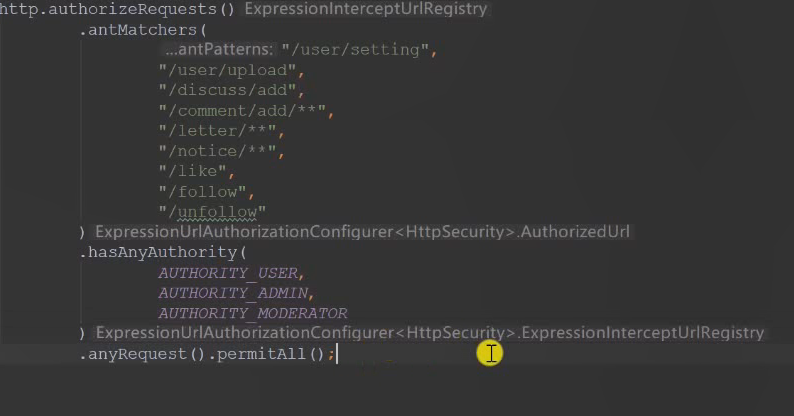
废弃之前拦截器实现登录检查

先ignore静态资源

**设置授权**

在security的**配置类中设置授权**，三个方法分别是：

1.设置需要授权的路径 2.访问所需要的身份(三者任意一个都可以) 3.除了这些路径，其他路径统一设置成不需要授权



**权限不够时的处理**

**没有登录时**，首先**需要判断访问的功能是需要异步返回json字符串还是页面**

1. 异步：返回一个json字符串状态码为403，错误信息是你还没有登录；

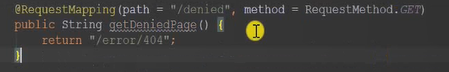
2. 需要返回页面：返回到登录页面，让其登录（将以前的用拦截器检查是否登录改为了用security检查）



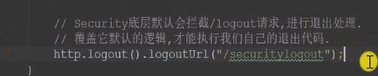
**权限不足**，比如你普通用户访问了管理员才能访问的功能

同样也要判断返回的格式是json还是页面，若为json则返回403状态码，信息为你没有访问此功能的权限，若为页面则返回一个404页面



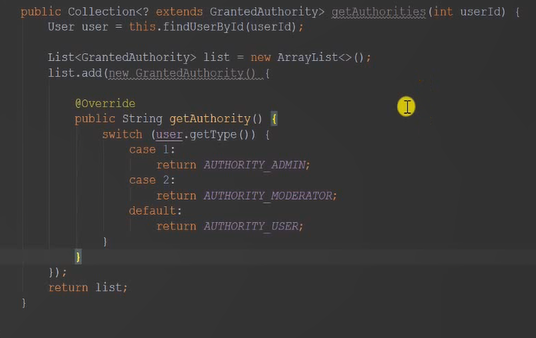


**Security底层默认会拦截退出请求，直接进行退出**，而不走我们自己写的退出代码，因为security是通过filter进行拦截的，拦截较为靠前，security默认的退出路径为/logout，**可以给security传一个假的退出路径，这样就不会拦截我们的退出请求**



我们绕过了security的认证，但security的认证会将信息（用户、权限等等）传给securityContent对象，供security使用，我们想用security来授权就需**要知道每个用户的权限是什么，所以需要给security传入用户的权限(也就是这件事也被绕过了但也要做)**

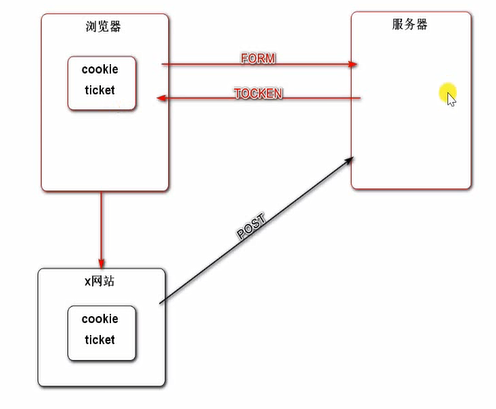
首先需要在userService中写一个查询用户权限的方法，而且需要符合security的格式，将其所拥有的权限加入到集合中



在**检查登录凭证的拦截器**（intercepter）中通过调用userService的查询权限方法**将权限传给security**（请求结束和退出程序后将其清理掉，节省空间）



**csrf攻击**



**csrf：**浏览器访问服务器后得到cookie，之后访问非法网站之后，cookie和里面的ticket(凭证)被盗取。之后非法网站模拟你的身份用post表单向服务器提交数据。

如果提交的是转账的数据就很危险有可能将钱划走

**security解决方案**：会在返回给浏览器的表单里添加一个tocken，tocken每次请求都不一样，是随机生成的，非法网站能窃取cookie但不能猜到tocken，所以post提交数据的时候，服务器看到ticket对但没有tocken则判断为csrf攻击，提交失败（但security只能在非异步请求，在返回网页时自动传一个tocken，异步时需要咱们自己处理，前端模板上编写逻辑加入tocken ）

在post请求时服务器给客户端一个表单，供其填写数据写入服务器，但cookies在用户端并不安全，可能会被非法网站窃取到，之后非法网站通过这个cookie提交数据给服务器，如果涉及到金额问题，甚至有可能将钱划走（网上购买物品时，非法获取到cookie，去买自己商铺的商品？？？这样是两个表单，一个是你要买的物品付款表单，一个是非法的物品购买表单，tocken不同，非法的失败（不知道这样理解对不对）），所以security在给你表单的时候还会给你一个tocken，每个表单都不一样，在提交时会判断cookie和tocken，若不符合，则判断为csrf攻击，提交失败（但security只能在非异步请求，在返回网页时自动传一个tocken，异步时需要咱们自己处理，前端模板上编写逻辑加入tocken ）