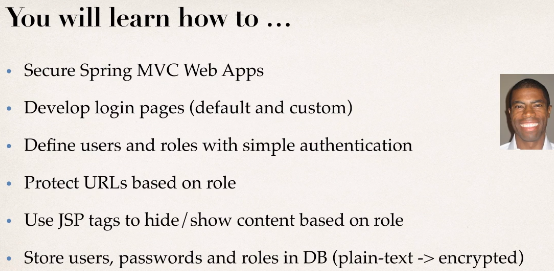
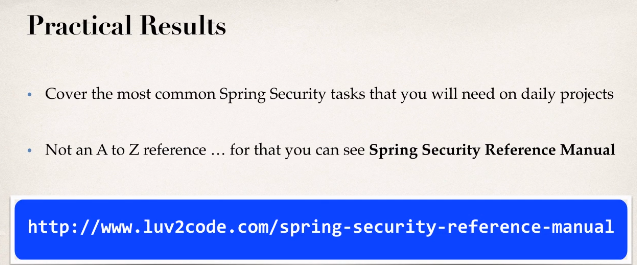
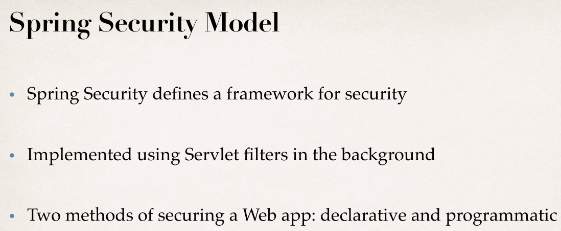
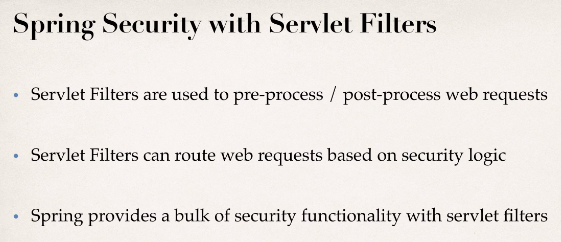
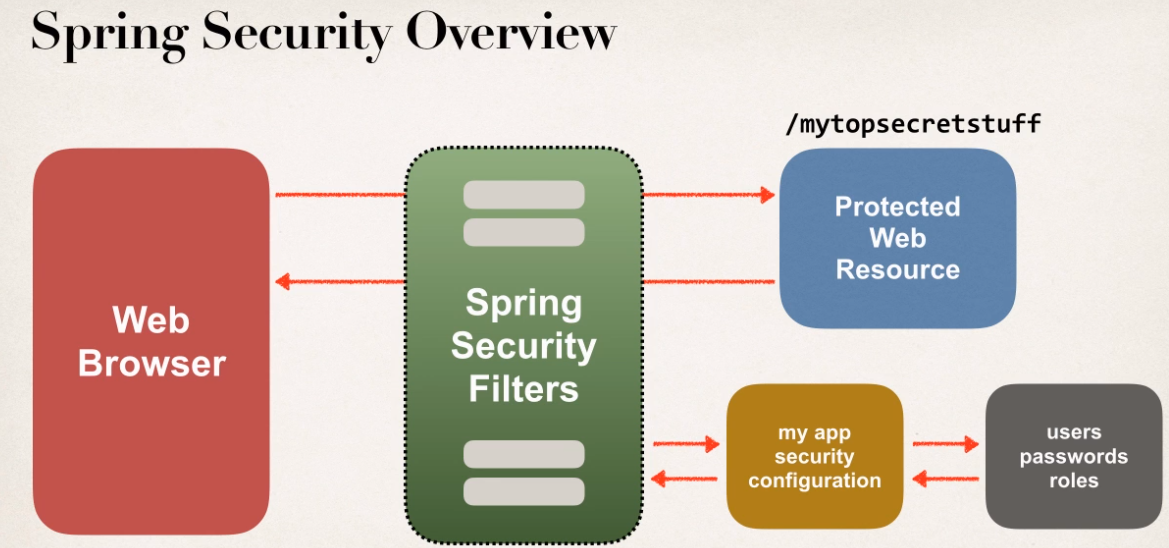
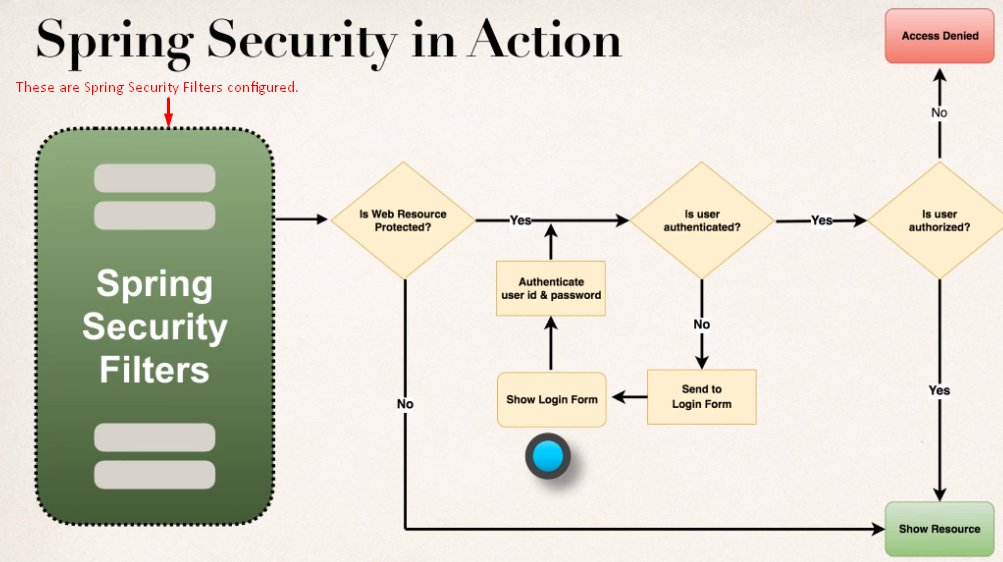
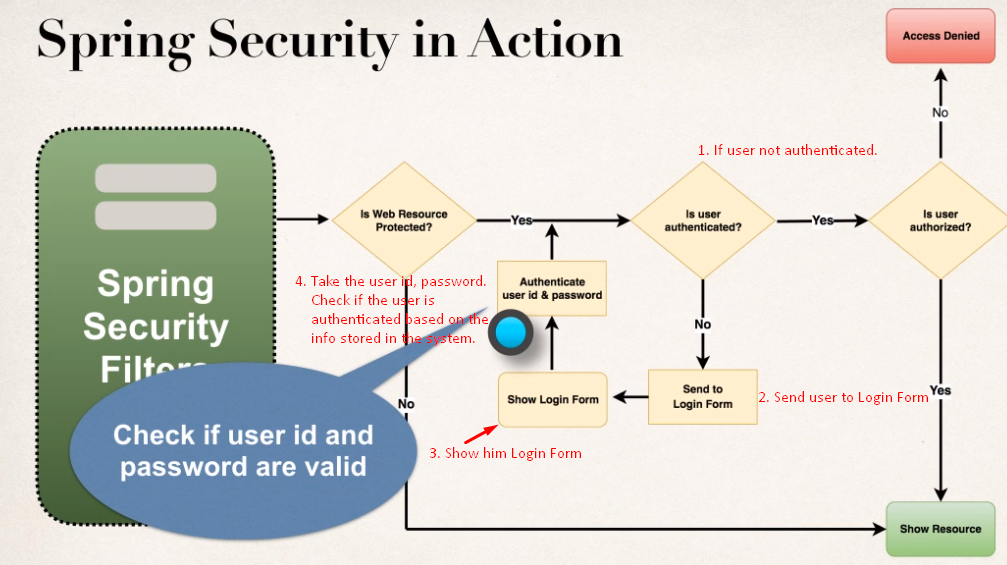
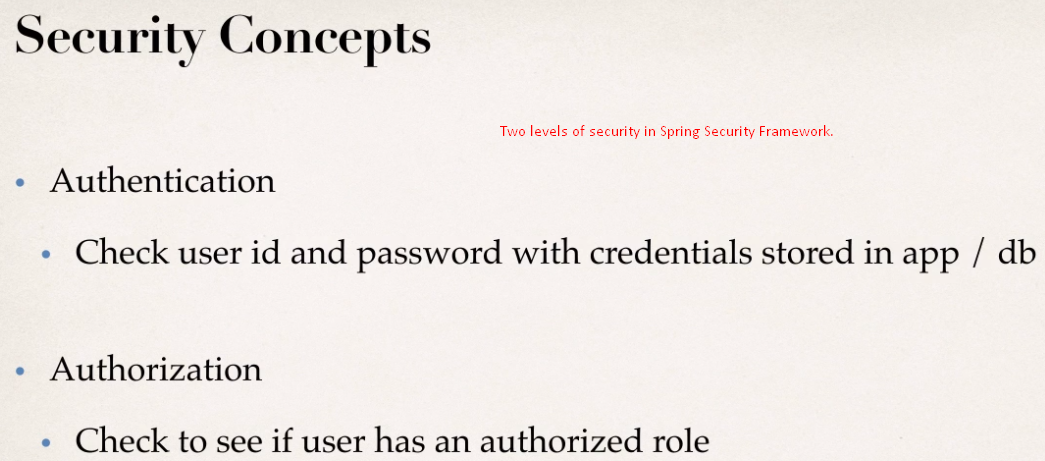
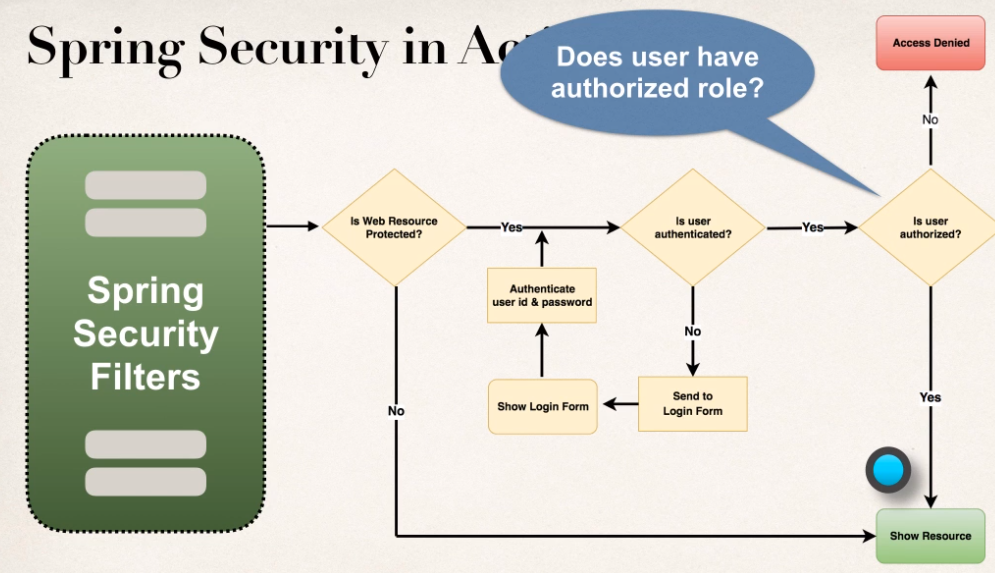
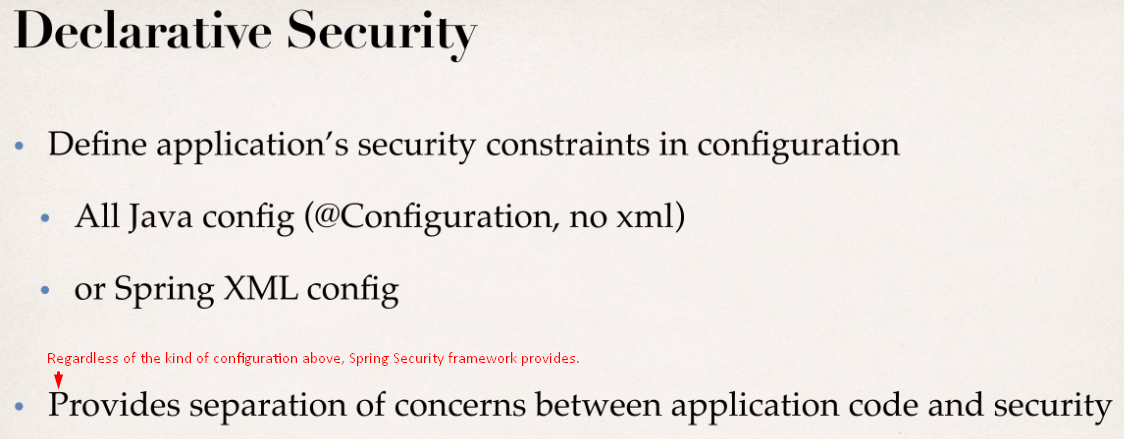
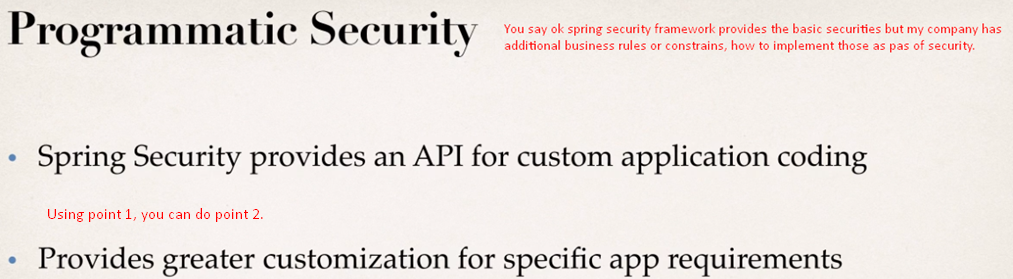
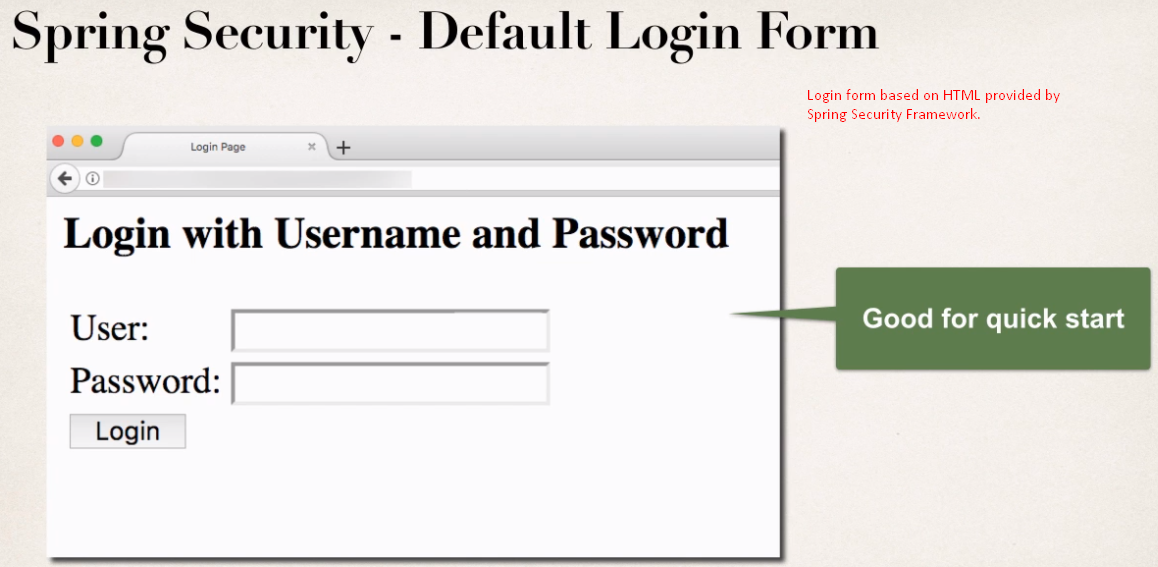
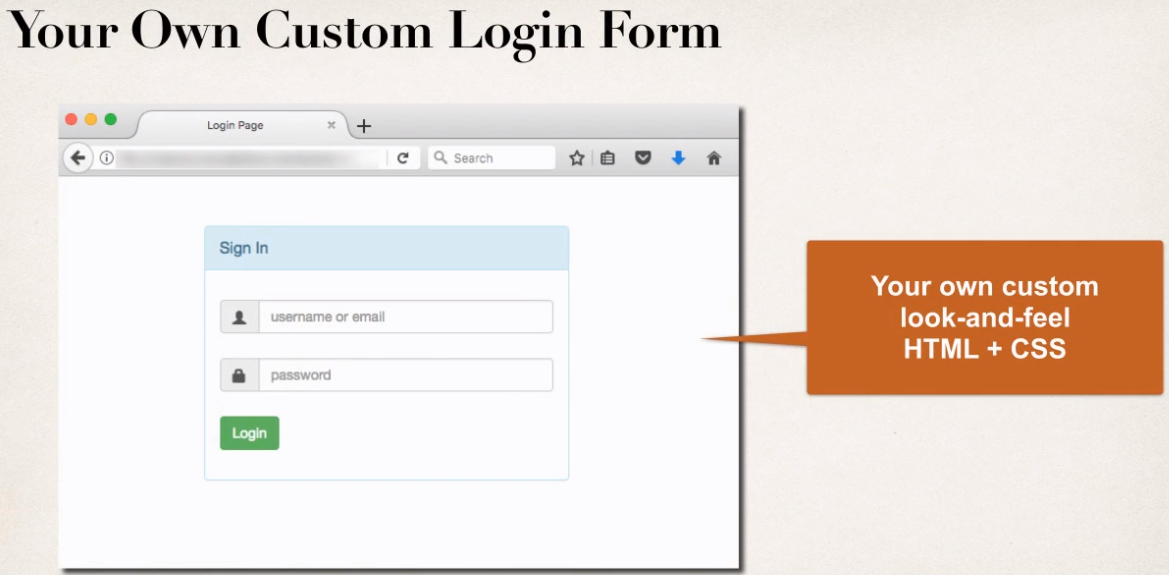
1. 
2. 
3.  
4. **Spring Security Filters**: Intercept the request, pre-process them & check if the user can access that protected web resource.
   1. So, it looks at your **application security configuration**,
   2. And then also it’ll take a look at the **users, passwords & roles** that are in your database to see if a user is authenticated, also if they’re authorized to access this web resource.   
      **NOTE**: This all happens in the background.
5. 
6. 
7. 
8. After authenticating the user, we check if the user is authorized? Does it have authorization to access this resource? Just like the security badge in your University. If you’ve security badge to get into main building. However, you may still have no access to all rooms in the building. It all depends on your **access level** or **authorization role**.  
   So, you’ve have good user id and password, it doesn’t mean you’ve access to everything. There are still additional levels of security in place & Spring has support for that using security roles. So if you don’t have given role, it says “**access denied**” or if you’ve then it would show the resource.   
   
9. 
10.   
    You can easily extend the framework for doing that, you can plug in your own custom security implementation.
11. 
12. 
13. 
14. 
15. 