**Shiro**

# 授权

**授权**，也叫**访问控制，即在应用中控制谁访问哪些资源**（如访问页面/编辑数据/页面操作 等）。在授权中需了解的几个关键对象：主体（Subject）、资源（Resource）、权限

（Permission）、角色（Role）。

• **主体(Subject)：**访问应用的用户，在 Shiro 中使用 **Subject** 代表该用户。用户只有授权 后才允许访问相应的资源。

• **资源(Resource)：在应用中用户可以访问的 URL**，比如访问 JSP 页面、查看/编辑某些 数据、访问某个业务方法、打印文本等等都是资源。用户只要授权后才能访问。

• **权限(Permission)：**安全策略中的原子授权单位，通过权限我们可以表示在应用中用户 有没有操作某个资源的权力。即**权限表示在应用中用户能不能访问某个资源**，如：访问用 户列表页面查看/新增/修改/删除用户数据（即很多时候都是CRUD（增查改删）式权限控 制）等。权限代表了用户有没有操作某个资源的权利，即反映在某个资源上的操作允不允 许。

• Shiro 支持粗粒度权限（如用户模块的所有权限）和细粒度权限（操作某个用户的权限，

即实例级别的）

• **角色(Role)**：**权限的集合**，一般情况下会赋予用户角色而不是权限，即这样用户可以拥有 一组权限，赋予权限时比较方便。典型的如：项目经理、技术总监、CTO、开发工程师等 都是角色，不同的角色拥有一组不同的权限。

## 授权方式

**Shiro 支持三种方式的授权：**

**编程式**：通过写if/else 授权代码块完成

**注解式**：通过在执行的Java方法上放置相应的注解完成，没有权限将抛出相应的异常

**JSP/GSP 标签**：在JSP/GSP 页面通过相应的标签完成

# 拦截器

Shiro 内置了很多默认的拦截器，比如身份验证、授权等 相关的。默认拦截器可以参考 org.apache.shiro.web.filter.mgt.DefaultFilter中的枚举 拦截器：

## 身份验证相关的

## 授权相关的

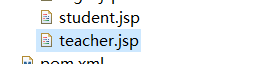
# 需求实现

用户认证（登录）成功后，根据不同角色，访问对应资源。

例如：

老师用户 --》对应老师角色，访问老师相应资源

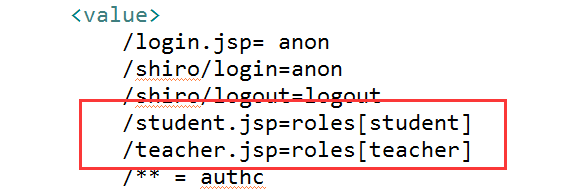
学生用户--》对应学生角色，访问学生相应资源



## 授权需求实现

步骤：

##### 1、在spring\_shiro.xml中配置资源和页面相应信息：



**此时再访问该页面，将会发现无权限操作**

##### 2、Realm类继承AuthorizingRealm



##### 3、编码实现：



# 标签

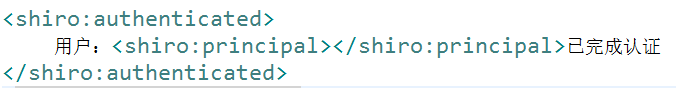
**Shiro 标签**

Shiro 提供了 JSTL 标签用于在 JSP 页面进行权限控制，如

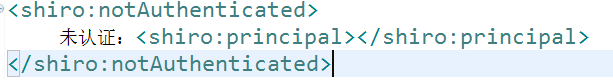
根据登录用户显示相应的页面按钮。



authenticated 标签：用户已经身份验证通过，即 Subject.login登录成功



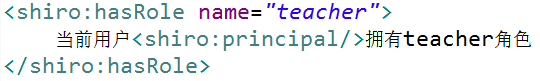
notAuthenticated 标签：用户未进行身份验证，即没有调 用Subject.login进行登录



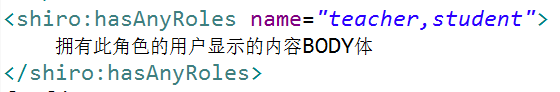
pincipal 标签：***显示用户身份信息***，默认调用 Subject.getPrincipal() 获取，即Primary Principal



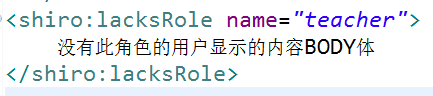
hasRole 标签：如果当前 Subject 有角色将显示 body 体内容：



hasAnyRoles 标签：如果当前Subject有任意一个 角色（或的关系）将显示body体内容



lacksRole：如果当前 Subject 没有角色将显 示 body 体内容



hasPermission：如果当前 Subject 有权限 将显示 body 体内容

lacksPermission：如果当前Subject没有权 限将显示body体内容。

# 数据库存储初始化资源和权限

