Shiro入门Demo

一、shiro简介

Apache Shiro是一个强大易用的Java安全框架，提供了认证、授权、加密和会话管理等功能：

认证 - 用户身份识别，常被称为用户“登录”；

授权 - 访问控制；

密码加密 - 保护或隐藏数据防止被偷窥；

会话管理 - 每用户相关的时间敏感的状态。

对于任何一个应用程序，Shiro都可以提供全面的安全管理服务。

相对于其他安全框架（比如spring-security），Shiro要简单的多。

二、Shiro的架构介绍

首先，来了解一下Shiro的三个核心组件：Subject, SecurityManager 和 Realms. Subject：即“当前操作用户”。但是，在Shiro中，Subject这一概念并不仅仅指人，也可以是第三方进程、后台帐户（Daemon Account）或其他类似事物。它仅仅意味着“当前跟软件交互的东西”。但考虑到大多数目的和用途，你可以把它认为是Shiro的“用户”概念。 Subject代表了当前用户的安全操作，SecurityManager则管理所有用户的安全操作。 SecurityManager：它是Shiro框架的核心，典型的Facade模式，Shiro通过SecurityManager来管理内部组件实例，并通过它来提供安全管理的各种服务。 Realm： Realm充当了Shiro与应用安全数据间的“桥梁”或者“连接器”。也就是说，当对用户执行认证（登录）和授权（访问控制）验证时，Shiro会从应用配置的Realm中查找用户及其权限信息。 从这个意义上讲，Realm实质上是一个安全相关的DAO：它封装了数据源的连接细节，并在需要时将相关数据提供给Shiro。当配置Shiro时，你必须至少指定一个Realm，用于认证和（或）授权。配置多个Realm是可以的，但是至少需要一个。 Shiro内置了可以连接大量安全数据源（又名目录）的Realm，如LDAP、关系数据库（JDBC）、类似INI的文本配置资源以及属性文件等。如果缺省的Realm不能满足需求，你还可以插入代表自定义数据源的自己的Realm实现。

除前文所讲Subject、SecurityManager 、Realm三个核心组件外，Shiro主要组件还包括： Authenticator ：认证就是核实用户身份的过程。这个过程的常见例子是大家都熟悉的“用户/密码”组合。多数用户在登录软件系统时，通常提供自己的用户名（当事人）和支持他们的密码（证书）。如果存储在系统中的密码（或密码表示）与用户提供的匹配，他们就被认为通过认证。 Authorizer ：授权实质上就是访问控制 - 控制用户能够访问应用中的哪些内容，比如资源、Web页面等等。 SessionManager ：在安全框架领域，Apache Shiro提供了一些独特的东西：可在任何应用或架构层一致地使用Session API。即，Shiro为任何应用提供了一个会话编程范式 - 从小型后台独立应用到大型集群Web应用。这意味着，那些希望使用会话的应用开发者，不必被迫使用Servlet或EJB容器了。或者，如果正在使用这些容器，开发者现在也可以选择使用在任何层统一一致的会话API，取代Servlet或EJB机制。 CacheManager :对Shiro的其他组件提供缓存支持。

三、项目代码

四、运行结果

五、项目托管

[shiro\_spring源码](https://github.com/zhenghaishu/shiro\_demo)