没有被认证，则把用户名和密码封装为 UserPasswordToken 对象

【1】创建一个表单页面

【2】把请求提交到SpringMVC的Handler

【3】获取用户名和密码

1. 执行登录：调用Subject的login（AuthenticationToken）方法
2. 自定义Realm的方法，从数据库获取对应的记录，返回给Shiro

【1】实际上需要继承 org.apache.shiro.realm.AuthenticatingRealm类

【2】实现 doGetAuthenticationInfo(AuthenticationToken)方法

6）由Shiro完成密码的比对：

【1】通过AuthenticatingRealm的credentialsMatcher属性来进行比对

【参考shiroDemo1-2/login.jsp、list.jsp、ShiroHandler.java、ShiroRealm.java】