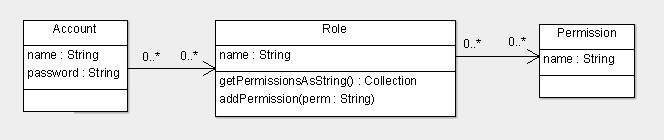
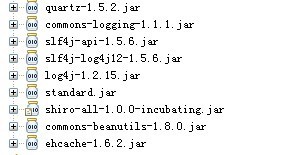
应用Shiro到Web Application

**一、简介**  
如果你正想学习权限方面的知识，或者正打算把Shiro作为权限组件集成到自己的web application中，或许正在为Shiro如何实现CAPTCHA（验证码）功能而伤透脑筋。那么，本文正是为你准备的。本文简单介绍权限方面的基础知识并以实际例子，带你进入Shiro的世界。

**二、权限基础**  
a)    认证(你是谁？)  
判断你(被认证者)是谁的过程。通常被认证者提供用户名和密码。  
  
常见的认证包含如下几种  
匿名认证：允许访问资源，不做任何类型的安全检查。  
表单认证：访问资源之前，需要提交包含用户名和密码的表单。这是web application最常用的认证方式。这个过程一般会接合Session，只在第一次（新会话）访问资源时提交认证表单。  
基本HTTP认证：基于RFC 2617的一种认证方式。  
用户认证：Filter that allows access to resources if the accessor is a known user, which is defined as having a known principal. This means that any user who is authenticated or remembered via a 'remember me' feature will be allowed access from this filter.  
  
b)    授权(你可以做什么？)  
判断被认证者(你)是否能做什么操作的过程。  
端口授权：必须通过指定的某个端口才能访问资源。  
Permission授权：Filter that allows access if the current user has the permissions specified by the mapped value, or denies access if the user does not have all of the permissions specified.  
Role授权：Filter that allows access if the current user has the roles specified by the mapped value, or denies access if the user does not have all of the roles specified.  
  
perms     org.apache.shiro.web.filter.authz.PermissionsAuthorizationFilter  
port     org.apache.shiro.web.filter.authz.PortFilter  
roles     org.apache.shiro.web.filter.authz.RolesAuthorizationFilter  
ssl     org.apache.shiro.web.filter.authz.SslFilter  
  
c)    加密  
使用技术手段（如：MD5、SHA等）把待加密的数据变为密文（如：信息摘要等）过程。  
  
d)    RBAC  
基于角色的访问控制（Role-Based Access Control）。  
e)    Realm  
data access object for an application’s security components (users,roles, permissions)  
  
f)    Permission  
最小粒度的授权，不与用户关联。  
例如：导出报表、查看id号为“PO20090008”的采购单、创建FAQ。  
g)    Role  
Permission的集合。

**三、Shiro特点**  
简单  
功能强大  
能独立运行，不依赖其它框架或容器  
包含了认证、授权、Session管理、加密  
易于扩展

**四、web application 集成Shiro**  
接下来，按照如下步骤开始我们的Shiro之旅：  
a)    数据模型 **[](http://photo.blog.sina.com.cn/showpic.html#blogid=76a8f8780100r2c9&url=http://s2.sinaimg.cn/orignal/76a8f878g76cb0669a9a1)**  
用户账号Account，可以简单的理解为用户。  
一个账号可以拥有多个角色（Role）。  
一个角色包含了多个权限（Permission）。  
   
  
b)    创建工程，新建实体，添加与Shiro相关的Jar包  
如果你正在全用Eclipse:  
File--New--Other--Web--Dynamic Web Project  
  
在 /WEB-INFO/lib/目录下添加如下Jar包 **[](http://photo.blog.sina.com.cn/showpic.html#blogid=76a8f8780100r2c9&url=http://s4.sinaimg.cn/orignal/76a8f878ga3ee4322ebc3)**  
   
相关Jar包，你可以在http://incubator.apache.org/shiro/download.html  
c)    配置web.xml，添加过滤器

      <filter>

           <filter-name>ShiroFilter</filter-name>

           <filter-class>org.apache.shiro.web.servlet.IniShiroFilter</filter-class>

      </filter>

      <filter-mapping>

           <filter-name>ShiroFilter</filter-name>

           <url-pattern> / \*</url-pattern>

</filter-mapping>

d)    INI配置  
[main]  
#SHA256加密  
sha256Matcher = org.apache.shiro.authc.credential.Sha256CredentialsMatcher  
  
#realm  
myRealm = com.wearereading.example.shiro.MyShiroRealm  
myRealm.credentialsMatcher = $sha256Matcher  
  
#缓存  
myRealm.authorizationCachingEnabled = true  
cache=org.apache.shiro.cache.ehcache.EhCacheManager  
myRealm.cacheManager=$cache  
  
[filters]  
shiro.loginUrl = /login.jsp  
#authc=org.apache.shiro.web.filter.authc.FormAuthenticationFilter  
authc.successUrl =/background.jsp  
perms.unauthorizedUrl =/401.jsp  
  
[urls]  
/login.jsp=authc  
/logout.jsp=anon  
/about.jsp=anon  
/background.jsp=authc  
  
/faq/test.jsp=authc  
/faq/list.jsp=authc,perms["faq:list"]  
/faq/view.jsp=authc,perms["faq:view"]  
  
位置：  
配置参数可以写在web.xml文件中，也可以单独文件形式存放在本地类根路径、文件系统以及网络环境中。  
Shiro INI Inline Config 和External Config  
  
public class MyShiroRealm extends AuthorizingRealm{  
  
     
    protected AuthorizationInfo doGetAuthorizationInfo(  
            PrincipalCollection principals) {  
        String username = (String) principals.fromRealm(  
                getName()).iterator().next();  
         
        if( username != null ){  
            AccountManager accountManager = new AccountManagerImpl();  
            Collection<Role> myRoles = accountManager.getRoles( username );  
            if( myRoles != null ){  
                SimpleAuthorizationInfo info = new SimpleAuthorizationInfo();  
                for( Role each:myRoles ){  
                    info.addRole(each.getName());  
                    info.addStringPermissions( each.getPermissionsAsString() );  
                }  
                 
                return info;  
            }  
        }  
         
        return null;  
    }  
  
     
    protected AuthenticationInfo doGetAuthenticationInfo(  
            AuthenticationToken authcToken ) throws AuthenticationException {  
        UsernamePasswordToken token = (UsernamePasswordToken) authcToken;  
        String accountName = token.getUsername();  
                 
        //用户名密码验证  
        if( accountName != null && !"".equals(accountName) ){  
            AccountManager accountManager = new AccountManagerImpl();  
            Account account = accountManager.get( token.getUsername() );  
     
            if( account != null )  
                return new SimpleAuthenticationInfo(  
                        account.getName(),account.getPassword(), getName() );  
        }  
  
        return null;  
    }  
  
}  
f)    登录页面  
        <%  
            Object obj = request.getAttribute(org.apache.shiro.web.filter.authc.  
FormAuthenticationFilter.DEFAULT\_ERROR\_KEY\_ATTRIBUTE\_NAME);  
            boolean flag = false;  
            String msg = "";                     
            if( obj != null ){  
                if( "org.apache.shiro.authc.UnknownAccountException".equals( obj ) )  
                    msg = "未知帐号错误！";  
                else if("org.apache.shiro.authc.IncorrectCredentialsException".equals( obj ))  
                    msg = "密码错误！";                     
                else if( "org.apache.shiro.authc.AuthenticationException".equals( obj ))  
                    msg = "认证失败！";  
                flag = !"".equals(msg);  
            }             
            if( flag )  
                out.print( msg );  
%>  
  
<form action="login.jsp" method="post">  
            <br/>用户帐号：  
            <input type="text"  name="username" id="username" value=""/>  
            <br/>登录密码：  
            <input type="password" name="password" id="password" value="" />                            <br/><input value="登录" type="submit" >  
</form>  
g)    登出页面  
<%  
            SecurityUtils.getSubject().logout();  
%>

五、在Shiro中实现CAPTCHA（验证码）功能

a)     验证码表单认证过滤器

**package** com.wearereading.example.shiro;

**import**javax.servlet.ServletRequest;

**import**javax.servlet.ServletResponse;

**import**org.apache.shiro.authc.AuthenticationToken;

**import**org.apache.shiro.web.filter.authc.FormAuthenticationFilter;

**import**org.apache.shiro.web.util.WebUtils;

**public** **class**CaptchaFormAuthenticationFilter **extends** FormAuthenticationFilter{

**public** **static** **final** String *DEFAULT\_CAPTCHA\_PARAM* = "captcha";

**private** String captchaParam = *DEFAULT\_CAPTCHA\_PARAM*;

**public** String getCaptchaParam() {

**return** captchaParam;

    }

**protected** String getCaptcha(ServletRequest request) {

**return** WebUtils.*getCleanParam*(request, getCaptchaParam());

    }

**protected** AuthenticationToken createToken(

ServletRequest request, ServletResponse response) {

        String username = getUsername(request);

        String password = getPassword(request);

        String captcha = getCaptcha(request);

**boolean** rememberMe = isRememberMe(request);

        String host = getHost(request);

**return** **new** CaptchaUsernamePasswordToken(

username, password, rememberMe, host,captcha);

    }

}

b)     用户名密码令牌UsernamePasswordToken

**package** com.wearereading.example.shiro;

**import**org.apache.shiro.authc.UsernamePasswordToken;

**public** **class**CaptchaUsernamePasswordToken **extends** UsernamePasswordToken{

**private** **static** **final** **long** *serialVersionUID* = 1L;

**private** String captcha;

**public** String getCaptcha() {

**return** captcha;

      }

**public** **void** setCaptcha(String captcha) {

**this**.captcha = captcha;

      }

**public** CaptchaUsernamePasswordToken() {

**super**();

      }

**public** CaptchaUsernamePasswordToken(String username, **char**[] password,

**boolean** rememberMe, String host,String captcha) {

**super**(username, password, rememberMe, host);

**this**.captcha = captcha;

      }

}

c)     添加AuthenticationException

**public** **class**IncorrectCaptchaException **extends** AuthenticationException{

**private** **static** **final** **long** *serialVersionUID* = 1L;

**public** IncorrectCaptchaException() {

**super**();

      }

**public** IncorrectCaptchaException(String message, Throwable cause) {

**super**(message, cause);

      }

**public** IncorrectCaptchaException(String message) {

**super**(message);

      }

**public** IncorrectCaptchaException(Throwable cause) {

**super**(cause);

      }

}

d)     Shiro INI文件

authc= **com.wearereading.example.shiro.CaptchaFormAuthenticationFilter**

e)     实现Realm

**protected**AuthenticationInfo doGetAuthenticationInfo(

                 AuthenticationToken authcToken ) **throws**AuthenticationException {

           CaptchaUsernamePasswordToken token =

(CaptchaUsernamePasswordToken) authcToken;

           String accountName = token.getUsername();

                 //验证码 验证

                 String captcha = **null**;

                 Object obj\_captcha = SecurityUtils.*getSubject*().getSession()

.getAttribute( SessionKey.*CAPTCHA* );

                 Object obj\_count = SecurityUtils.*getSubject*().getSession()

.getAttribute( SessionKey.*LOGIN\_FAILED\_COUNT* );

**int** failed\_count = (obj\_count == **null** || !(obj\_count **instanceof** Integer))

?0:(Integer)obj\_count;

**if**( obj\_captcha **instanceof** String)

                      captcha = (String)obj\_captcha;

**if**( captcha != **null** && failed\_count >0

                      && !captcha.equalsIgnoreCase( token.getCaptcha() )){

**throw** **new**IncorrectCaptchaException("验证码错误！");

           }

           //用户名密码验证

**if**( accountName != **null** && !"".equals(accountName) ){

                 AccountManager accountManager = **new**AccountManagerImpl();

                 Account account = accountManager.get( token.getUsername() );

**if**( account != **null** )

**return** **new** SimpleAuthenticationInfo(

                                  account.getName(),account.getPassword(), getName() );

           }

**return** **null**;

      }

}

f)      登录页面

<%

           Object obj = request.getAttribute(org.apache.shiro.web.filter.authc.

FormAuthenticationFilter.DEFAULT\_ERROR\_KEY\_ATTRIBUTE\_NAME);

**boolean** flag = **false**;

           String msg = "";

**if**( obj != **null** ){

**if**( "org.apache.shiro.authc.UnknownAccountException".equals( obj ) )

                      msg = "未知帐号错误！";

**else** **if**("org.apache.shiro.authc.IncorrectCredentialsException".equals( obj ))

                      msg = "密码错误！";

**else** **if**("com.wearereading.example.shiro.IncorrectCaptchaException".equals( obj ))

                      msg = "验证码错误！";

**else** **if**( "org.apache.shiro.authc.AuthenticationException".equals( obj ))

                      msg = "认证失败！";

                 flag = !"".equals(msg);

           }

**if**( flag ){

                 out.print( msg );

                 Integer count = (Integer)request.getSession().getAttribute(

SessionKey.LOGIN\_FAILED\_COUNT );

**if**( count == **null** )

                 count = Integer.valueOf(0);

                 count++;

                 request.getSession().setAttribute(SessionKey.LOGIN\_FAILED\_COUNT, count );

           }

%>

<form action=*"login.jsp"* method=*"post"*>

           <br/>用户帐号：

           <input type=*"text"*  name=*"username"* id=*"username"* value=*""*/>

           <br/>登录密码：

           <input type=*"password"* name=*"password"* id=*"password"* value=*""* />

           <br/>验证码：

           <input type="text" name="captcha" id="captcha" size="6"/>

           <img src="/captcha" alt="captcha" />

           <br/><input value=*"登录"* type=*"submit"* >

</form>

g)     CAPTCHA实现

h)

六、代码的开发环境

JAVA1.6

Tomcat

Eclipse

七、参考资料

http://www.captcha.net/

将 Shiro作为应用的权限基础

http://www.ibm.com/developerworks/cn/opensource/os-cn-shiro/index.html

**转自：http://blog.sina.com.cn/minssh。**