

**让 Apache Shiro 保护你的应用**

Shiro是一个强大易用的 Java 安全框架，提供了认证、授权、加密和会话管理功能，可为任何应用提供安全保障 - 从命令行应用、移动应用到大型网络及企业应用。

Shiro 为解决下列问题提供了保护应用的 API：

* 认证 - 用户身份识别，常被称为用户“登录”；
* 授权 - 访问控制，权限控制；
* 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。

**核心概念：Subject，SecurityManager 和 Realms**

**Subject**

一词是一个安全术语，其基本意思是“当前的操作用户”。称之为“用户”并不准确，因为“用户”一词通常跟人相关。在安全领域，术语“Subject”可以是人，也可以是第三方进程、后台帐户（Daemon Account）或其他类似事物。一旦获得 Subject，你就可以立即获得你希望用 Shiro 为当前用户做的 90% 的事情，如登录、登出、访问会话、执行授权检查等

**SecurityManager**

在幕后为Subject工作的是SecurityManager 。SecurityManager是 Shiro 框架的核心，它在内部管理组织用于实现不同功能的组件。一旦SecurityManager及其内部组件配置好后，应用开发人员几乎把他们的所有时间都花在 SubjectAPI 调用上。

SecurityManager 实例是一个Singleton（不必是一个**静态**Singleton）跟 Shiro 里的几乎所有组件一样，SecurityManager 的缺省实现是POJO，而且可用 POJO 兼容的任何配置机制进行配置 - 普通的 Java 代码、Spring XML、YAML、.properties 和.ini 文件等。基本来讲，能够实例化类和调用 JavaBean 兼容方法的任何配置形式都可使用。

**Realms**

Realm充当了Shiro与应用安全数据间的“桥梁”或者“连接器”。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。Shiro拿到数据后，可以据此执行验证身份，权限检查等功能。由于Realm封装了数据源的连接细节，安全相关的数据可以来自于文件系统、数据库或 内存等不同的数据源。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。

3个组件的关系如图

