**Java shiro面试题**

<http://www.bjpowernode.com/tutorial_baseinterviewquestions/220.html>

**1、简单介绍一下Shiro框架？**

Apache Shiro是Java的一个安全框架。使用Shiro可以非常容易的开发出足够好的应用。其不仅可以用在JavaSE环境，也可以用在JavaEE环境。Shiro可以帮助我们完成功能：认证、授权、加密、会话管理、与Web集成、缓存等。

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：它是管理所有用户的安全操作，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。

**2、Shiro主要的四个组件？**

● SecurityManager

典型的Facade，Shiro通过它对外提供安全管理的各种服务。

● Authenticator

对“Who are you？”进行核实。通常涉及用户名和密码。这个组件负责收集principals 和credentials，并将它们提交给应用系统。如果提交的credentials跟应用系统中提供的 credentials吻合，就能够继续访问，否则需要重新提交principals和credentials，或者直 接终止访问。

● Authorizer

身份份验证通过后，由这个组件对登录人员进行访问控制的筛查，比如“who can do what”，或者“who can do which actions”。Shiro采用“基于Realm”的方法，即用户 （又称Subject）、用户组、角色和permission的聚合体。

● Session Manager

这个组件保证了异构客户端的访问，配置简单。它是基于POJO/J2SE的，不跟任何的 客户端或者协议绑定。

**3、Shiro运行原理？**

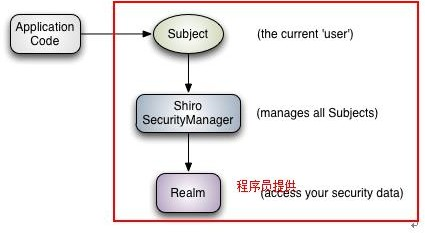
● Application Code：应用程序代码，就是我们自己的编码，如果在程序中需要进行权限控制，需要调用Subject的API。

●Subject：主体代表了当前用户。所有的Subject都绑定到SecurityManager，与Subject的所有交互都会委托给SecurityManager，可以将Subject当成一个门面，而真正执行者是SecurityManager。

● SecurityManage：安全管理器，所有与安全有关的操作都会与SecurityManager交互，并且它管理所有的Subject。

● Realm：域shiro是从Realm来获取安全数据（用户，角色，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，即安全数据源。



**4、Shiro的四种权限控制方式？**

● url级别权限控制

● 方法注解权限控制

● 代码级别权限控制

**5、什么是粗颗粒和细颗粒权限？**

对资源类型的管理称为粗颗粒度权限控制，即只控制到菜单、按钮、方法。粗粒度的例子比如：用户具有用户管理的权限，具有导出订单明细的权限。

对资源实例的控制称为细颗粒度权限管理，即控制到数据级别的权限，比如：用户只允许修改本部门的员工信息，用户只允许导出自己创建的订单明细。

● 总结：

粗颗粒权限：针对url链接的控制。

细颗粒权限：针对数据级别的控制。

比如：卫生局可以查询所有用户，卫生室只可以查询本单位的用户。

**6、粗颗粒和细颗粒如何授权？**

对于粗颗粒度的授权可以很容易做系统架构级别的功能，即系统功能操作使用统一的粗颗粒度的权限管理。对于细颗粒度的授权不建议做成系统架构级别的功能，因为对数据级别的控制是系统的业务需求，随着业务需求的变更业务功能变化的可能性很大，建议对数据级别的权限控制在业务层个性化开发，比如：用户只允许修改自己创建的商品信息可以在service接口添加校验实现，service接口需要传入当前操作人的标识，与商品信息创建人标识对比，不一致则不允许修改商品信息。

粗颗粒权限：可以使用过虑器统一拦截url。

细颗粒权限：在service中控制，在程序级别来控制，个性化编程。