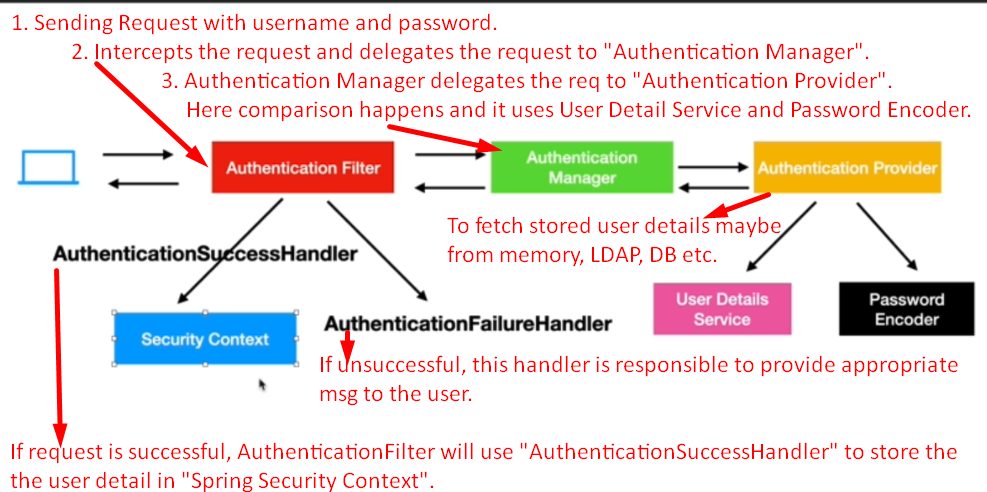
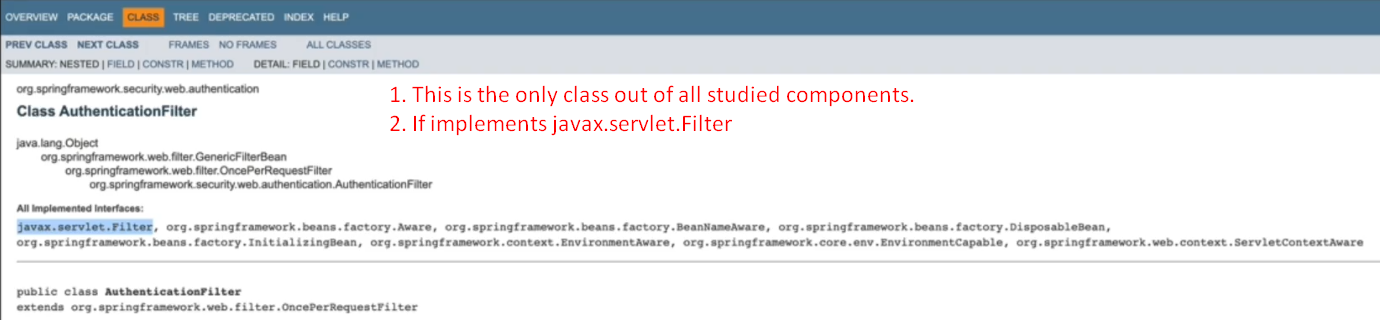
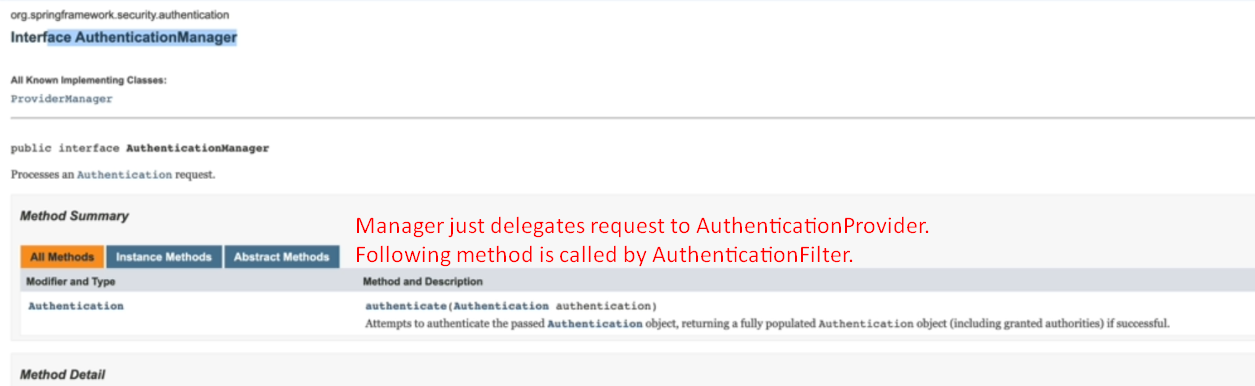
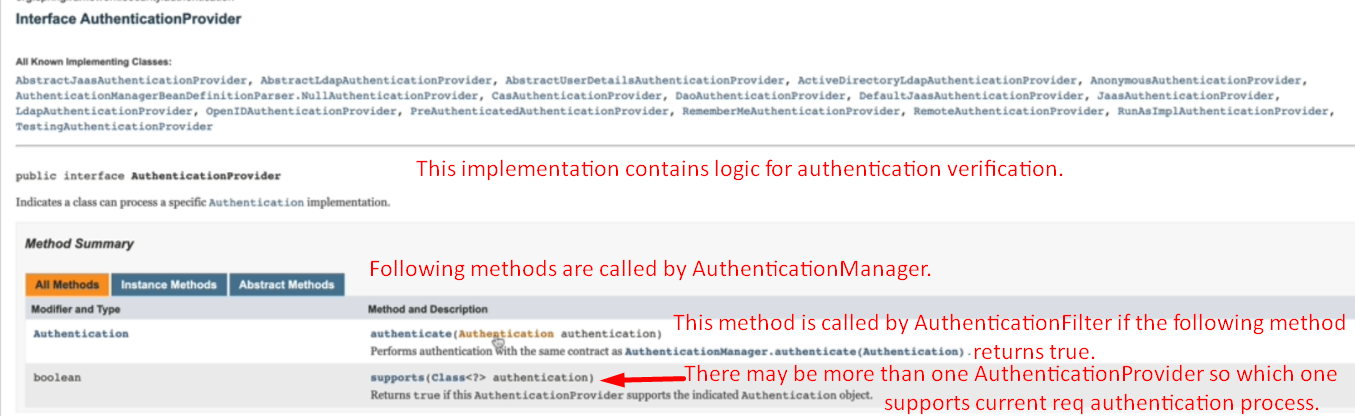
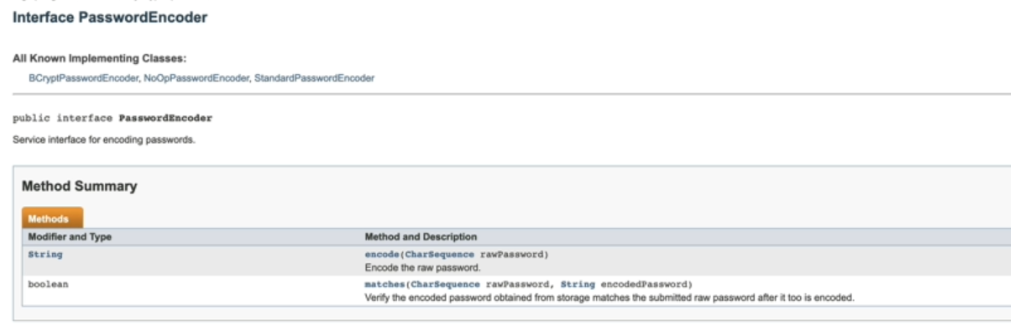
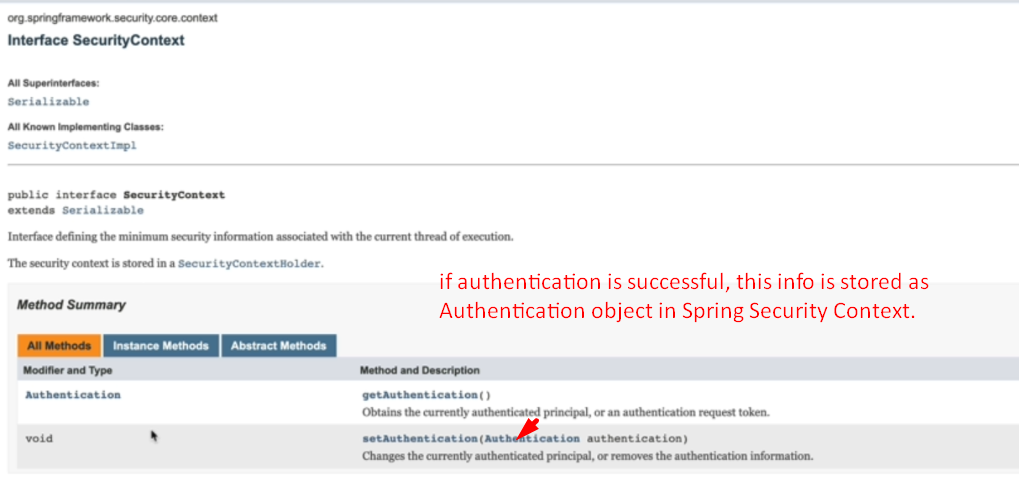
1. As soon as we add Spring Security Dependency, a lot of magic happens behind the scene.
2. In this lecture, we will see all the components involved in that magic.
3. **Components**:



* 1. rg.springframework.security.web.authentication.**AuthenticationFilter**(C):
     1. Servlet Filter **class** by Spring.
     2. This is the very first component in spring security that intercepts the request coming from browser/system.
     3. Responsible to check if the user making request has authenticated or not.
     4. If not authenticated, will delegate the request to **AuthenticationManager** responsible to check validity of username and password.  
        If already authenticated by checking from Spring Security Context, it will not delegate the request.
  2. **Authentication Manager**:
     1. Responsible to check if the username and password are valid or not.
     2. But it delegates the checking to **Authentication Provider.**
  3. **Authentication Provider**:
     1. Containing the real logic to check the validity of username and password.
     2. But it requires username and password (means user details) and it will not fetch them from DB or memory as it is just responsible for comparison.
     3. So, it takes help of **User Detail Service**.
  4. **User Detail Service**:
     1. Responsible to fetch user detail from memory, DB, LDAP server etc.
  5. **Password Encoder**:
     1. For Password Encoding and decoding when storing into DB and Comparison.
  6. **AuthenticationSuccessHandler**:
  7. **AuthenticationFailureHandler**:

1. Let’s study these components by looking at docs.  
   
2. 
3. 
4. 
5. 
6. As you can see all components except for AuthenticationFilter are interface.   
   Spring security provides some default implementations for those interfaces.