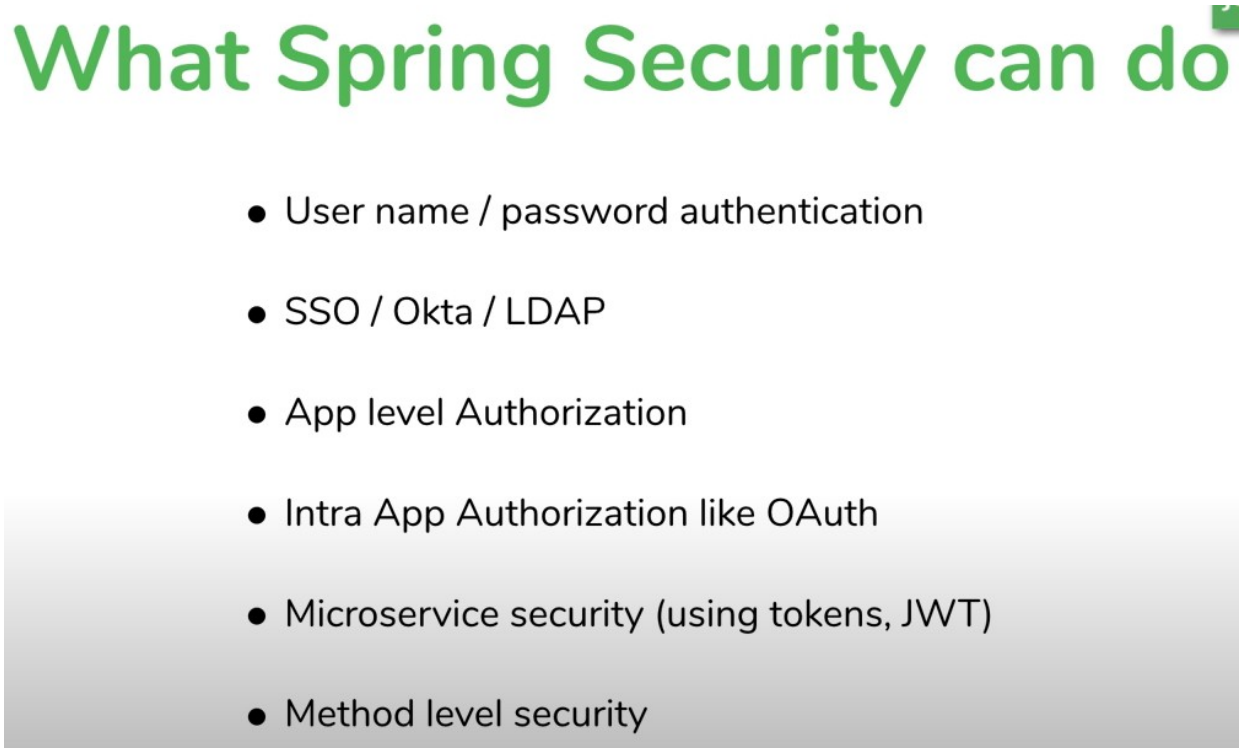
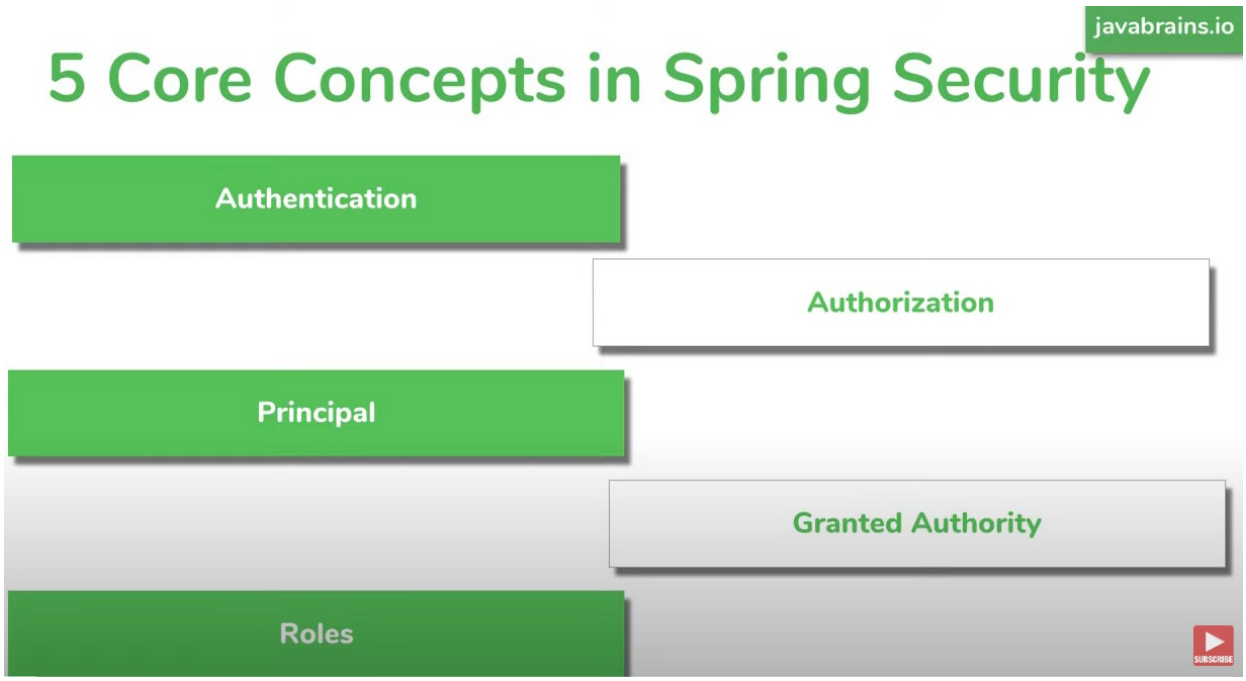
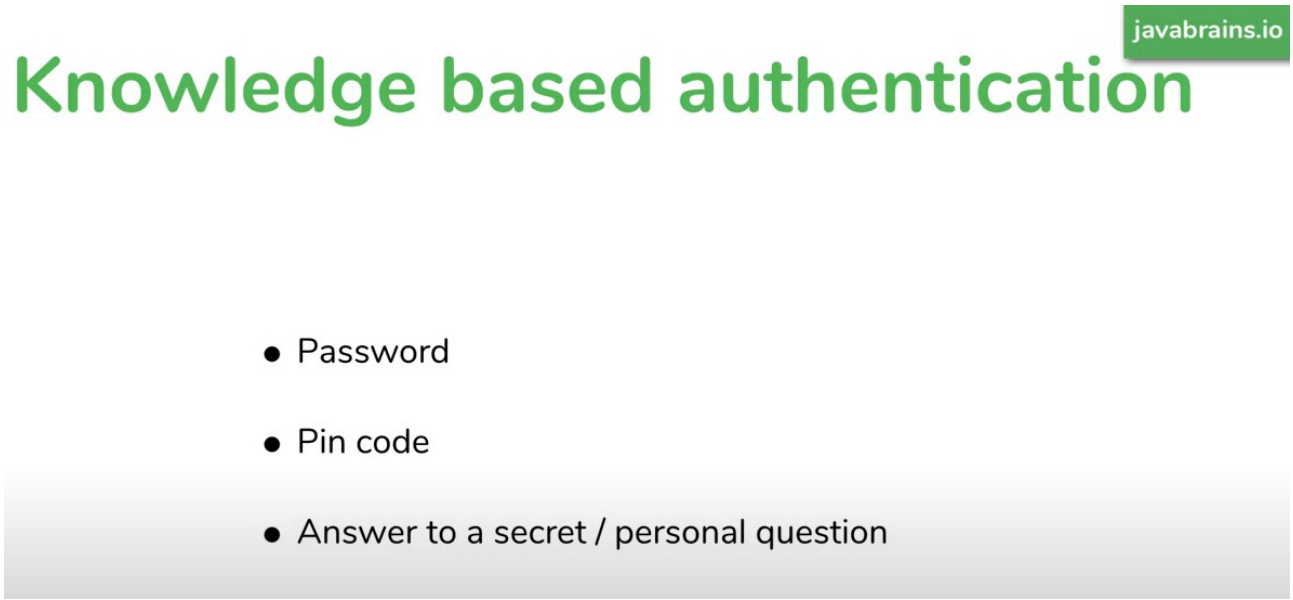
Spring security

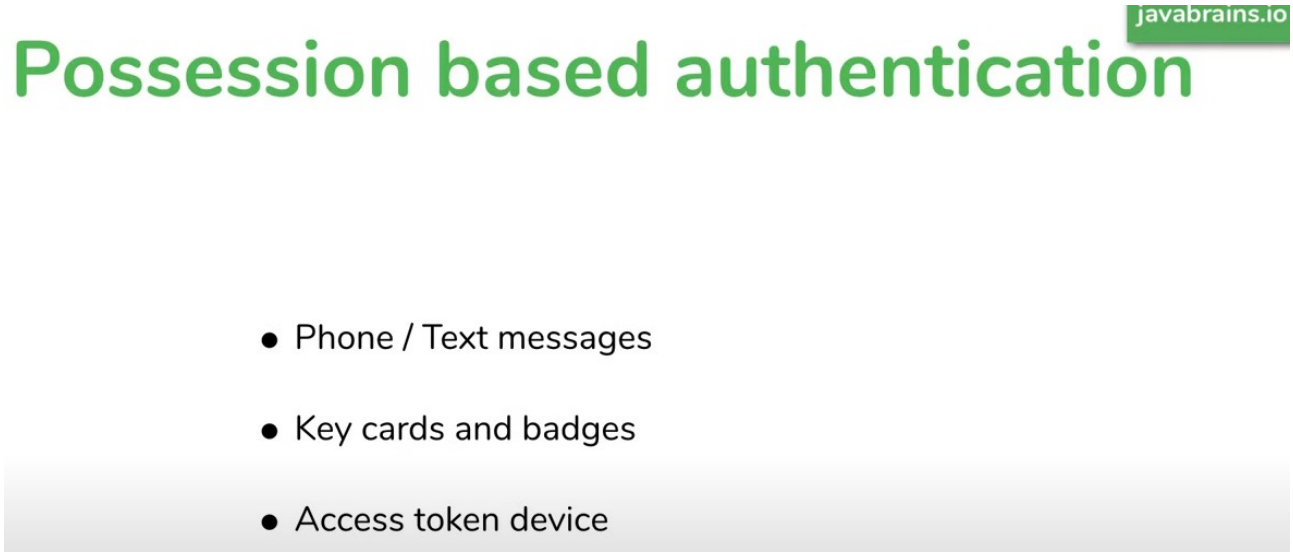
Source: https://www.youtube.com/playlist?list=PLqq-6Pq4lTTYTEooakHchTGglSvkZAjnE





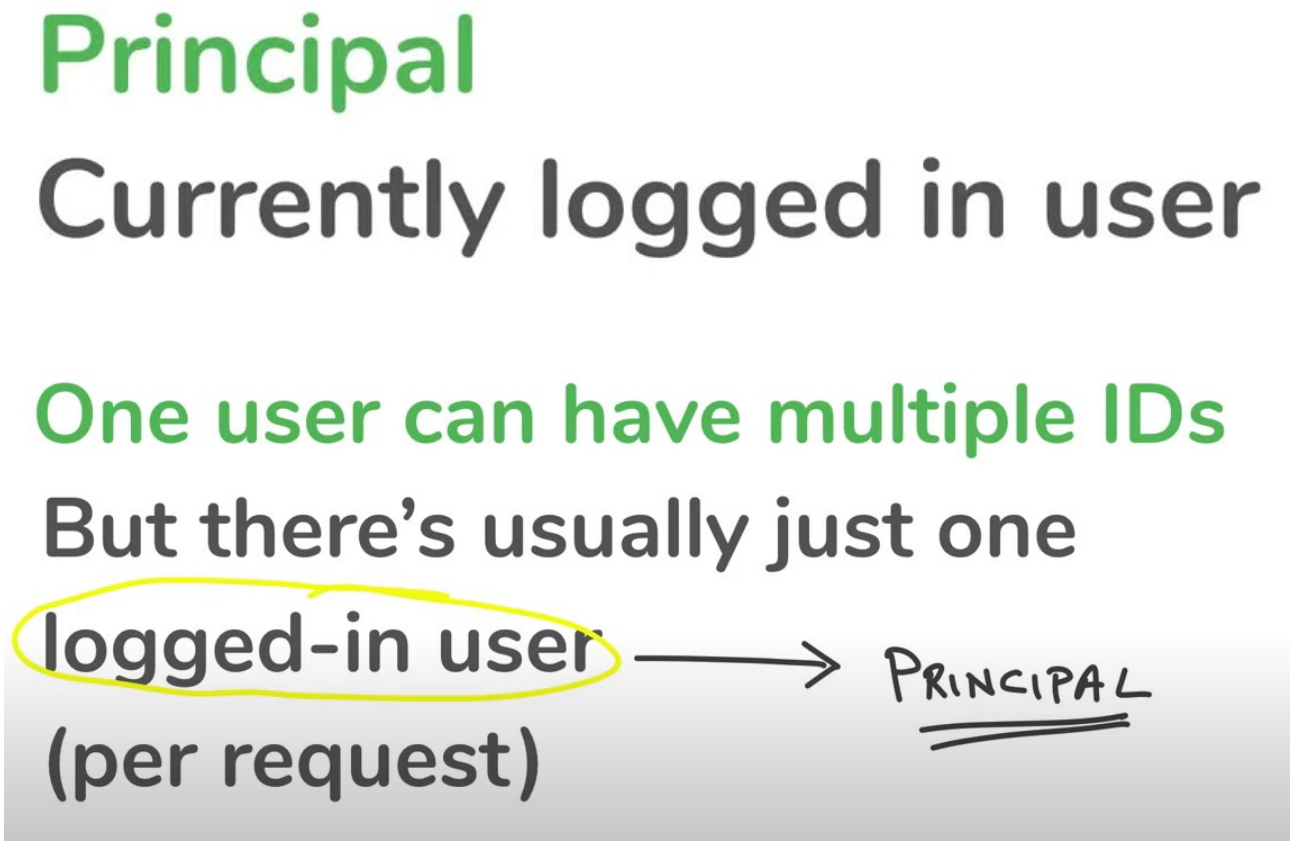
Authen = Who you are? Prove that?



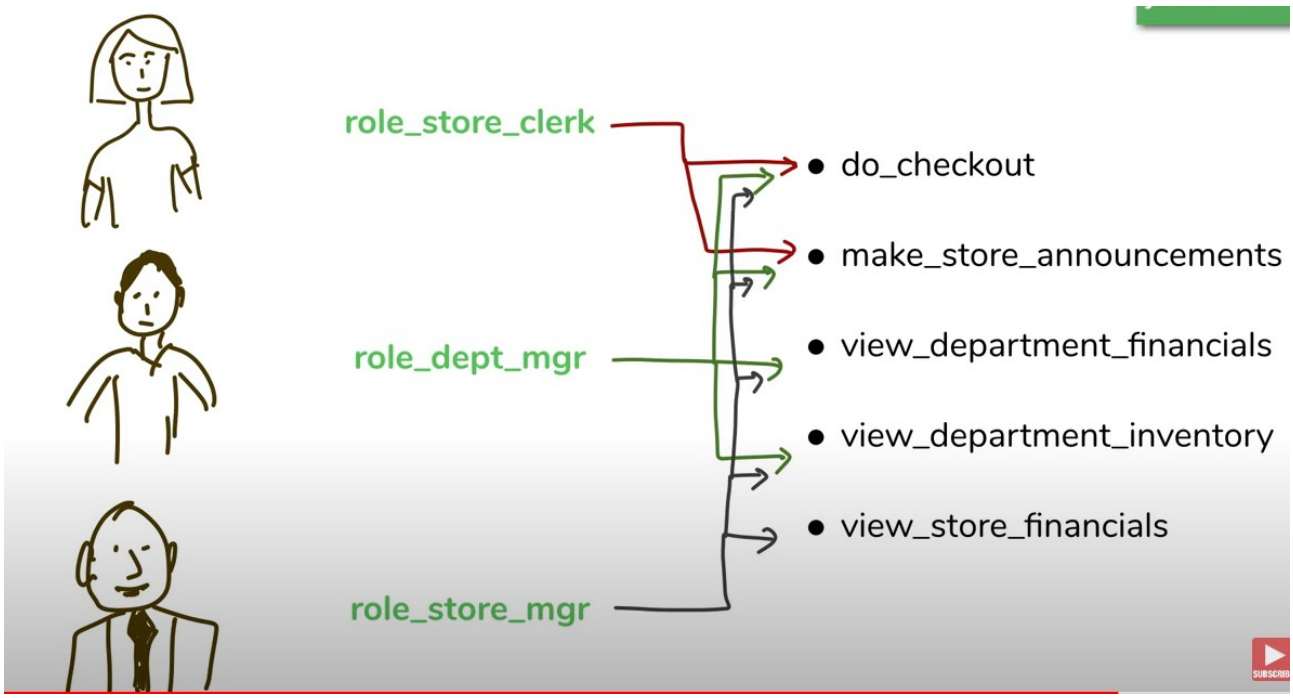


Author = What user can do?

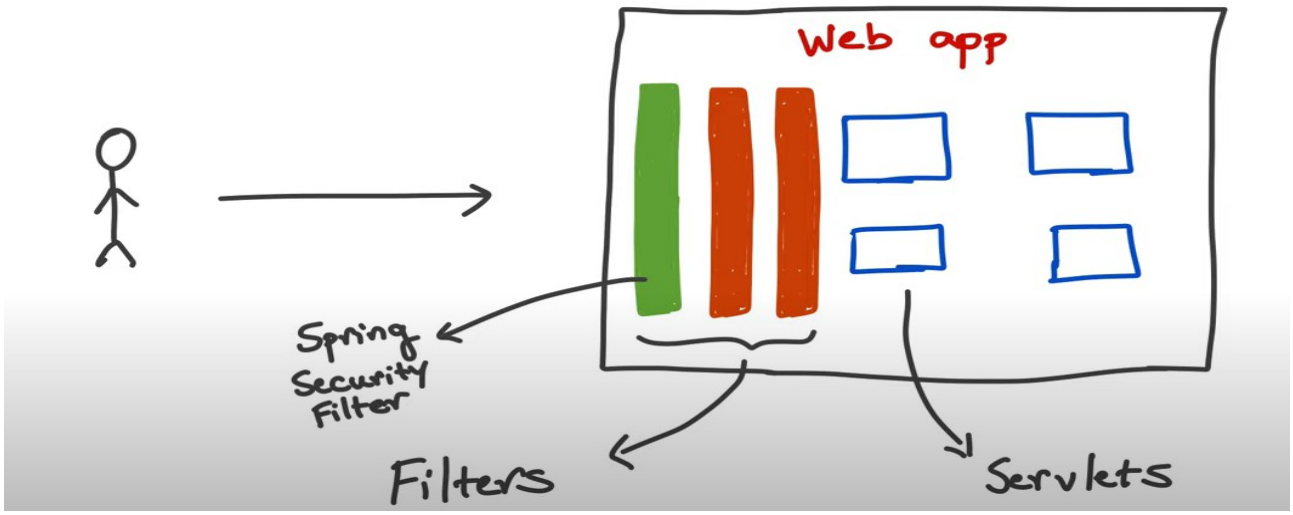
Principal = login once



Granted Authorities and Roles


role_store_clerk
role_dept_mgr
role_store_mgr
do_checkout
make_store_announcements
view_department_financials
view_department_inventory
view_store_financials

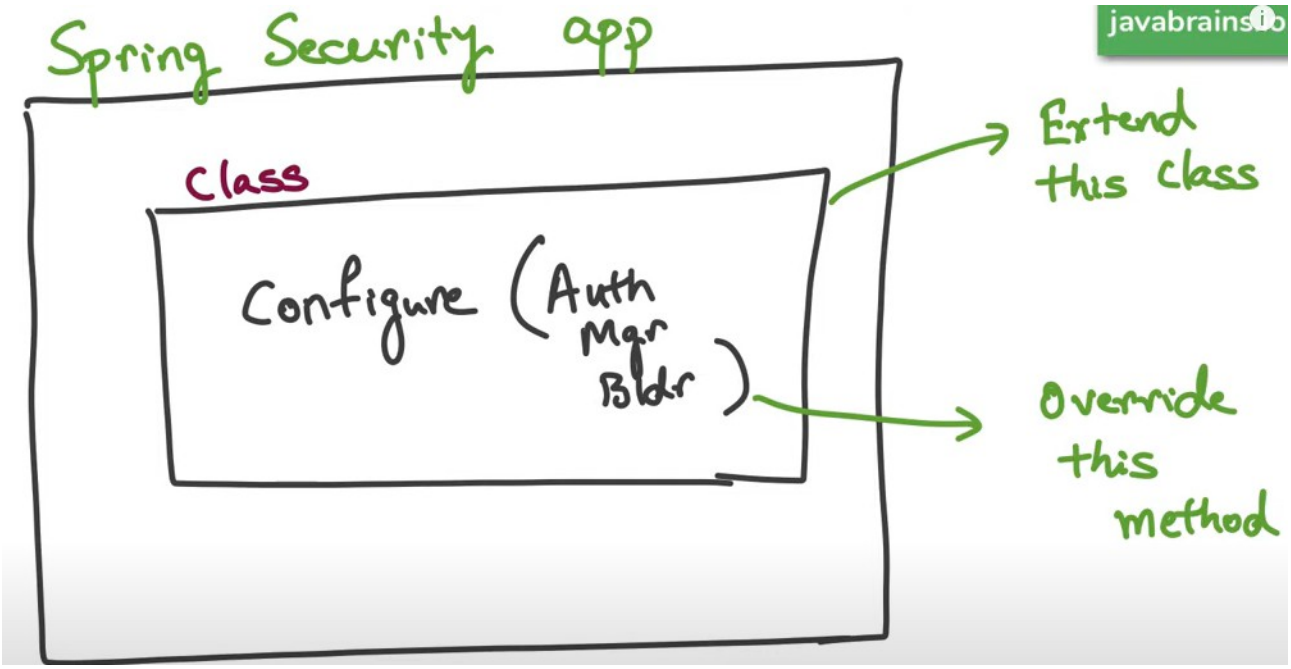
Filters


Web app
Spring Security Filter
Filters
Servlets

Spring Security default behavior

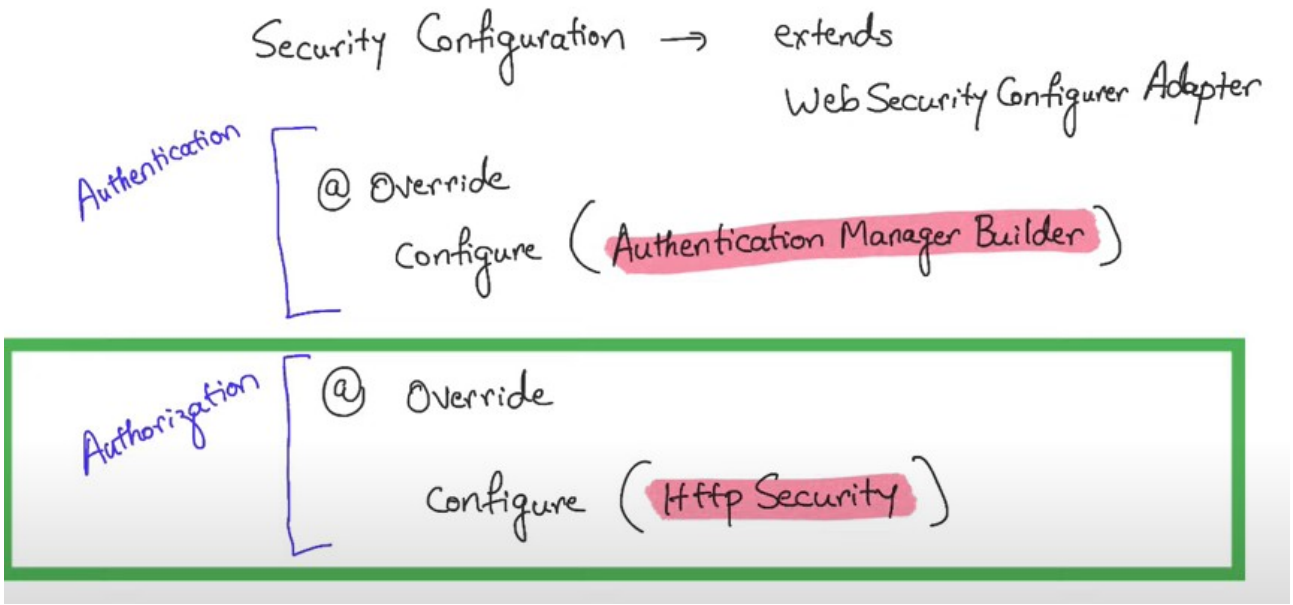
- Adds mandatory authentication for URLs
- Adds login form
- Handles login error
- Creates a user and sets a default password

Authentication Manager < Authentication Manager Build

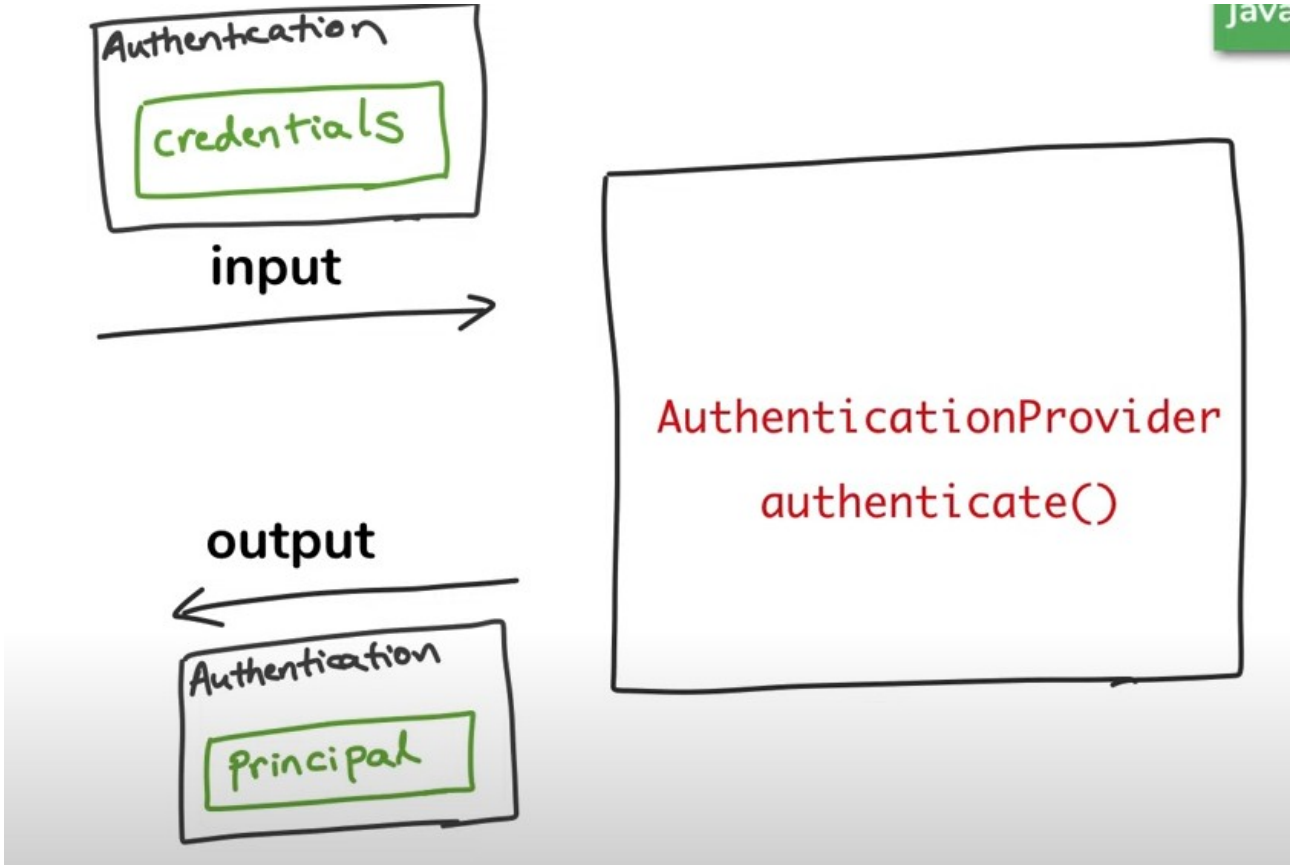


Authorization / HttpSecurity

API	Roles allows to access it
/	All (unauthenticated)
/user	USER and ADMIN roles
/admin	ADMIN role



Authentication Provider



org.springframework.security.authentication

Interface AuthenticationProvider

```
public interface AuthenticationProvider
```

Indicates a class can process a specific `Authentication` implementation.

Method Summary

All Methods	Instance Methods	Abstract Methods

Modifier and Type	Method and Description
`Authentication`	**`authenticate(Authentication`** `authentication)` Performs authentication with the same contract as **`AuthenticationManager.authenticate(Authentication)`**.

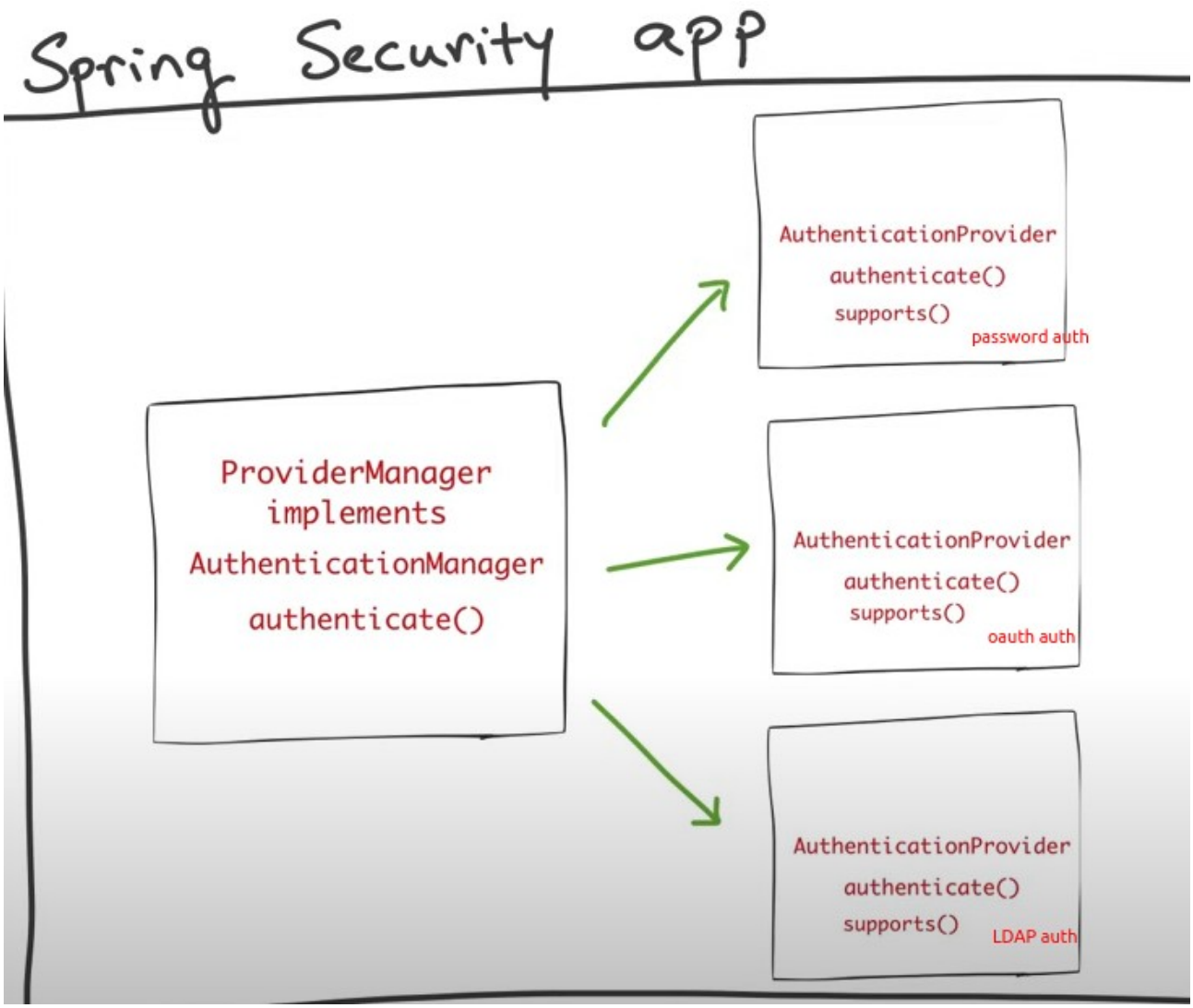
org.springframework.security.core

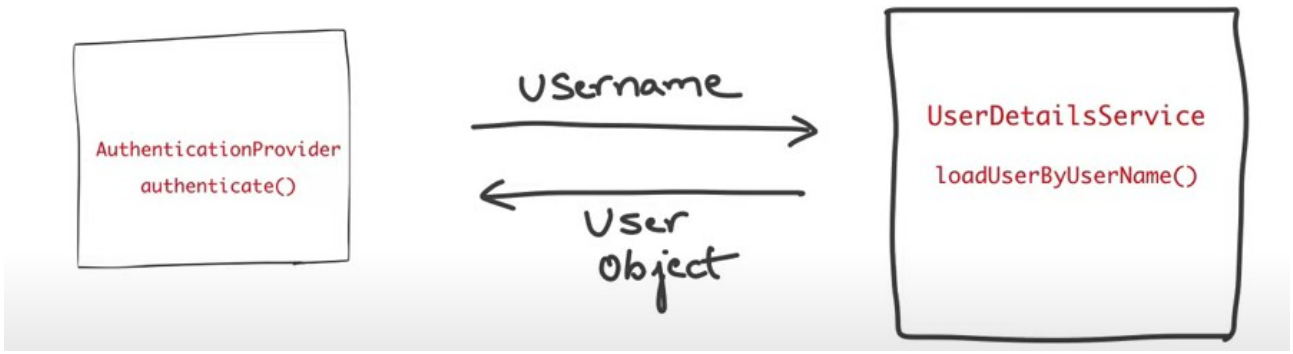
Interface Authentication

Method Summary

All Methods | **Instance Methods** | **Abstract Methods**

Modifier and Type	Method and Description
`java.util.Collection<? extends GrantedAuthority>`	**`getAuthorities()`** Set by an `AuthenticationManager` to indicate the authorities that the principal has been granted.
`java.lang.Object`	**`getCredentials()`** The credentials that prove the principal is correct.
`java.lang.Object`	**`getDetails()`** Stores additional details about the authentication request.
`java.lang.Object`	**`getPrincipal()`** The identity of the principal being authenticated.
`boolean`	**`isAuthenticated()`** Used to indicate to `AbstractSecurityInterceptor` whether it should present the authentication token to the `AuthenticationManager`.
`void`	**`setAuthenticated`**`(boolean isAuthenticated)` See **`isAuthenticated()`** for a full description.





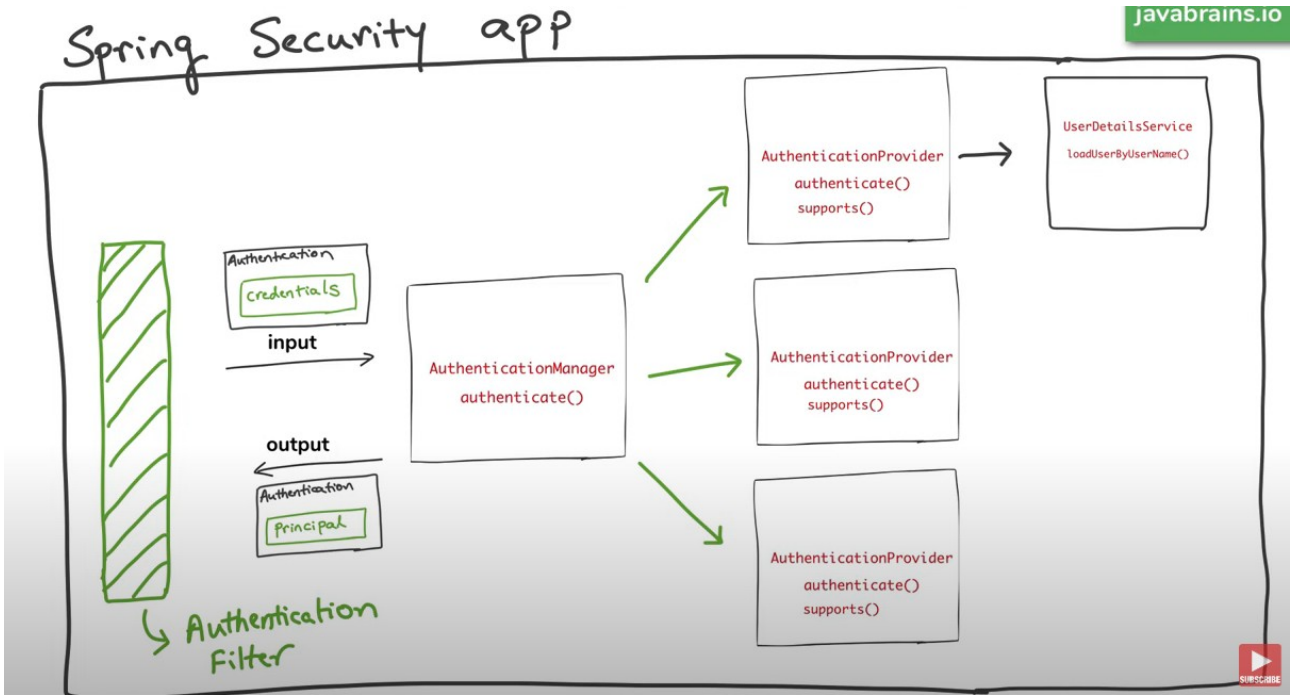
org.springframework.security.core.userdetails

Interface UserDetails

Method Summary

All Methods | Instance Methods | Abstract Methods

Modifier and Type	Method and Description
`java.util.Collection<? extends GrantedAuthority>`	**`getAuthorities()`** Returns the authorities granted to the user.
`java.lang.String`	**`getPassword()`** Returns the password used to authenticate the user.
`java.lang.String`	**`getUsername()`** Returns the username used to authenticate the user.
`boolean`	**`isAccountNonExpired()`** Indicates whether the user's account has expired.
`boolean`	**`isAccountNonLocked()`** Indicates whether the user is locked or unlocked.
`boolean`	**`isCredentialsNonExpired()`** Indicates whether the user's credentials (password) has expired.
`boolean`	**`isEnabled()`** Indicates whether the user is enabled or disabled.

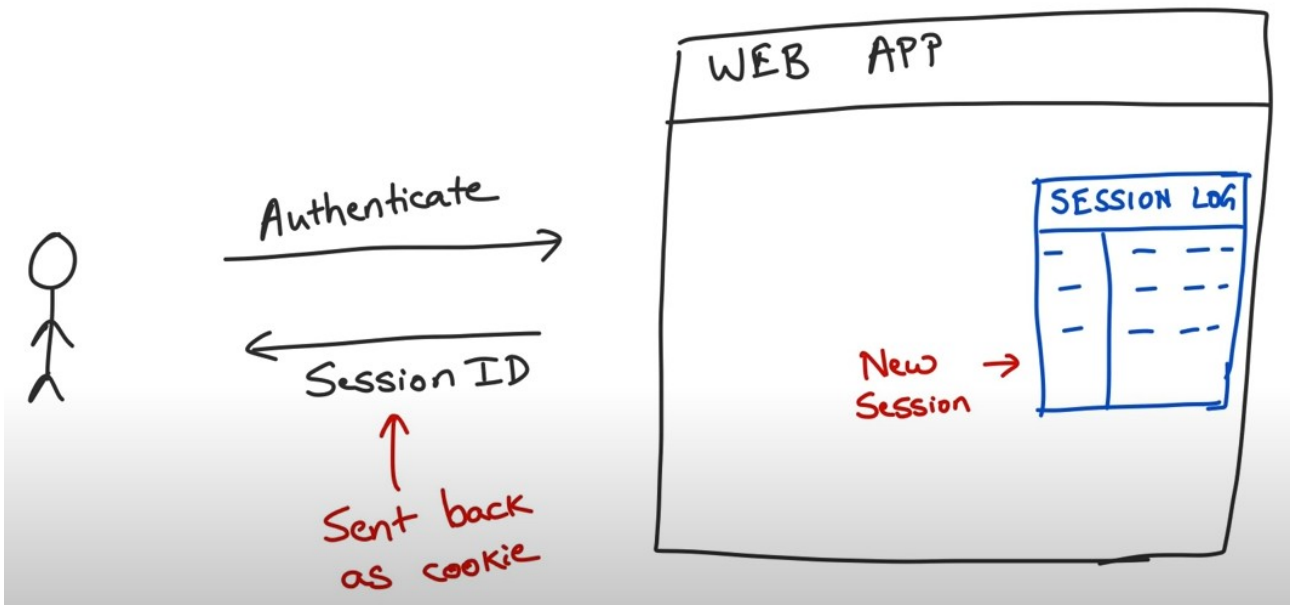


ThreadLocal (Authentication) => Session

Authorization strategies

Using tokens

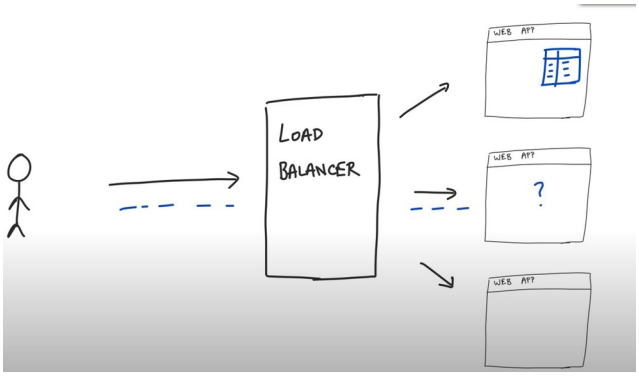
- Session token
- JSON web token

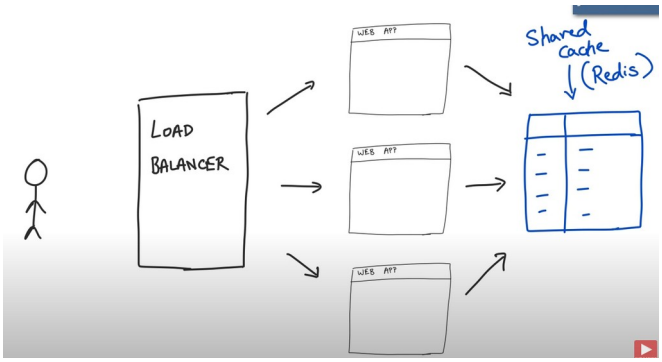


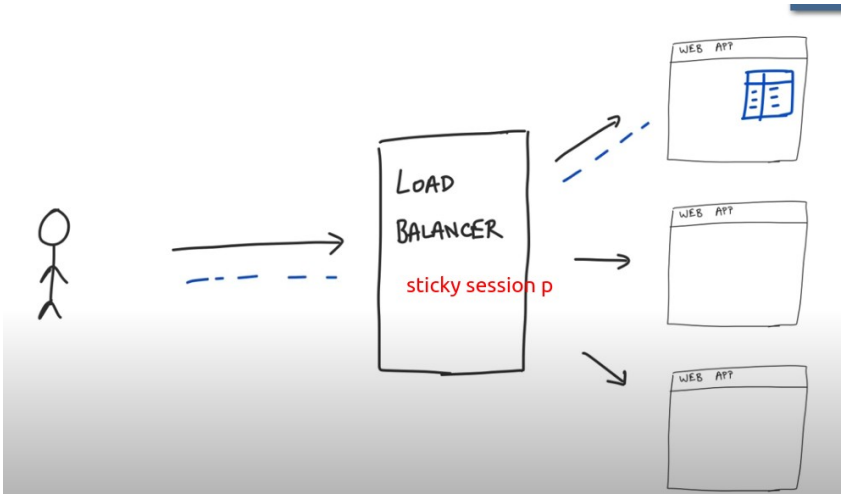
Session ID + Cookies

Most popular mechanism for authorization

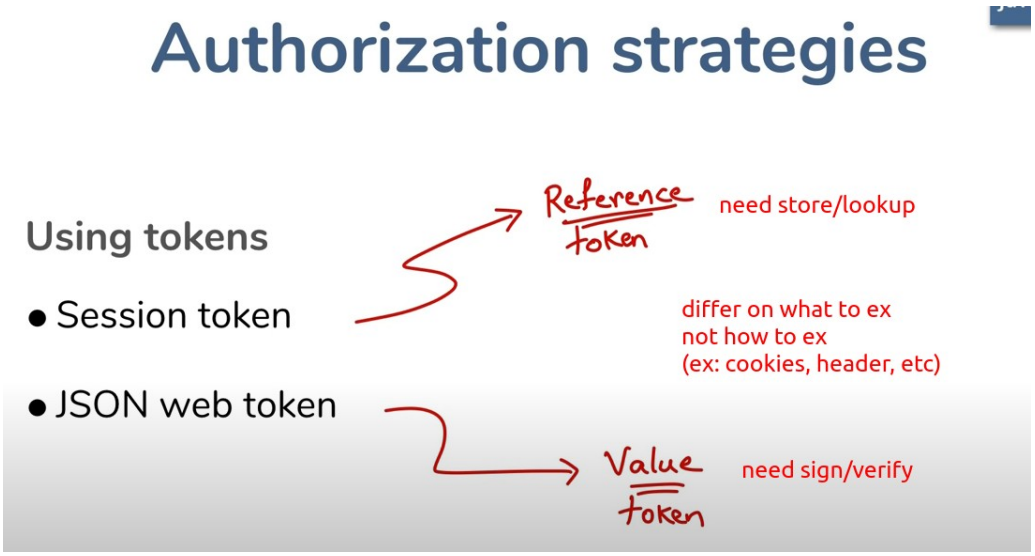
Session ID + Cookies problem ?







JWT



Algorithm HS256

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.ey
JzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6Ikpva
G4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKx
wRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c
```

To authentication

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

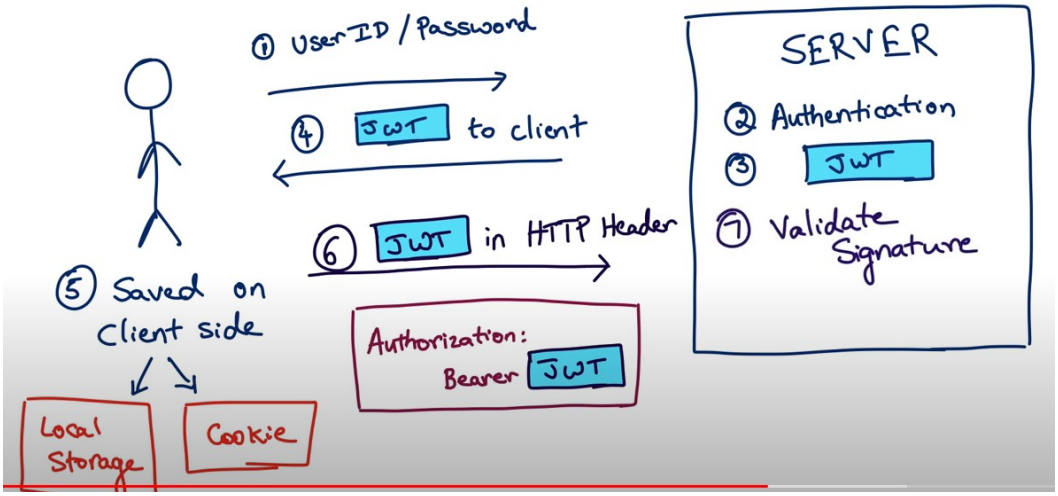
```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

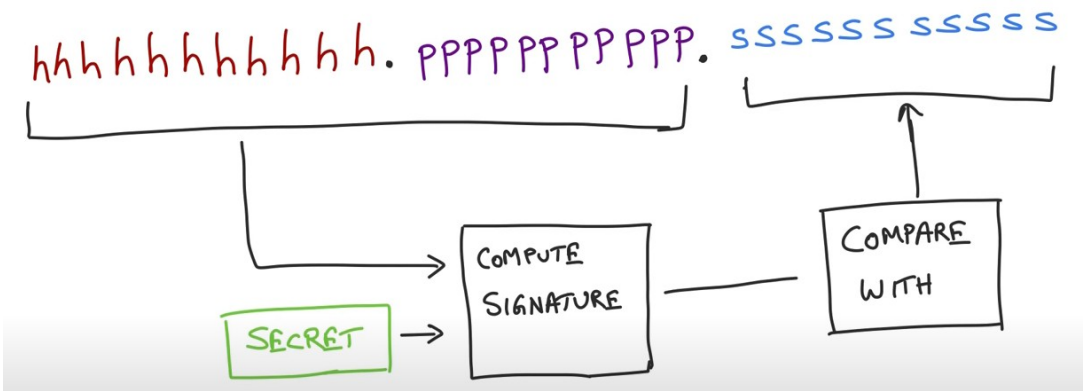
PAYLOAD: DATA

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "iat": 1516239022
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) secret base64 encoded
```





JWT Issues

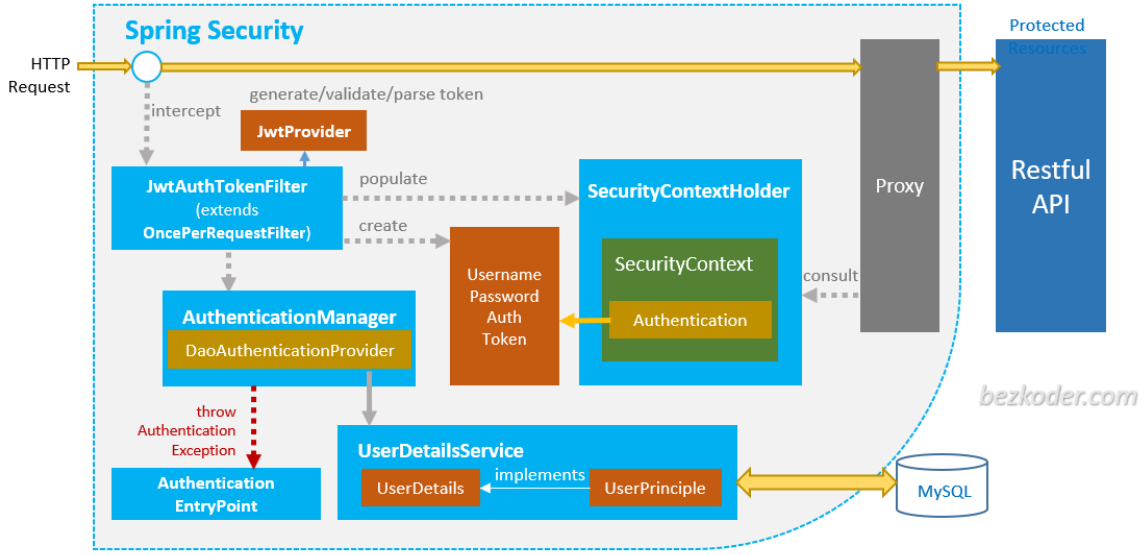
- How secure? No confidential / sensitive information put in a JWT

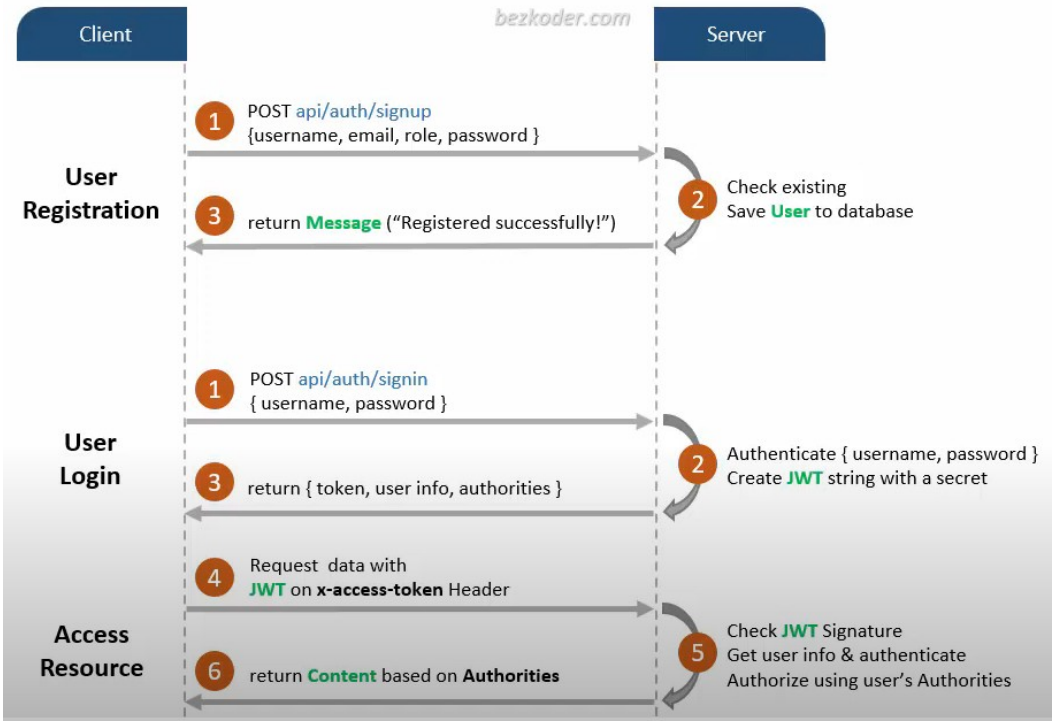
- What if someone steals JWT and use it? => how to transmit JWT: https, oauth

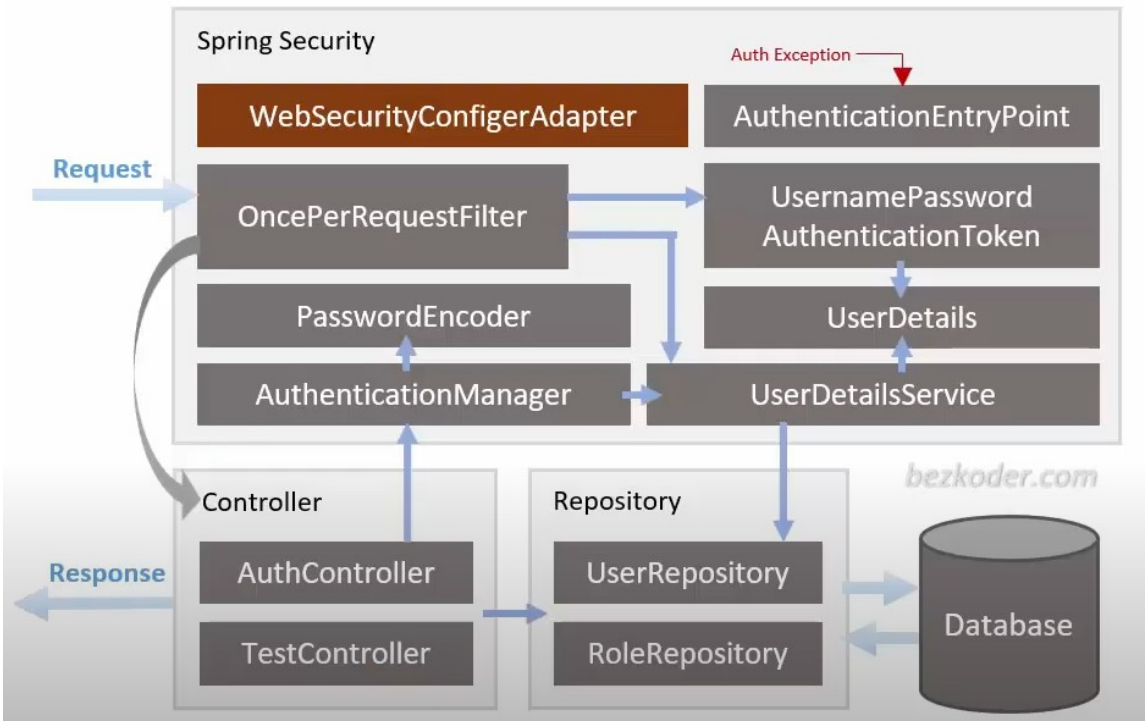
- How to invalidate a JWT? => blacklist???

JWT Coding

https://www.bezkoder.com/spring-boot-jwt-mysql-spring-security-architecture/







Ref:

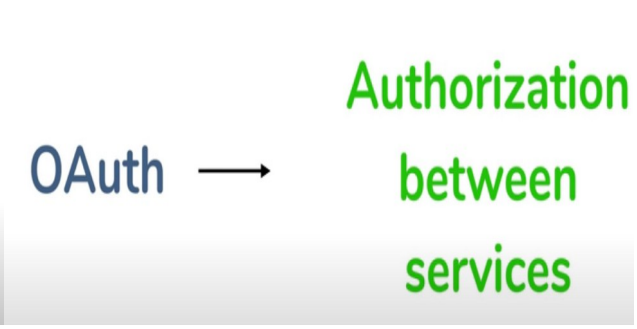
https://www.marcobehler.com/guides/spring-security

https://levunguyen.com/laptrinhspring/2020/04/21/su-dung-spring-security-trong-spring/

https://mehmetozanguven.github.io/spring/2020/12/29/spring-security-1-basic-concepts.html

OAuth


~~Authentication?~~
OAuth
Authorization


OAuth
Authorization
between
services

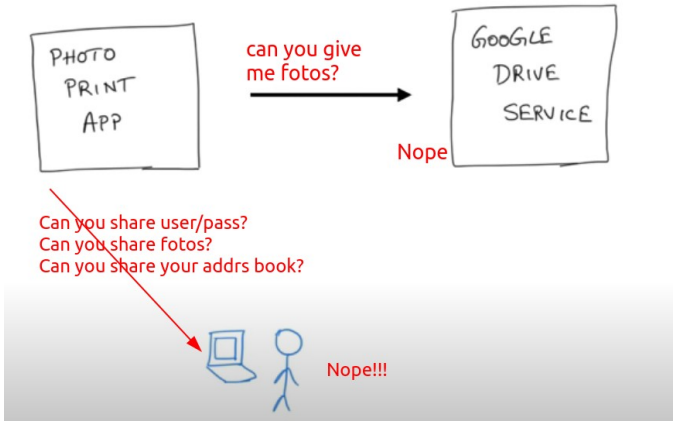
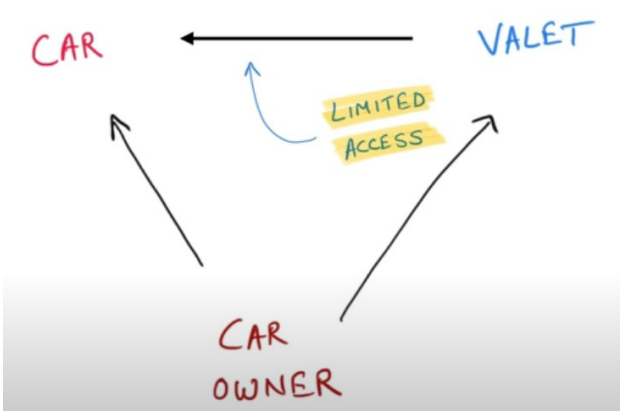
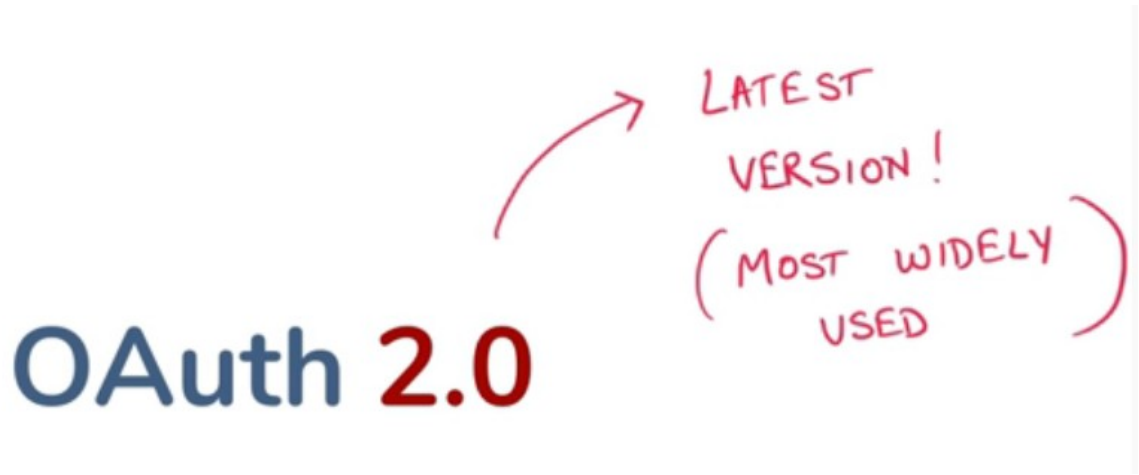
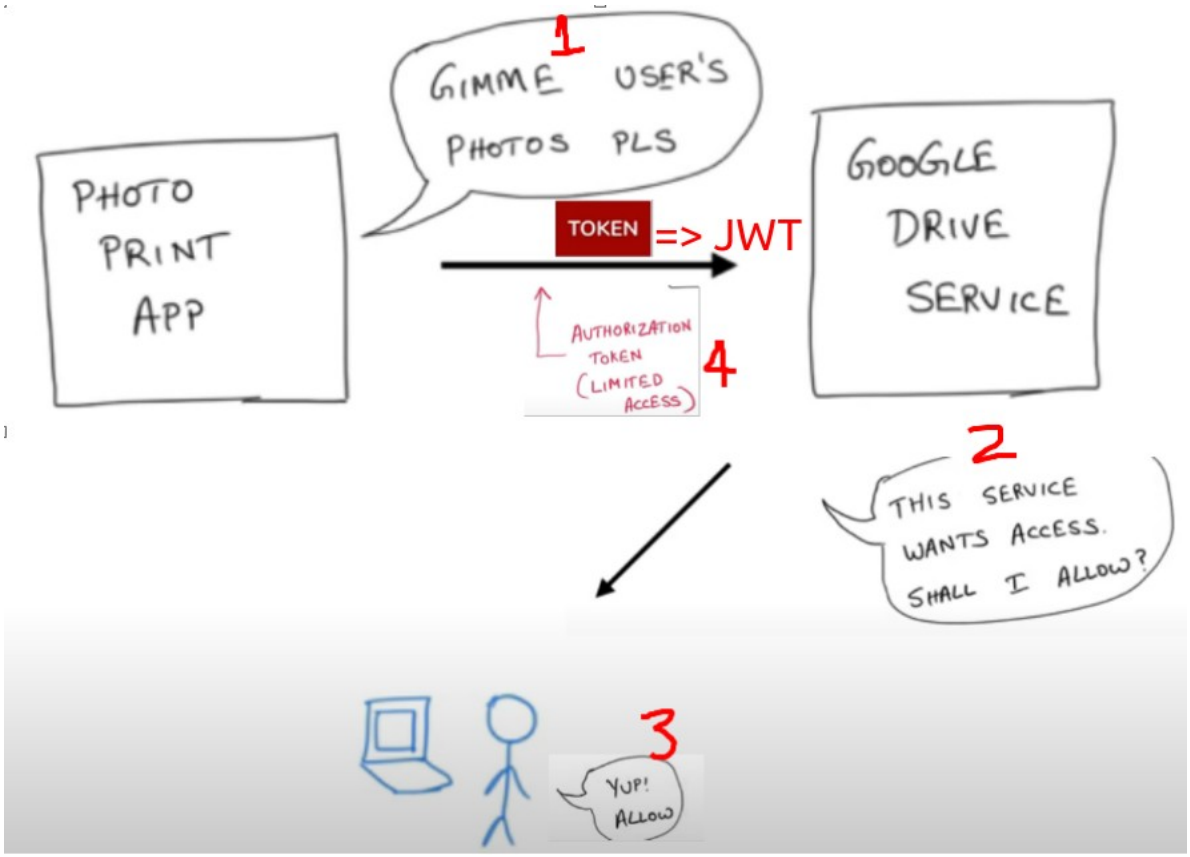


PHOTO
PRINT
APP
can you give
me fotos?
GOOGLE
DRIVE
SERVICE
Nope
Can you share user/pass?
Can you share fotos?
Can you share your addrs book?
Nope!!!

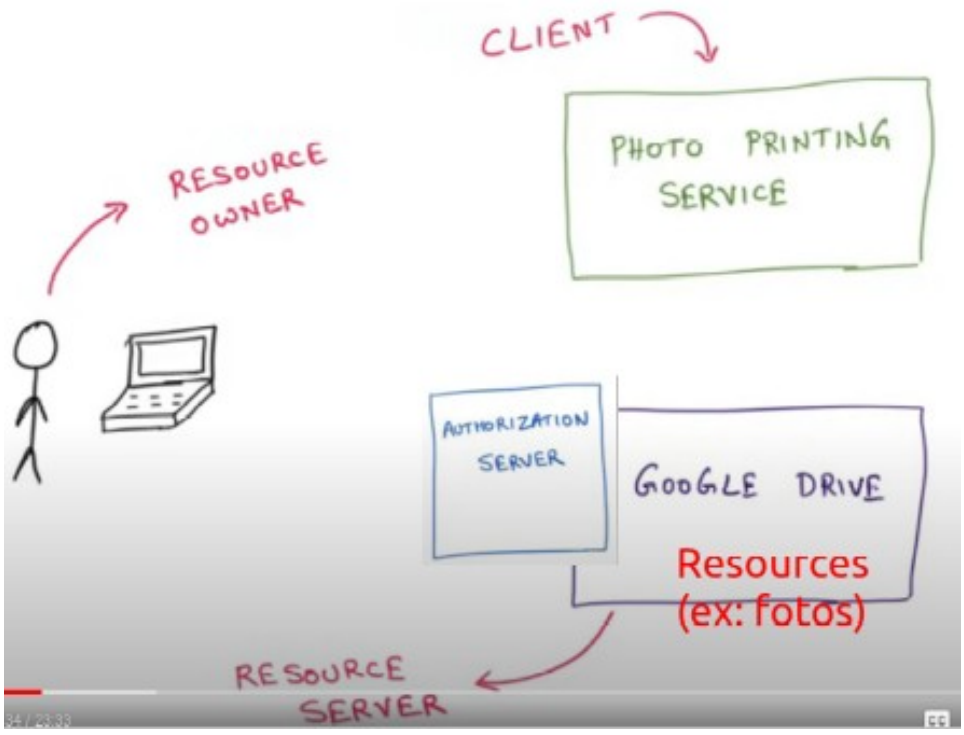

CAR
VALET
LIMITED
ACCESS
CAR
OWNER


LATEST
VERSION!
(MOST WIDELY
USED)
OAuth 2.0



OAuth Terms

- Resource, Resource Owner, Auhentication Server/Resource Server, Client App



OAuth for authen (signin from FB, GH, GG)