Sabar权限模块设计文档

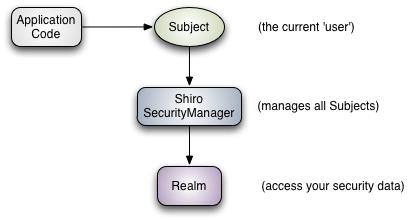
# 背景与原理

在权限模块中，shiro在java开发的系统中有着举足轻重的地位，而在go语言开发中，缺少这样的库，本类库是为了本场景而存在的。

在本初始版本中，将模仿shiro来实现，但是功能比较弱，并提供web程序的接口。

以下设计来源于shiro，并保持与shiro基本相同，基本概念与设计也是来源于http://www.ituring.com.cn/article/305。

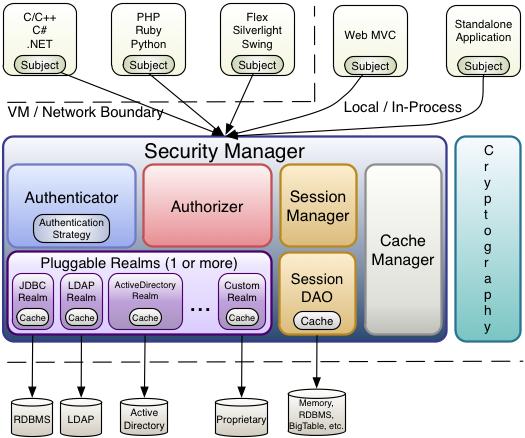
三个主要概念：Subject、SecurityManager、Realms。



Subject：Subject是当前用户在安全领域的一个缩影。User通常暗含人的意思，而Subject可以是人，可以是第三方服务，守护账户，corn作业。或者说是任何与系统交互的都可以叫做Subject。 所有的Subject实例都必须绑定到一个SecurityManager，这样当与Subject交互时，其实已经转化成与这个Subject相关联的SecurityManager进行交互。

SecurityManager：SecurityManager作为Shiro框架的核心，作为“保护伞”对象的形式存在，它协调其内部安全组件形成一个对象图。当在应用中完成SecurityManager和其内部对象的配置后，SecurityManager就退居二线，开发人员大部分时间都是在使用Subject的API。 随后会深入讲解SecurityManager，再次强调：当与Subject交互时，实际上是隐藏其背后的SecurityManager负责Subject繁重的安全操作。这点在上面的图中也有所体现。

Realms：Realms作为Shiro和应用中安全数据源之间的桥梁或者连接器。当需要获取用户账户进行认证(登陆)或授权(访问控制)时，Shiro就会在应用的配置中查找负责这项工作的Realm(一个或多个)来完成获取安全数据。 从这点上讲，Realm实质上就是一个与安全相关的Dao：它封装了数据源链接的详情，并根据Shiro的需要为其提供数据。配置Shiro时，要为认证和授权至少提供一个Reaml。可配置多个Realm，但是至少需要一个。 Shiro内置了可以连接大量安全数据源的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。如果内置的Realm不能满足需求，你还可以使用代表自定义数据源的自己的Realm实现。 和其他内部组件一样，SecurityManager管理Realm如何获取与Subject相关的安全和身份信息。



# 功能

2.1 权限

2.2 登陆

2.3 退出

2.4 加密策略

2.5

# 类图

# 使用示例

# 开发计划