RABC

权限管理的方面

用户，角色，资源

资源：菜单，按钮，普通链接

真实项目中，可能部门超级多，人员会有异动离职

递归方式id、parentId

路径的方式path、path/path然后将数据存储在es中

权限分类：

粗粒度：菜单按钮链接

细粒度：数据权限

权限框架不能解决数据问题

例：客户属于谁，谁能看见，谁能操作

CRM系统：客户表，客户属于谁，谁能看见，谁能操作，细粒度权限

划分客户的权限问题，客户身上不一定有部门，级别属于业务范畴，不一定和权限划分，所属公司可以。只存拥有者。

分布式最大的问题就是session共享问题，session需要存在一个公共地方

subject主体（用户视图，第三方进程），认证主体（账号），证书（密码），连接应用程序和shiro的桥梁

Realm与数据库的沟通桥梁，realm可以有多个，实现Realm接口

缓存：缓存一致性，缓存命中率

shiro的缓存

为什么要用缓存？

shiro的集中式缓存

shiro的realm每次鉴权都会走，但是单点登录时不能用缓存，会有数据问题

CacheManager接口，就可以设置缓存。把写好的缓存类set到管理器中，shiro提供了ehCache

缓存管理，可以把shiro的接口拿出去，使用spring的缓存。

缓存有更新的问题?

shiro在更新缓存时，提供了要主动调用它的办法

http超文本传输协议：超： 文本：记事本 文本的确是文本，但是会附加一些功能

http分发密文包

淘宝应用服务器：判断未登录—>定向到SSO服务器

SSO判断用户是否登录，

Spring的Interceptor或者Filter都可以

第一次淘宝进入：

被应用服务拦截，应用服务器session中的登录标识为空，继续

第一次token为空

重定向到SSO服务器checkLogin验证登录接口，直接跳转到SSO验证中心登录页面

SSO服务器校验通过后，浏览器直接重定向请求原路返回，并携带token

如果校验登录成功，SSO服务器端需要保存token

浏览器再次被应用服务拦截器拦截，此时可以拿到token，应用服务器再次拿token到SSO服务器的verity验证token合法接口

这次应用服务器会传token和logoutUrl和jsessionId到SSO验证服务器

这时如果成功，这次登陆验证成功，拦截器正式放行。

第二次天猫进入：

因为没有登录，会直接进入SSO登录验证，SSO从session获取token，可以获取到token。

SSO单点登录系统因为拿到了token所以直接重定向到天猫服务器端，并携带token

登出：

登出和超时，会清空session，使用之前接受过的jsessionId，循环发送到其他服务器，清空session。HttpSessionListener session.invalidate();