DelegatingFilterProxy

/admin/*

Filter

/api/*

Filter

UsernamePasswordAuthenticationFilter

DelegatingFilterProxy

Authentication attemptAuthentication(

UsernamePasswordAuthenticationToken

:AuthenticationManager

this.getAuthenticationManager().authenticate(authRequest)

DaoAuthenticationProvider

LdapAuthenticationProvider

InMemoryUserDetailsManager

① ⊳ UserDetails
⊕ ⊳ getAuthorities(): Collection<? extends
⊕ ⊳ getPassword(): String
⊕ ⊳ getUsername(): String
⊕ ⊳ isAccountNonExpired(): boolean
⊕ ⊳ isAccountNonLocked(): boolean
⊕ ⊳ isCredentialsNonExpired(): boolean
⊕ ⊳ isEnabled(): boolean

UserDetailsService

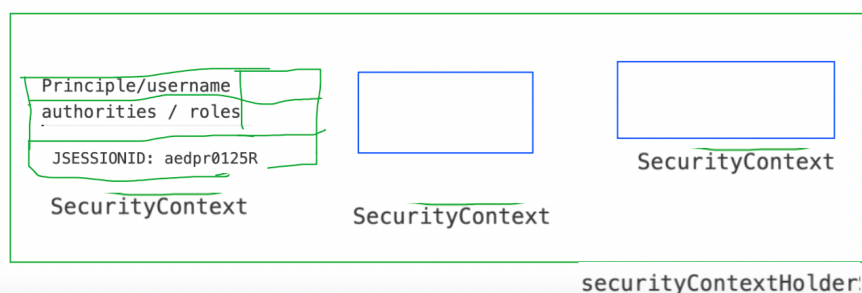UserDetails loadUserByUsername(String username) throws UsernameNotFoundEx

```java
protected void successfulAuthentication(HttpServletRequest request,
    HttpServletResponse response, FilterChain chain,
    Authentication authResult) throws IOException, ServletException {
    SecurityContext context =
        this.securityContextHolderStrategy.createEmptyContext();
    context.setAuthentication(authResult);
    this.securityContextHolderStrategy.setContext(context);

    this.successHandler.onAuthenticationSuccess(request, response,
        authResult);
}
```
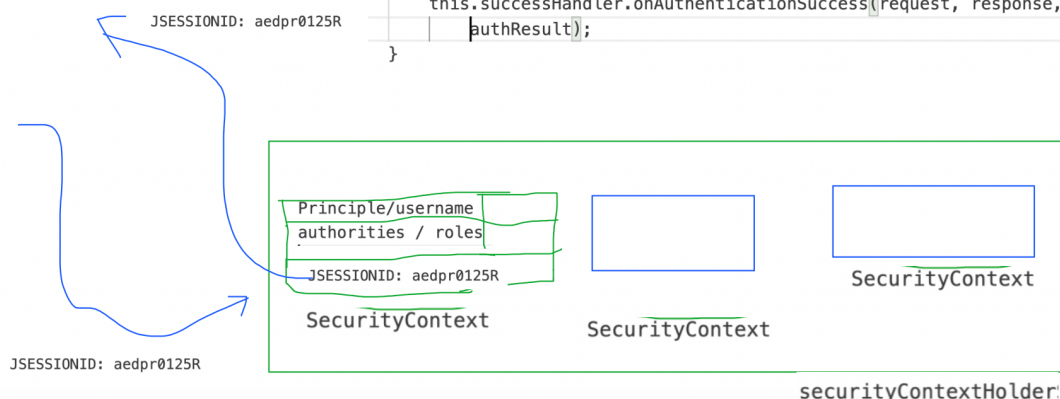
Principle/username
authorities / roles
.
JSESSIONID: aedpr0125R

SecurityContext

SecurityContext

SecurityContext

securityContextHolder

```java
protected void successfulAuthentication(HttpServletRequest request,
    HttpServletResponse response, FilterChain chain,
    Authentication authResult) throws IOException, ServletException {
    SecurityContext context =
        this.securityContextHolderStrategy.createEmptyContext();
    context.setAuthentication(authResult);
    this.securityContextHolderStrategy.setContext(context);

    this.successHandler.onAuthenticationSuccess(request, response,
        authResult);
}
```

JSESSIONID: aedpr0125R

Principle/username
authorities / roles
.
JSESSIONID: aedpr0125R

SecurityContext

SecurityContext

SecurityContext

JSESSIONID: aedpr0125R

securityContextHolder

RESTful Stateful Session tracking is not recommended

S1

X ✓

↓ SESSION ID

S2 ? ?

S3 ?

Username ①
Password

①

②

③

④

JWT token

JWT ⑤
token

JWT
OR
JWT

②

JWT → authorities?
who?

User /
Browser

Credentials →

← Token

Auth Server

Sign Token with
Cert Private Key

X.509 SSL Certificate Pair
(Public and Private Keys)

Token

Endpoint
Reponse

Protected REST
Endpoint

Verify Token with
Cert Public Key

AuthController

login and register

AuthenticationService

token

UserDao

JwtService

```
public String generateToken(UserDetails userDetails) {
```