# Spring security 的认证流程

## 认证流程解析

### 认证流程

1）过滤器拦截登录请求， 获取到用户名和密码，封装成Authentication。（通常情况下是UsernamePasswordAuthenticationToken这个实现类）

1. AuthenticationManager身份管理器负责验证这个Authentication，验证失败返回

3）认证成功后，AuthenticationManager身份管理器返回一个被填充了信息的Authentication实例（填充的信息包括授权）

4）SecurityContexHolder安全上下文容器将Authentication,通过SecurityContextHolder.getContext().setAuthentication()方法设置到其中

### 2.SecurityContextHolder解析

SecurityContextHolder用于存储安全上下文（security context）的信息。当前操作的用户是谁，该用户是否已经被认证，他拥有哪些角色权限…这些都被保存在SecurityContextHolder中。

### 3.AutenticationManager解析

AuthenticationManager是一个认证相关的核心接口，该类一般不直接进行认证，而是通过其实现类ProviderManager中的一个认证方式列表（List<AuthenticationProvider>）来进行认证。也就是认证入口只有一个：AuthenticationManager。认证方式：指纹、用户名+密码、邮箱加+密码等对应不同的AuthenticationProvider，默认策略下，只需要通过一个AutenticationProvider认证，认证成功则登录

### 4.UserDetails与UserDetailsService解析

## 二、授权解析

### 1.通过表达式控制URL权限

重写configure(HttpSecurity http)方法，

1)Anthenticated():保护url，需要用户登录。

2)permitAll():指定某些url不进行保护，一般针对静态文件和注册等授权情况下需要访问的页面

3)hasRole(String role):限制单个角色访问。

4)hasAnyrole(String … roles):允许多个角色访问。

5)access(String atribute):该方法可以穿件复杂的限制,比如可以增加RBAC的权限表达式

6)hasIpAddress(String ipaddressExpression):用于限制IP地址或者子网

### 2.通过表达式控制方法权限

在集成WebSecurityConfigurerAdapter类上加上EableGlobalMethodSecurity()注解，配置方法控制的医用类型，支持三种：

@EnableGlobalMethodSecurity(jsr250Enabled=true):开启JSR-250。

@ EnableGlobalMethodSecurity(prePostEnabled=true):开启prePostEnabled。

@ EnableGlobalMethodSecurity(prePostEnabled=true):开启sercured。

### 3.认识JWT

流程：

1. 用户使用浏览器发送账号和密码进行登录
2. 服务器使用私钥创建一个JWT
3. 服务器返回该JWT给浏览器
4. 浏览器将该JWT串在请求头中向服务器发送请求
5. 服务器验证该JWT
6. 根据授权规则返回资源给浏览器

实质就是生成了TOKEN，TOKEN在有效期内，客户端可以直接通过TOKEN校验来获取资源