May 11, 2017 self experience in ahm security implementation

=========================================================

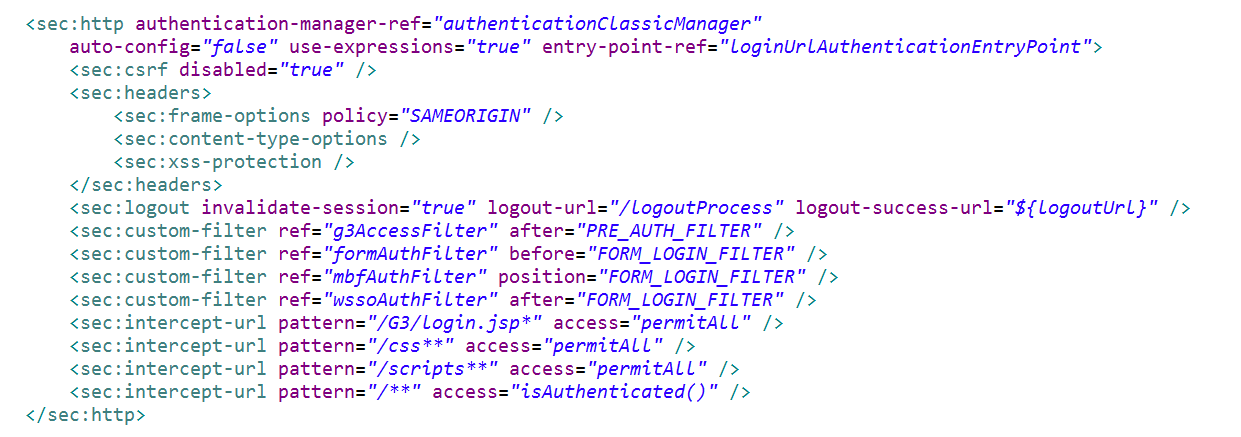
In this week, i was assigned to fix the bug in spring security. Ahm UI is made of angularJs. So ahm spring security is little more complex than normal. The requirement of ahm security should support the following functions.

. support form authentication which is only exposed for internal users

. support mbf/wsso authentication

Let’s start from the configuration. Then walk through the logic in the filter. Next to figure out the relation among those main classes and their functions.

From the above of notes, we know spring security majorly works for web application security. Most of cases we just leverage the default setting and customize some of functions like where is customized login page, what is default target page and failure page etc. the use case of ahm security is the integration of spring security framework with restful api. Let’s start from configuration.



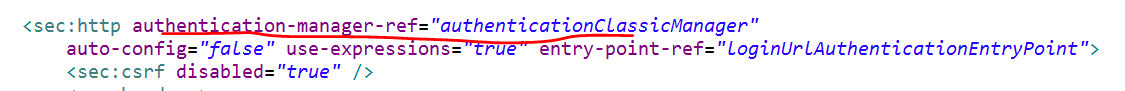
From the configuration above, we can see 3 customized filters defined to handle these 3 different of authentication approaches. If user starts from the login form by entering the user identity, the request hands over to formAuthFilter.

formAuthFilter is extended from UsernamePasswordAuthenticationFilter

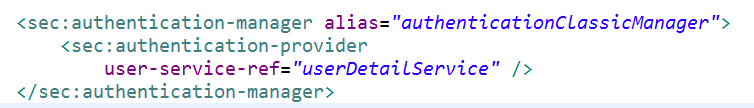
Below is the configuration of customized formAuthFilter



FormAuthFilter extends from UsernamePasswordAuthenticationFilter which extends from AbstractAuthenticationProcessingFilter. In doFilter() method, it wires all the logic together including authentication and authorization which is called Access Control List(ACL). Spring security framework exposes authentication in configuration (which is alias of authenticationManager).



We don’t always override AuthenticationManager. What we need to do is to override UserDetailService to tell framework how to get userDetails. Spring defined the interface of UserDetailsService. We can define our own structure of UserDetail class which contains the data we like to have eg. privileges etc. UserDetailsService exposes only one method loadUserByUsername which returns UserDetails(interface). UserDetails just a container stored all the user information.



Let’s back to the filter. In framework provided AbstractAuthenticationProcessingFilter class, it just calls attemAuthentication(request,response) method to get user’s Authentication(provided by framework) which contains user’s details and principle(can be considered as the same). If not null, it means identity check done. In attemAuthentication method, the user’s identity is set into securityContext for future usage.

Best approach to understand how spring security works. We can review the source code of **AbstractAuthenticationProcessingFilter file.**