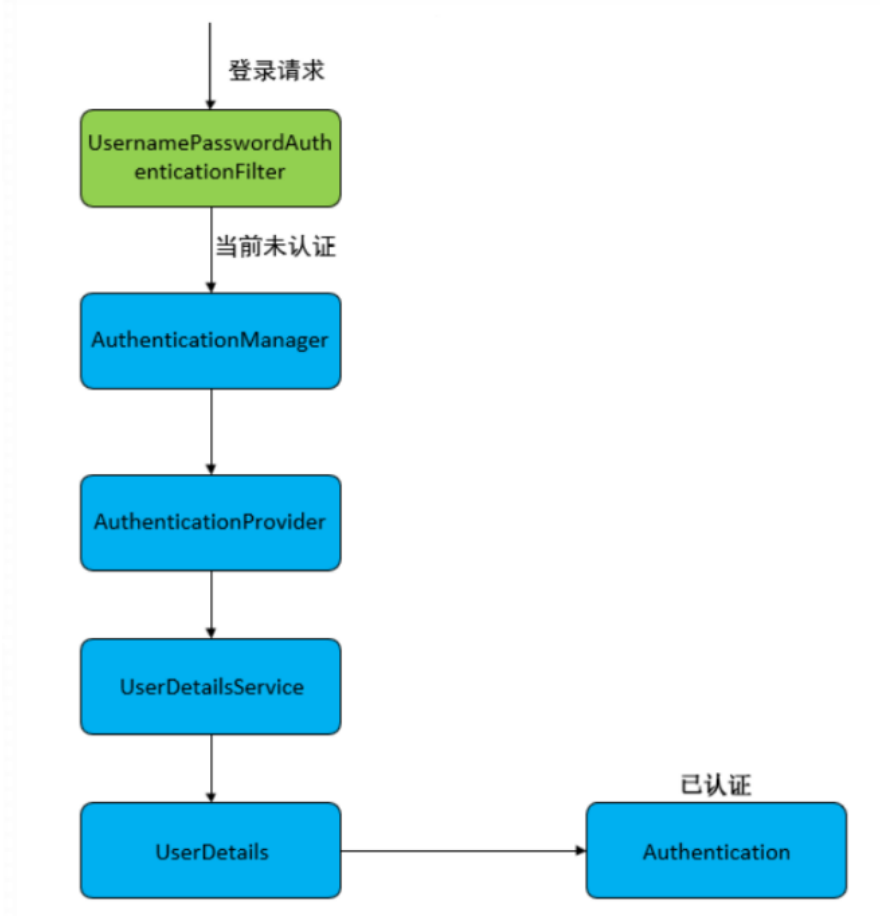
【第30话：很多人都回答不清楚的Spring Security认证流程】

Hello 小伙伴们，这结课给大家带来一个很多小伙伴都害怕的面试题：“Spring Security 认证流程”。

话不多说，我们先直接上图



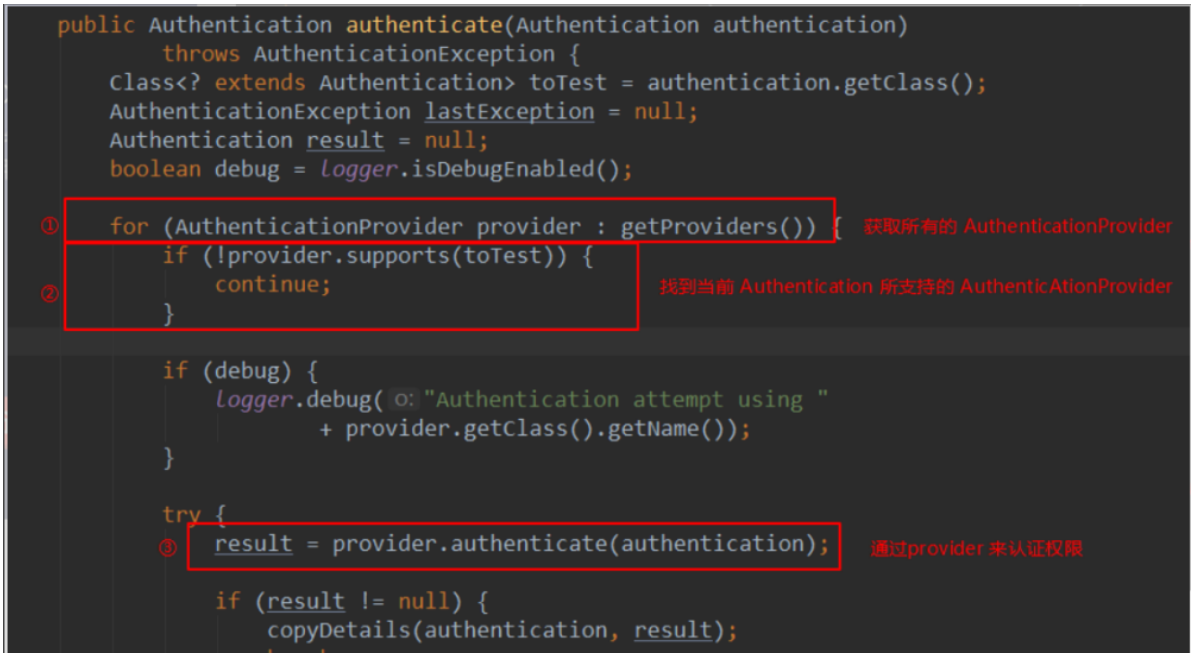
下面我们跟下源码

1. 用户发起表单登录请求后，首先进入 UsernamePasswordAuthenticationFilter

在 UsernamePasswordAuthenticationFilter 中根据用户输入的用户名、密码构建了 UsernamePasswordAuthenticationToken，并将其交给 AuthenticationManager 来进行认证处理。

AuthenticationManager 本身不包含认证逻辑，其核心是用来管理所有的 AuthenticationProvider，通过交由合适的 AuthenticationProvider 来实现认证。

2.下面跳转到了 ProviderManager ，该类是 AuthenticationManager 的实现类



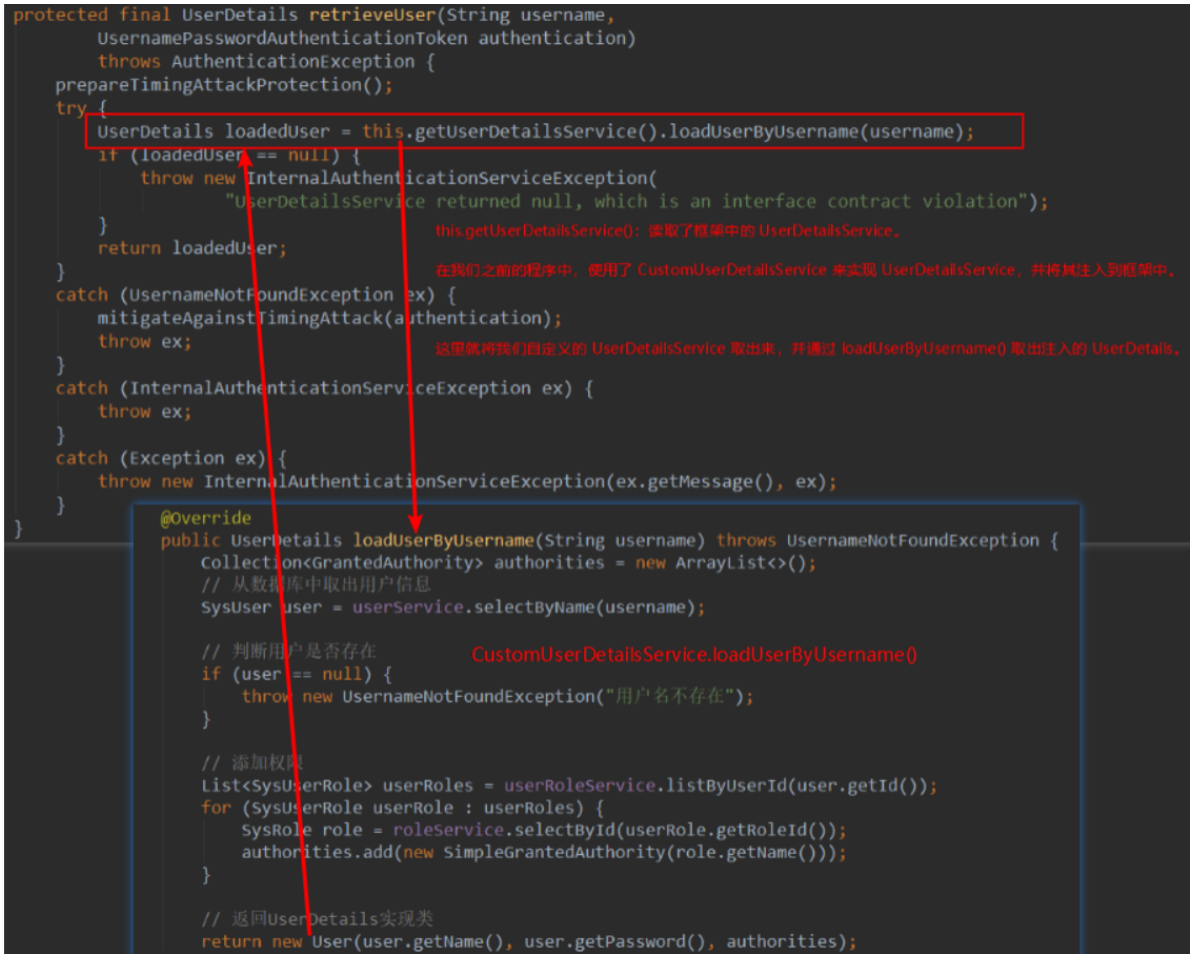
我们知道不同的登录逻辑它的认证方式是不一样的，比如我们表单登录需要认证用户名和密码，但是当我们使用三方登录时就不需要验证密码。

Spring Security 支持多种认证逻辑，每一种认证逻辑的认证方式其实就是一种 AuthenticationProvider。通过 getProviders() 方法就能获取所有的 AuthenticationProvider，通过 provider.supports() 来判断 provider 是否支持当前的认证逻辑。当选择好一个合适的 AuthenticationProvider 后，通过 provider.authenticate(authentication) 来让 AuthenticationProvider 进行认证。

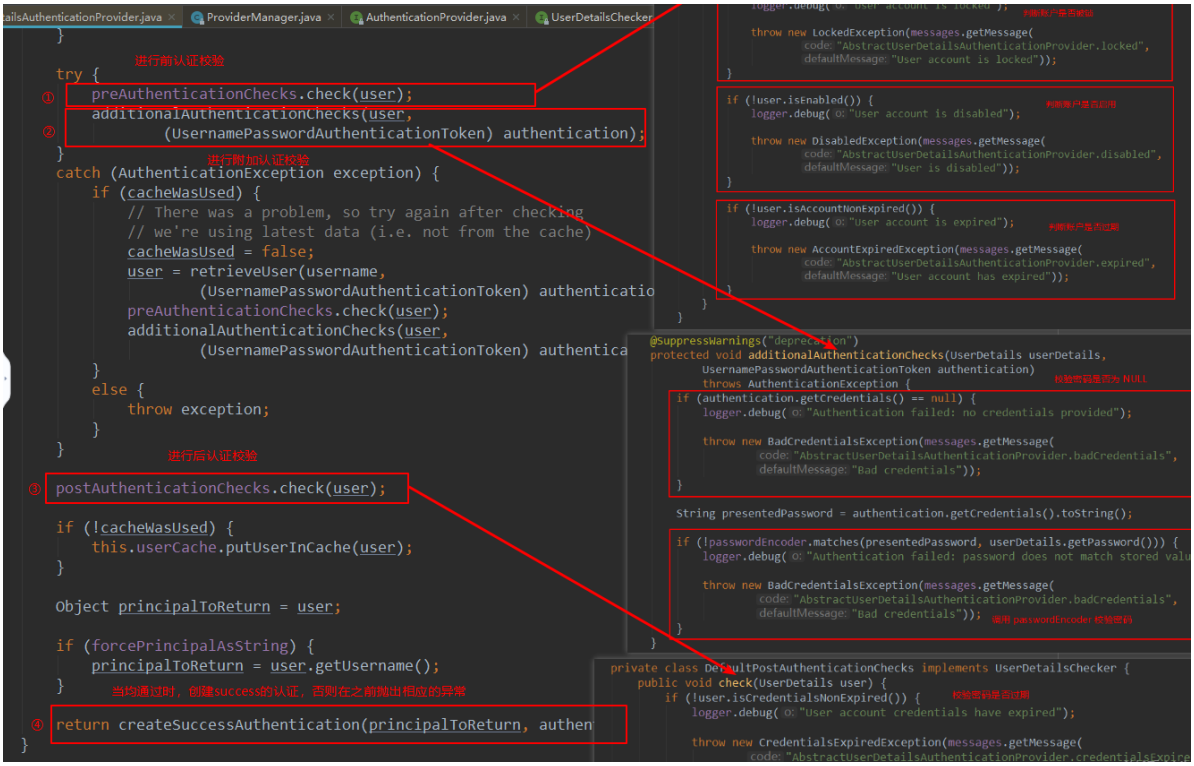
3.传统表单登录的 AuthenticationProvider 主要是由 AbstractUserDetailsAuthenticationProvider 来进行处理的，我们来看下它的 authenticate()方法。

```
1 | user = retrieveUser(username,(UsernamePasswordAuthenticationToken) authentication);
```

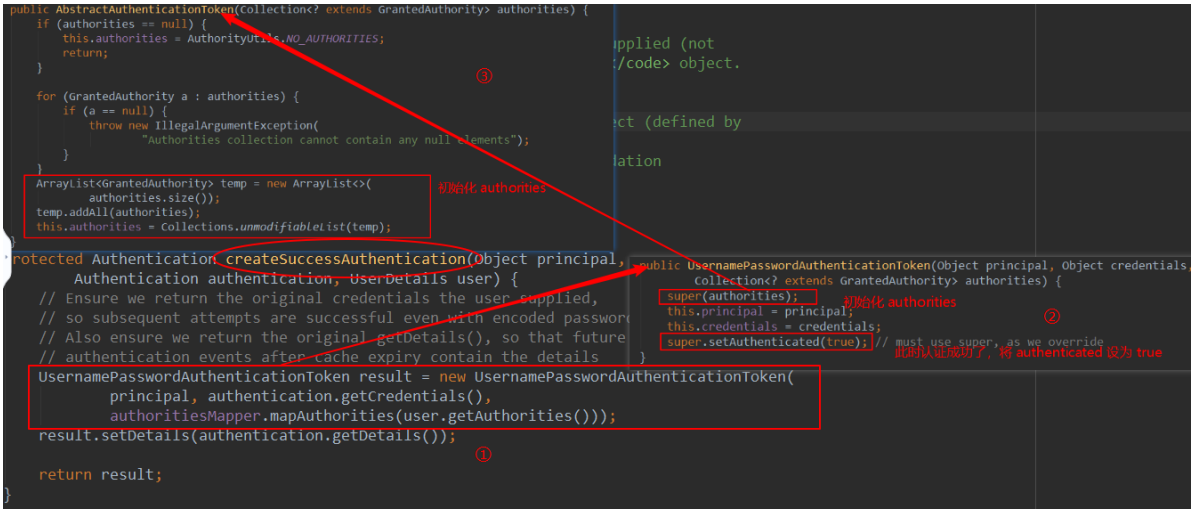
retrieveUser() 的具体实现在 DaoAuthenticationProvider 中，代码如下：



当我们成功的读取 UserDetails 后，下面开始对其进行认证：

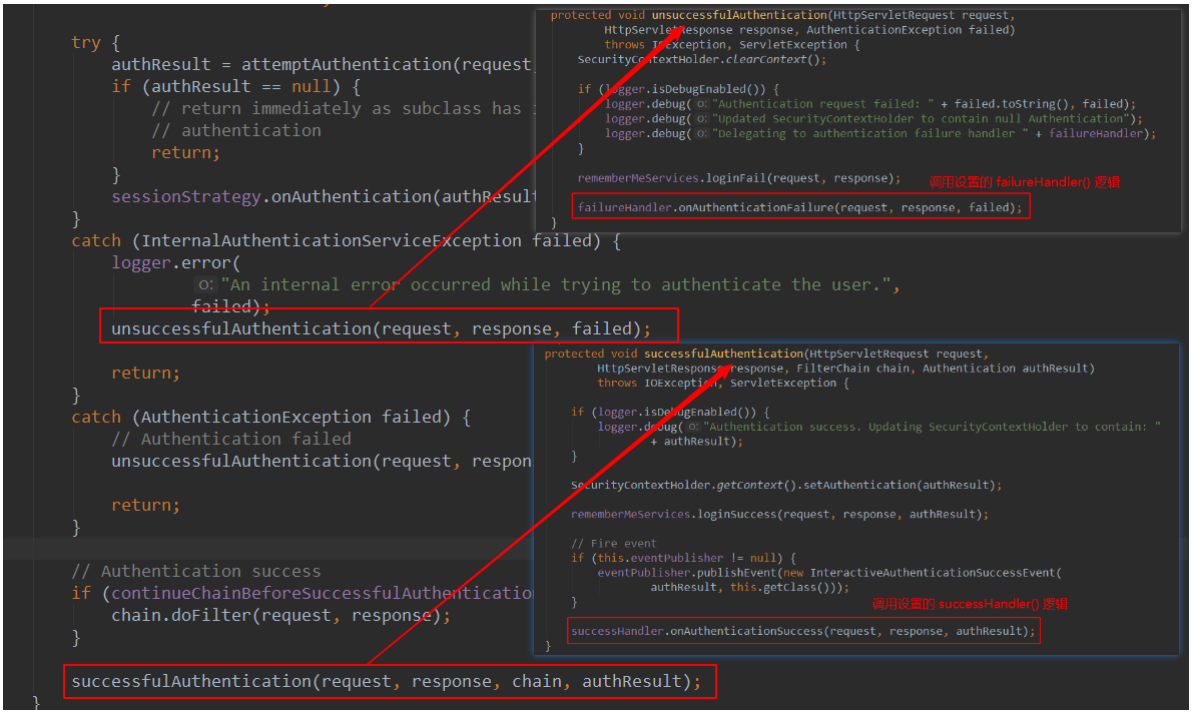


在上图中，我们可以看到认证校验分为 前校验、附加校验和后校验，如果任何一个校验出错，就会抛出相应的异常。所有校验都通过后，调用 createSuccessAuthentication() 返回认证信息



在 createSuccessAuthentication 方法中，我们发现它重新 new 了一个 UsernamePasswordAuthenticationToken，因为到这里认证已经通过了，所以将 authorities 注入进去，并设置 authenticated 为 true，即需要认证。

4.至此认证信息就被传递回 UsernamePasswordAuthenticationFilter 中，在 UsernamePasswordAuthenticationFilter 的父类 AbstractAuthenticationProcessingFilter 的 doFilter() 中，会根据认证的成功或者失败调用相应的 handler



这里调用的 handler 实际就是我们在配置文件中配置的 successHandler() 和 failureHandler()。

很多小伙伴就怕源码分析过程，但是这类问题是我们绕不过去的一道坎，面试官通过问这种原理性问题可以了解到面试者对于所学框架或技术的理解程度，也在考验我们平时是否跟踪源码。

Spring Security 认证流程回答示范

为了让小伙伴吗，更加容易记忆，给大家准备了文字版认证流程说明。

（1）用户在浏览器中随意输入一个URL

（2）Spring Security 会判断当前是否已经被认证（登录）如果已经认证，正常访问URL。如果没有被认证跳转到（AbstractAuthenticationFilterConfigurer的loginPage属性所对应的值）登录页面。

（3）在登录页面中输入用户名和密码后，点击登录按钮后，执行登录

（4）如果url是登录的url（AbstractAuthenticationFilterConfigurer的loginProcessingUrl）执行认证流程。否则重新显示认证页面。

（5）执行登录流程时通过过滤器UsernamePasswordAuthenticationFilter进行过滤，取出用户名和密码，放入到容器UsernamePasswordAuthenticationToken中。之后交给认证管理器AuthenticationManager去调用认证提供器AuthenticationProvider。

（6）由提供器AuthenticationProvider调用自定义登录逻辑UserDetailsService，返回结果UserDetails

（7）使用密码解析器PasswordEncoder，对UserDetails中的密码和客户端传递过来的密码进行匹配，如果匹配失败抛出异常BadCredentialsException，如果认证成功，跳转到对应页面