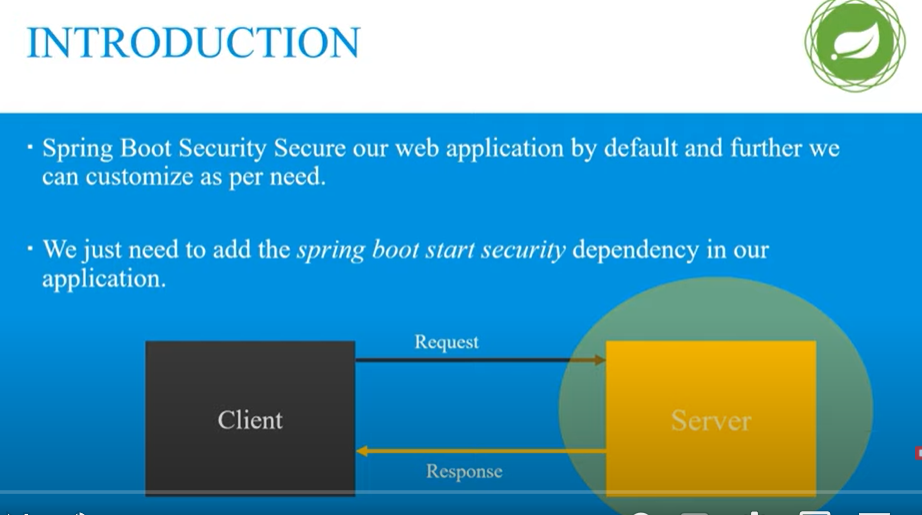
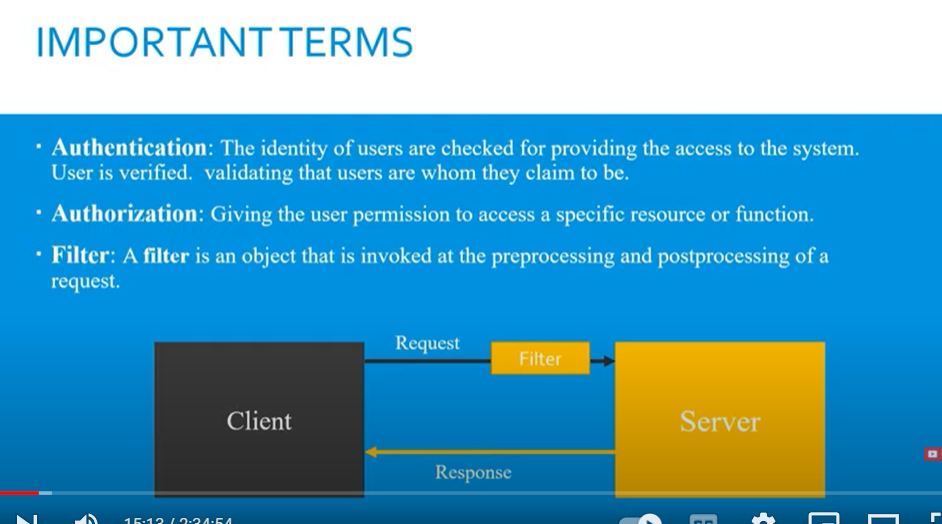
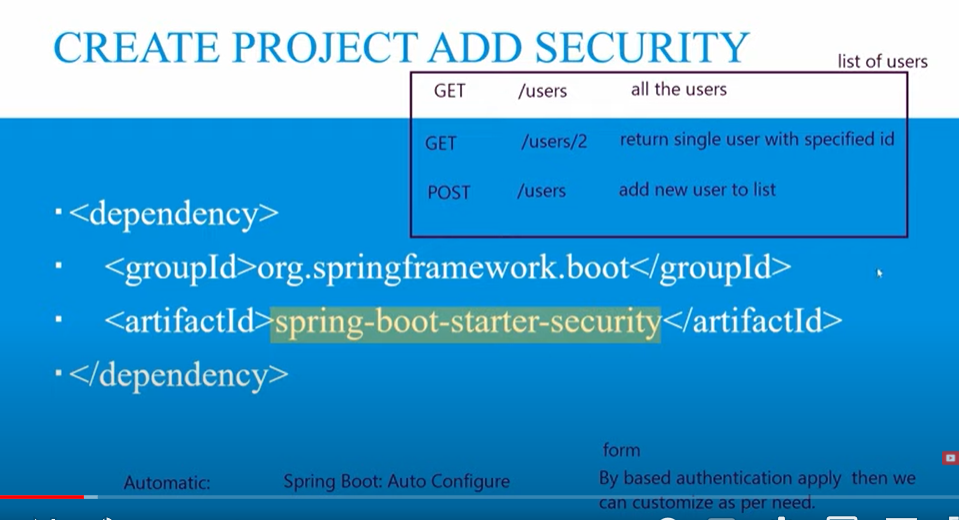
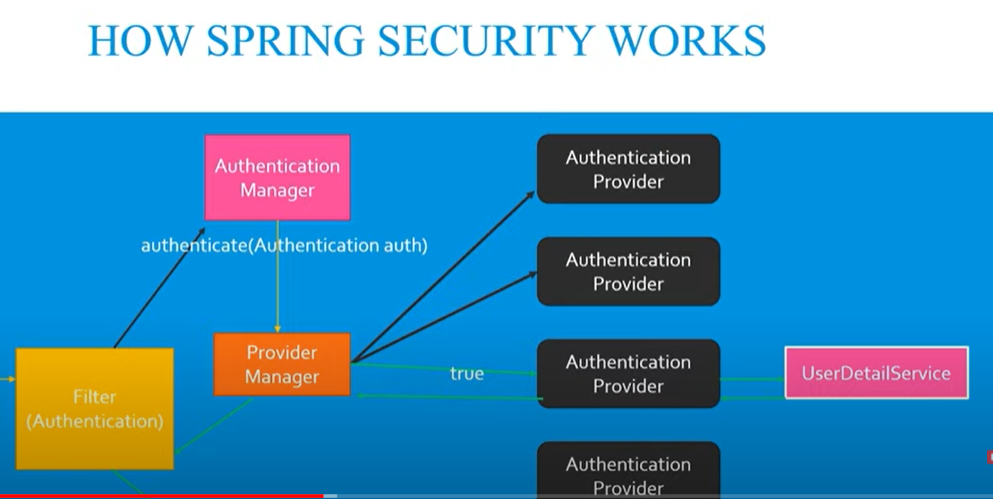
**Spring Security**









Hamare pass koe bhi request ayegi filter sab sai pehlay ayega request ko intercept karay ga filter sirf beech mai request ko pakar sakta hai filter k pass authentication ki koe power nahi hai

Filter 1 authentication ka object banayega ko us k andar basic info rakh day ga or request ko delegate karday ga authentication manager k pass

Authentication manager ka kaam hai ab wo validate karay ga k banda valid hai ya nahi

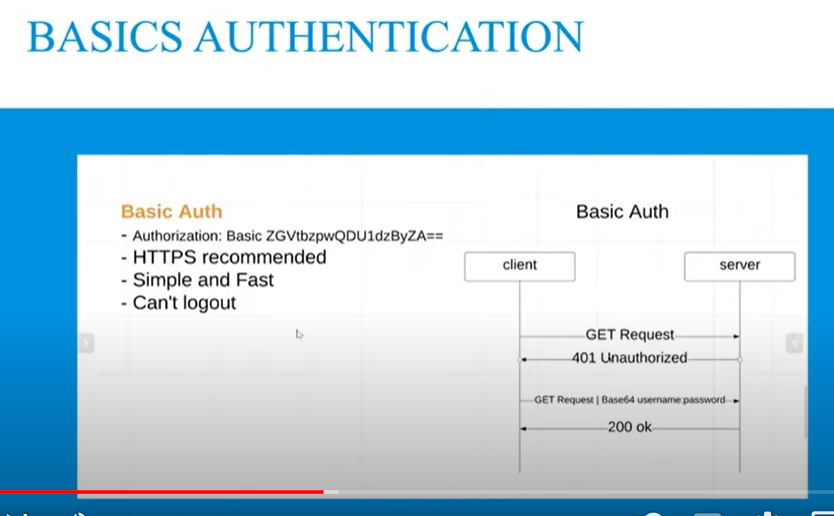
Authentication manager 1 authenticate naam ka function call karta hai or braget mai authentication pass kardeta hai

Authentication manager keh saktay hai yai 1 abstract interface hai is k pass authentication ki implementation nahi hai

Tw authentication ko call karne k liye mere pass alag alag authentication provider hai alag alag authentication provider alag alag kaam kartay hai suppose 1 token authenticate kar raha hai dosra username password authenticate kar raha hai….

Ab yai nahi pata chalay ga konsa authentication provider hamary liye sahi hai tw is k liye provider manager hota hai jo theek hota hamray liye us k against return true kardeta

Open file basicauthenticationfilter in your project



Yai recommended nahi hai q k ab JWT wagera agaye hai

Configuration karne k liye 1 new package banana hoga us k andar MySecurityConfig name sai class banayengay phr 2 annotaion humay laganai hai 1 configuration ki 2nd Enable web Security ki

Phr is class ko extend kardengay WebSecurityConfigorAdpater sai ab hamaray is mai almost sb method miljayengay but humay customize karne k liye un method ko overwrite karna hai

Alt + insert karengay to overwrite method k option ajayengay

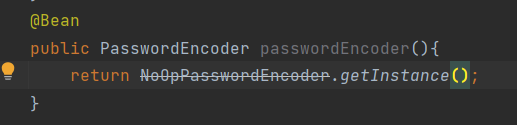
Ab humay by default form base authentication mil rahi hai tw usko change karengay basic authentication mai



Spring security humay by default user deta hai agar humay apna user banana hai tw tw configuration karni paray gi mysecurityconfig file mai in memory or database dono use karsaktay hai



Password encoder ka jab error ayega tw tab waha no password encoder karna parta hai



You can use this method with annotation

Bean annotation is liye use kartay hai takay hum isko kisi or jaga inject karsakay AutoWired kar sakay.

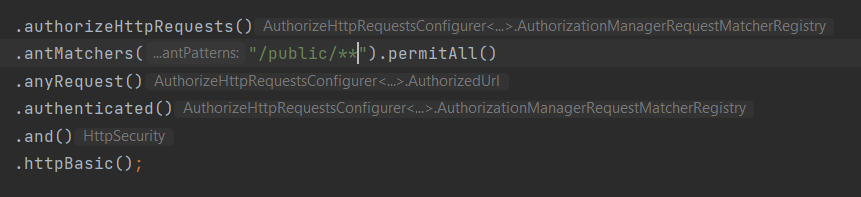
Humain database mai password encode kar k rakhna parta hai takay koe dekh na sakay tw us k liye hamaray pass BycryptPassowordEncoder hota hai

@Bean  
public PasswordEncoder passwordEncoder(){  
 return new BCryptPasswordEncoder(10);  
}

**Ant Matcher**

Ant matcher means by default kisi url ko public karsaktay means agar wo patten match karay ga tw usay public kardo

Ab for example humay multiple pages ko public karna hai tw hum



**CSRF**

Cross site request forgery (CSRF), also known as XSRF, Sea Surf or Session Riding, is **an attack vector that tricks a web browser into executing an unwanted action in an application to which a user is logged in**. A successful CSRF attack can be devastating for both the business and user.

****

**Yai karne k baad hamari form base authentication ki jaga javascript popup base authentication hojyegi.**

**Hum username password apne postman mai bhi use karsaktay**

**Basic authentication k 1 masla hai k humay har baar username password bhejna hoga is cheez ko solve karne k liye hum use karengay jwt.**

**Jwt stateless method par kaam karta hai means server par kuch store nahi karna jo bhi store karna hai token k andar info ajayegi or us token ko hum client par bhej dengay client par token handle hoga**

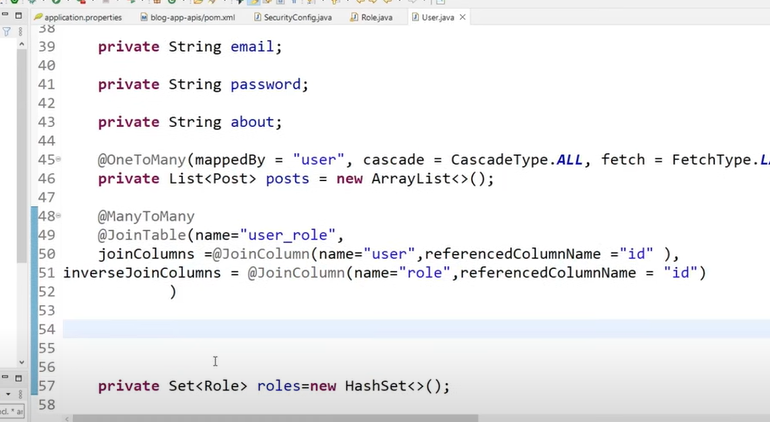
**Jwt ka yai hai k jab jab token bhejay ka api k sath tab tab api ka access allow karna hai warna basic auth sai koe matlab nahi**

**Ab humay info jo hai database mai store karni hai user k table mai**

****

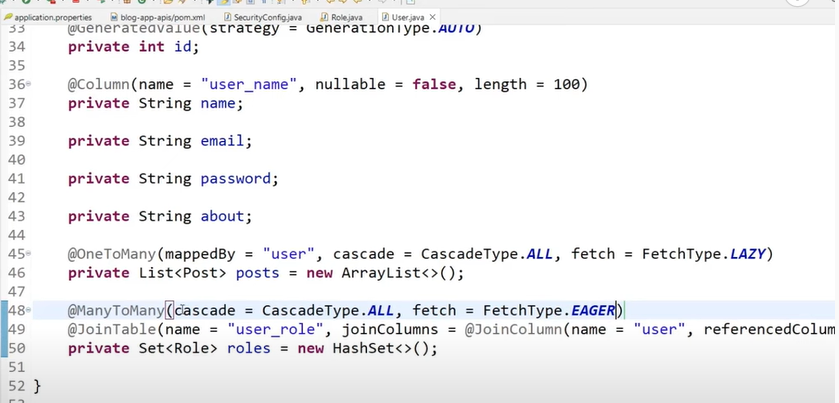
**Note: you can use inmemory database to store this info**

**Ab hum 1 role ki class banayengay entity package mai**

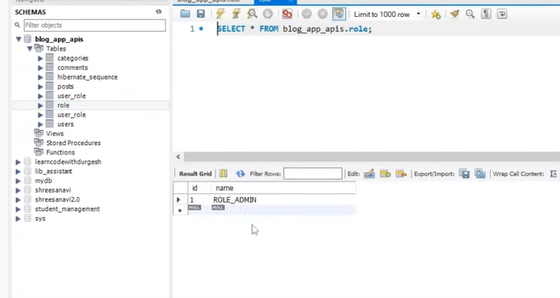
****

**Manytomany function hoga user and role mai.**

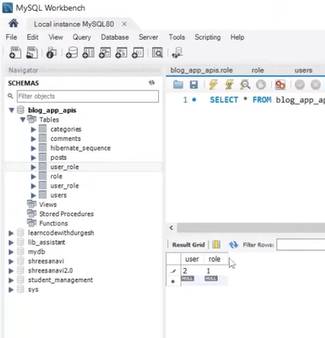
**Acha ab hamaray pass 2 new table create hongi 1 role ki dosri user\_role ki**

****

**Ab hum 1 role add kadengay manually**

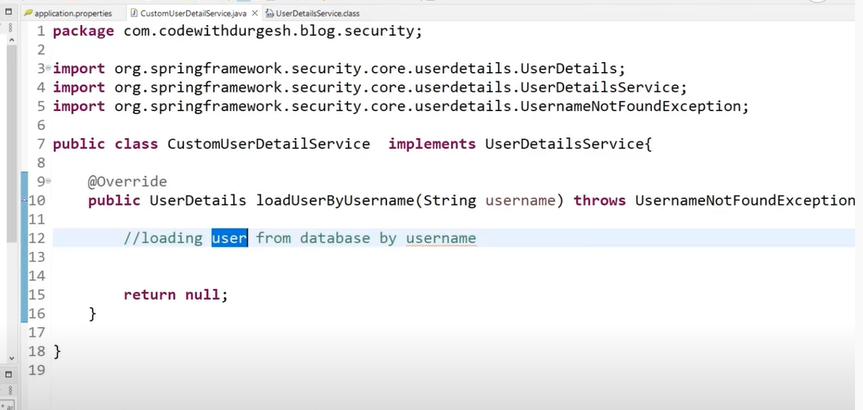
****

**Ab hum role assign karengay matlab 2 walay user ko 1 waala role assign hai**

****

**Database sai authentication k liye hum sai class banayengay**

**Now we will create class CustomUserDetailService and implement this class with UserDetailsService**

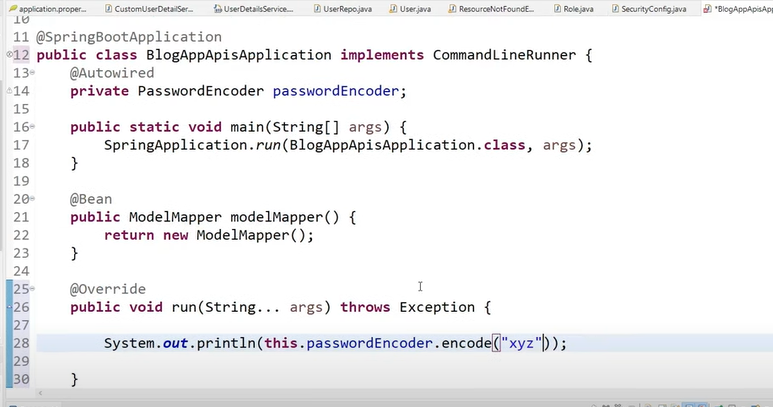
****

**Jab bhi spring security ko need paray username ki tab wo yai method call karday ga.**

**Ab hum userrepo mai 1 method banakar usi method ko use karengay UserDetailService mai**

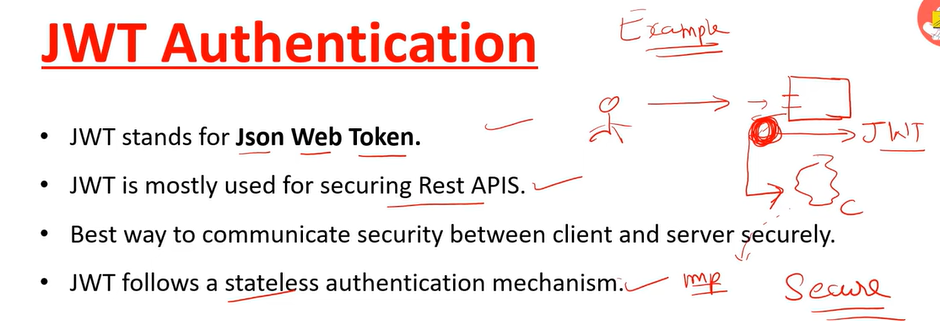
****

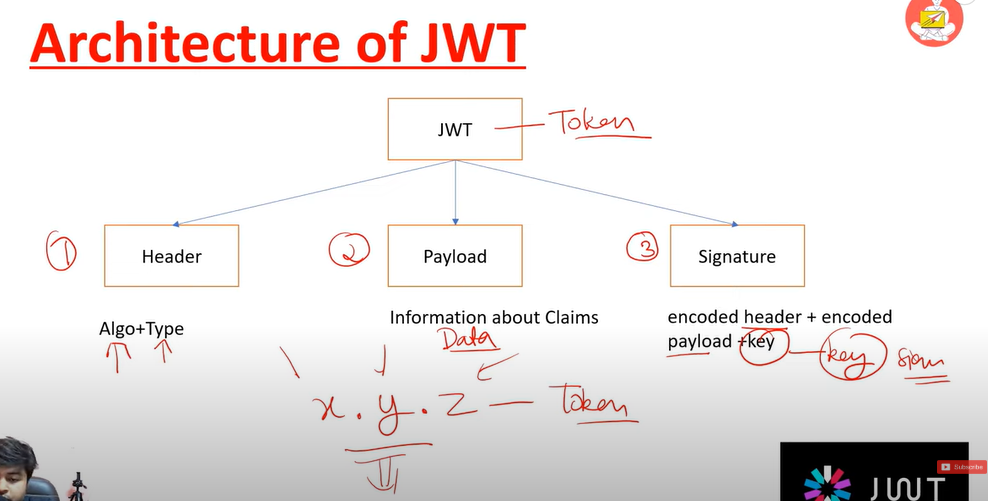
**Password endoder ko use karne k liye hum CommandLineRunner ko implement karengay us k 1 method ko override karnegay run**

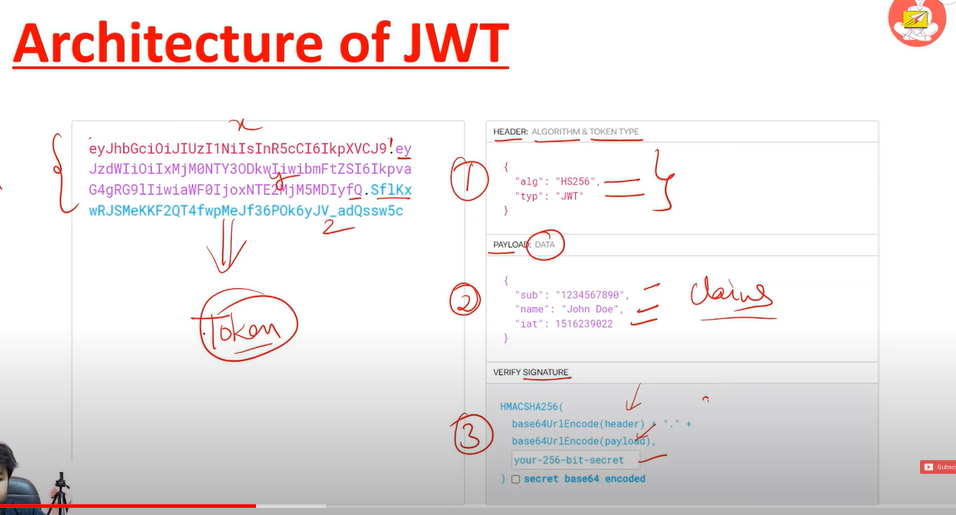
****

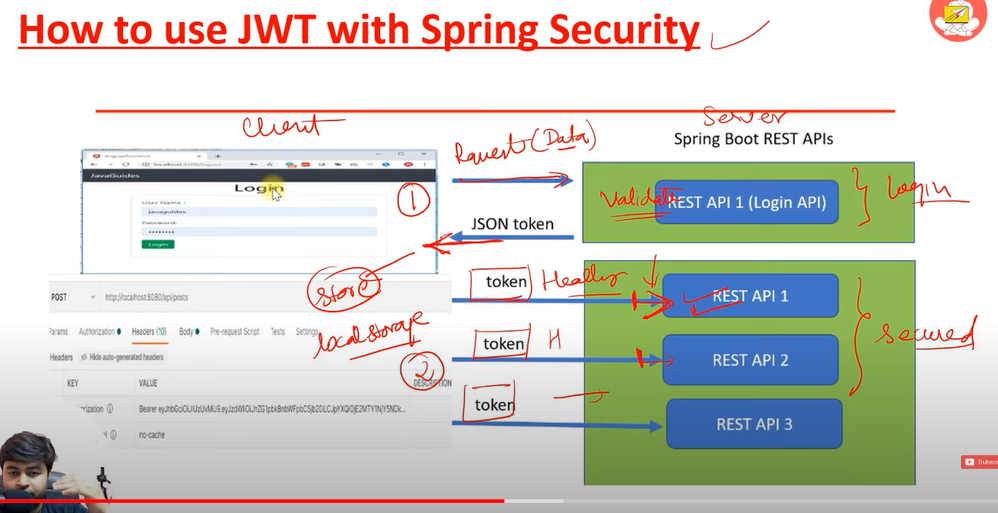
**Ab basic authentication ko hum jwt authenticaton mai karengay 1 step agay ab humay 1 token generate karna hoga usi mai saara kaam karna hoga.**

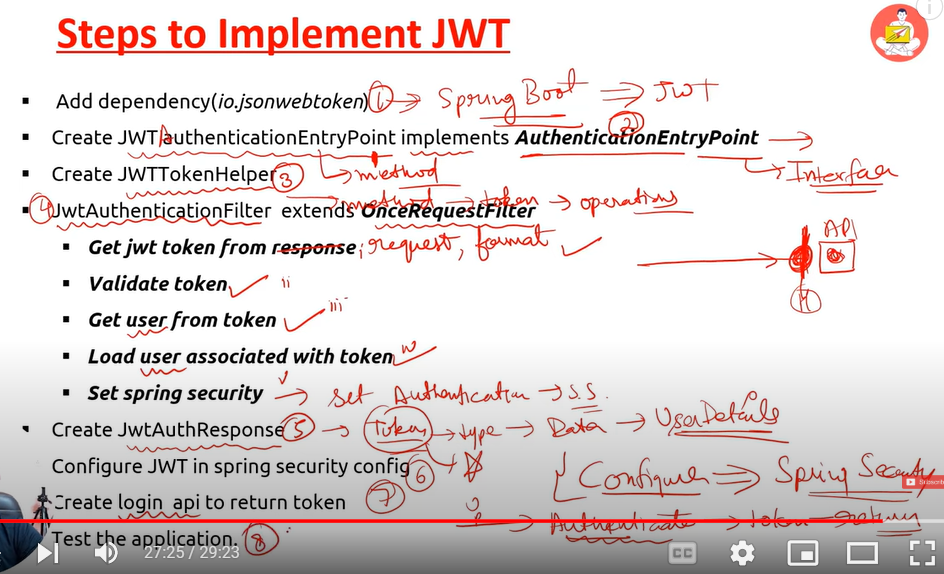
# Crisp and Clear explanation of JWT authentication

****

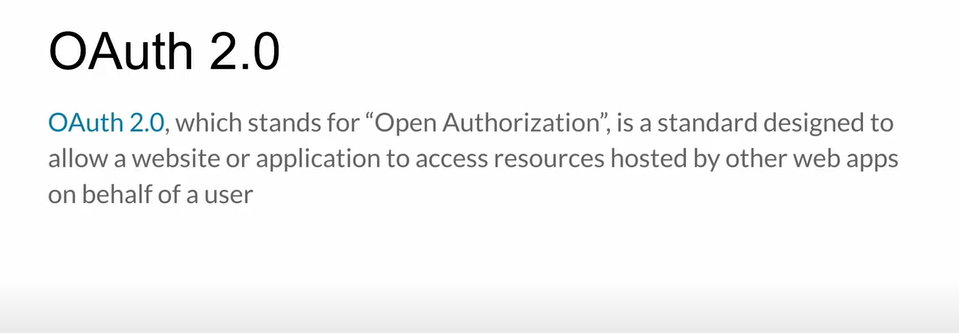
****

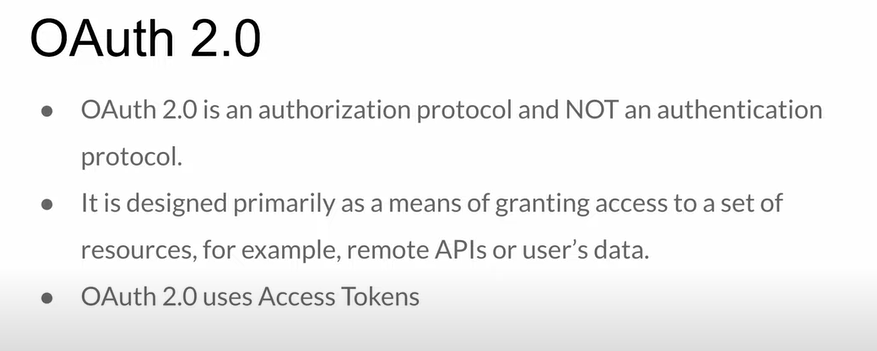
****

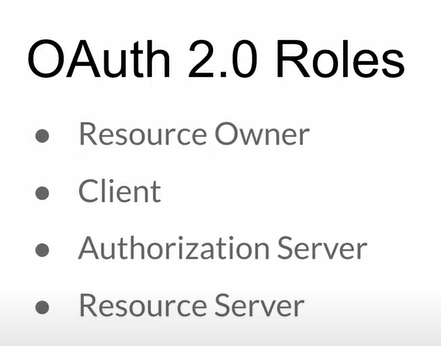
****

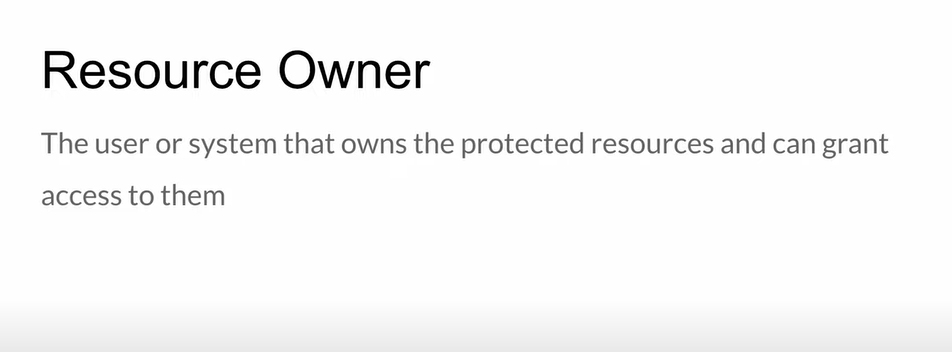
****

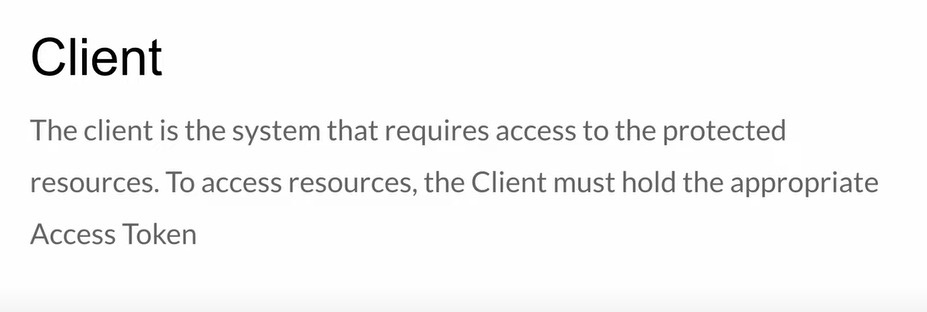
# OAuth 2.0 Implementation with Spring Security

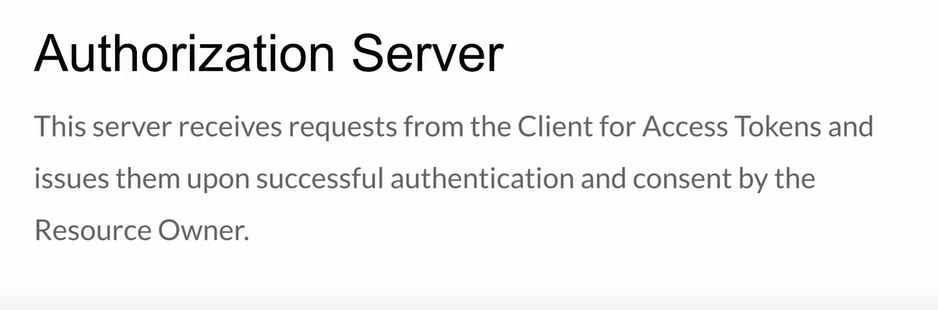
****

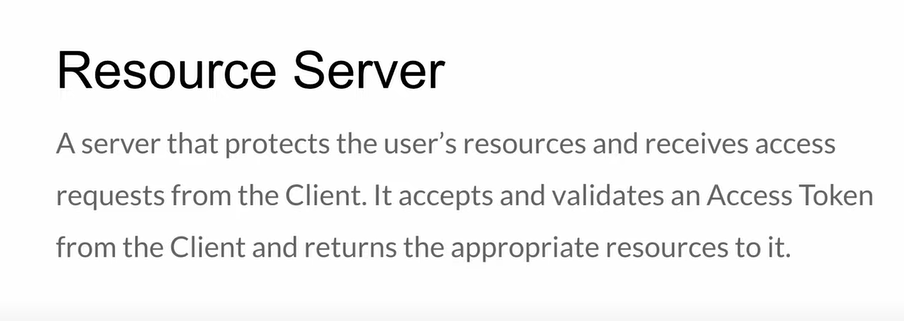
****

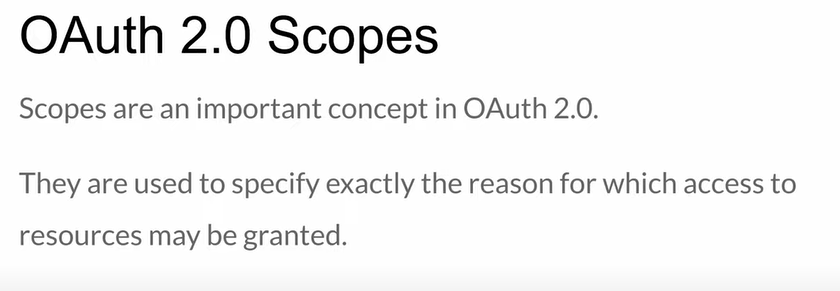
****

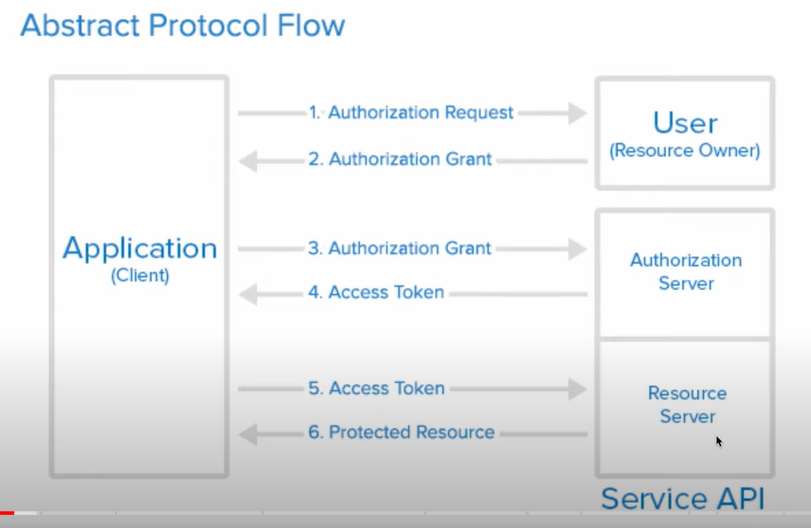
****

****

****

****

****

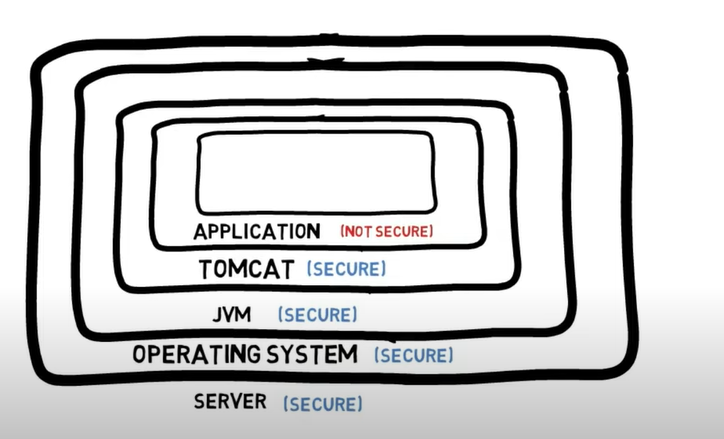
****

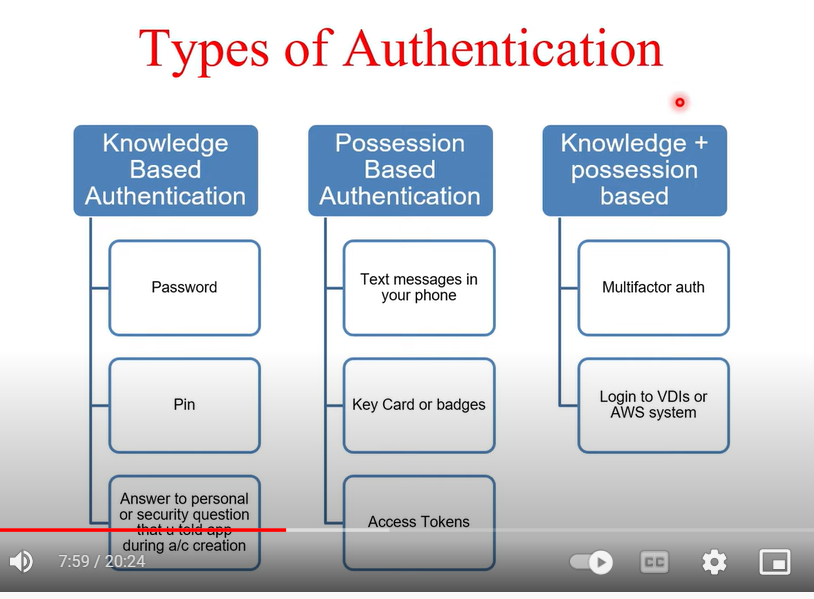
**’**

**https://www.javainuse.com/misc/oauth-interview-questions**

# Spring Security in Spring boot Interview Questions and Answers

****

****

****

****

****