

<https://docs.spring.io/spring-security/site/docs/5.0.11.BUILD-SNAPSHOT/reference/htmlsingle/#core-services>

10.1 AuthenticationManager, ProviderManager and AuthenticationProvider

AuthenticationManager(接口)：利用ProviderManager 来进行处理验证

ProviderManager (类)：Spring Security默认的实现类，利用成员属性List<AuthenticationProvider>，来调用每一个AuthenticationProvider，调用其supports 方法判断是否支持该类型的校验，最终到一个可以处理校验的AuthenticationProvider实例来进行具体的校验。

```
for (AuthenticationProvider provider : getProviders()) {  
    if (!provider.supports(toTest)) {  
        continue;  
    }  
    ...  
    result = provider.authenticate(authentication);  
    ....  
}
```

10.1.2 DaoAuthenticationProvider

把request中的信息拼成的UsernamePasswordAuthenticationToken和使用UserDetailsService获取到的数据库中的信息进行对比，用于验证。

10.2 UserDetailsService Implementations

UserDetails loadUserByUsername(String var1) throws UsernameNotFoundException;

只有这一个方法，目的是返回一个UserDetails实例，就是用于和XXXAuthenticationToken进行校验，需要自己实现该接口，实现获取逻辑。

10.2.2 JdbcDaoImpl

配置数据源，注入即可。

10.3 Password Encoding

用于把数据库中获取的密码进行解密，与request传进来的密码进行比较。

默认：BCryptPasswordEncoder. 加密结果形式：

\$2a\$10\$dXJ3SW6G7P50lGmMkkmwe.20cQQubK3.HZWzG3YB1tIRy.fqvM/BG

创建一个User必需使用一个id，否则会报错：

java.lang.IllegalArgumentException: There is no PasswordEncoder mapped for the id "null"