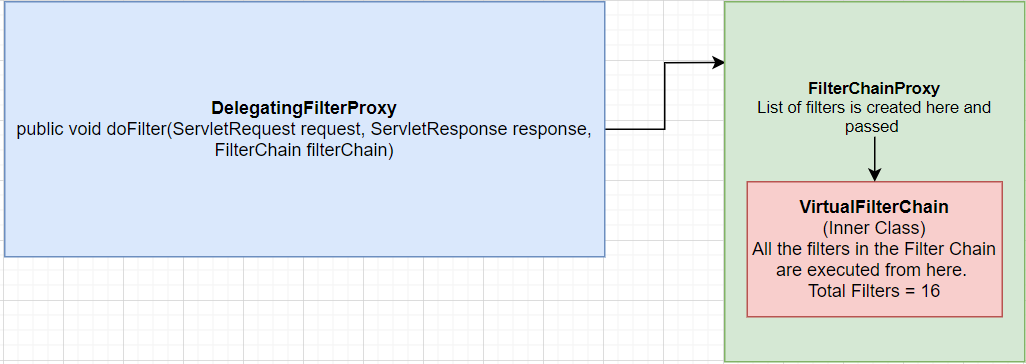
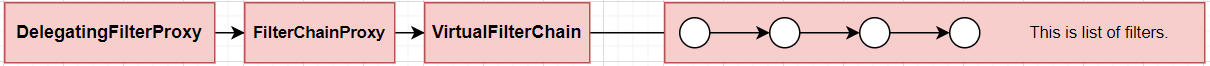
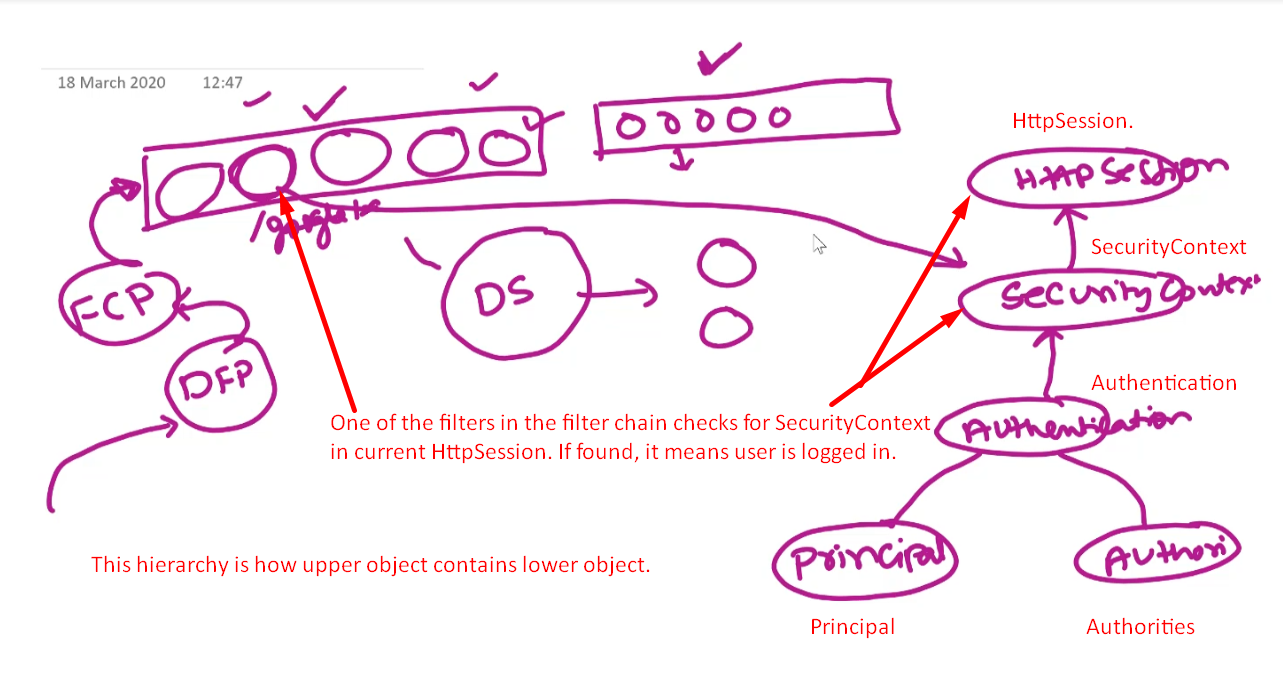
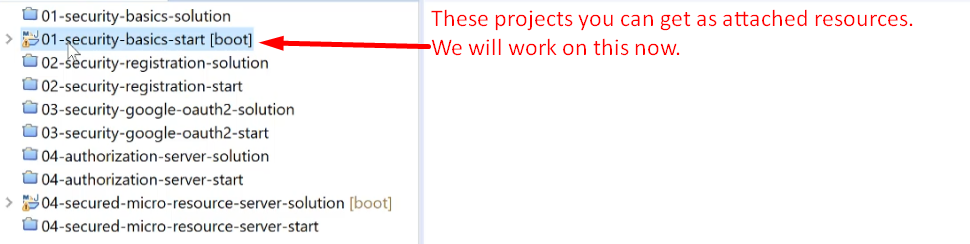
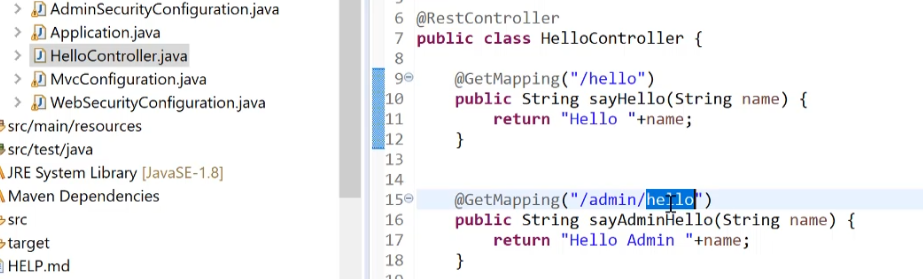
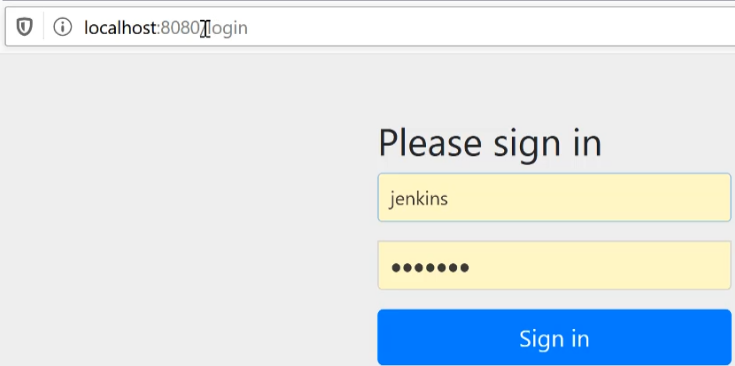
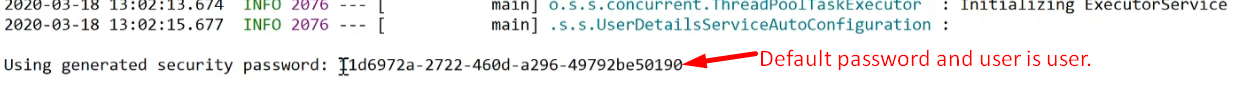
1. Let’s discuss about the basics of Spring Security and its internals.
2. We know all requests go to DispatcherServlet and then to a particular Controller method.
   1. We know first a filter is executed then controller method is called.
   2. So we can write security checking code such as the user making request is authenticated or not and what authorities he has.
3. Spring Security is also implemented using filters.
4. There are a lot of filters in Spring Security.
   1. One filter takes care of Authentication.
   2. One filter takes care of Authorization.
   3. Other one for logout.
   4. Other one for CSRF (Cross Site Request Forgery).
   5. Other one for “Remember Me” functionality.  
      **NOTE**: So we have a chain of filters.”  
      **NOTE**: So to configure this filter chain, we configure only one filter 🡺 **DelegatingFilterProxy**
5. So, we map all the requests to this filter 🡺 **DelegatingFilterProxy**
6. DelegatingFilterProxywill delegate the request to another object called FilterChainProxy.
7. FilterChainProxy 🡺 This class acts as proxy for a chain of filters.
   1. **NOTE**: There may be more than one filter chain.
   2. Why?
   3. **Answer**: Our app supports “form based login” and “Google based login” and “Github based login”. So separate filter chain for each one.  
      If our app supports two kinds of login such as google and github then we will have two filter chains and each filter chain will be mapped to a path such as for google, it filter chain will be mapped to /google.
8. NOTE: We can have multiple filter chains.  
   <https://app.diagrams.net/#Hjatinbansalprogrammer%2FNotes_Accolite_Computer%2Fmaster%2FSpring%2FSecurity%2FDiagram%2FFilterChain>
9.  
11. Let’s understand some Spring Security Terminology.
12. **Terminologies**:
    1. **Principal**: Represents logged in user/device/app.  
       Will be created when an entity successfully logs in.  
       Also **called** Security Principal.  
       Principal = A User Account = A Computer Account = A Group Account.
    2. **Authorities**:
       1. One Principal (a user) can have multiple authorities.
    3. **Authentication**
    4. **SecurityContext**
    5. **HttpSession**

****

1. Let’s see one app without Spring Security.
2. 
3. 
4. So, far no Security Configuration in this project.
5. Just make a request to any URI and you will get result. That is all.
6. 
7. 
8. 
9. 
10. 