

<http://localhost:8080/api/orders>

localhost:8080/login

Please sign in

Username

Password

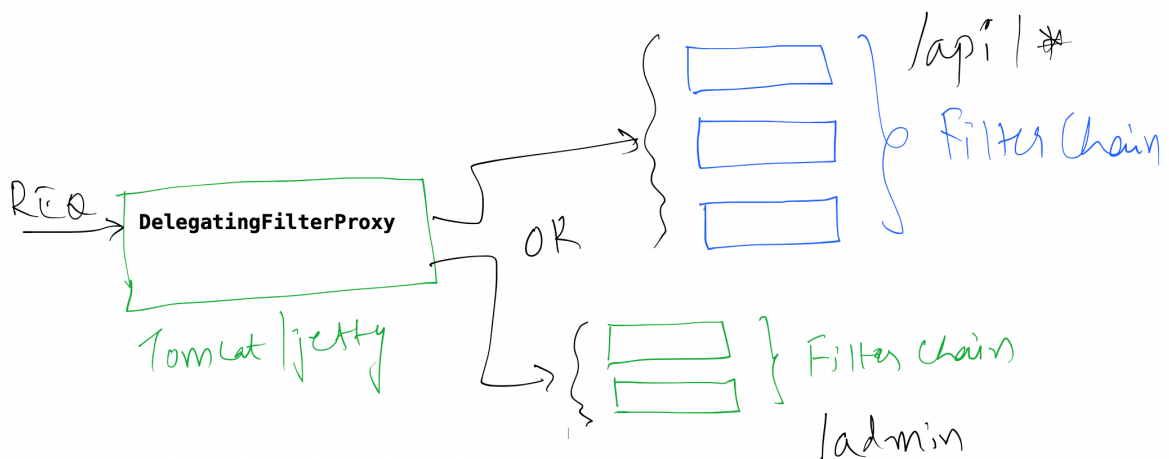
Sign in

All Resources
are protected

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-security</artifactId>
</dependency>
```

Username: user

Using generated security password: 8ef68b24-38f5-4714-8805-c552007c0346



↓ REQ
DelegatingFilterProxy

UsernamePasswordAuthenticationFilter

Please sign in

Authentication

- `getAuthorities()` : Collection<? extend
- `getCredentials()` : Object
- `getDetails()` : Object
- `getPrincipal()` : Object
- `isAuthenticated()` : boolean *false*
- `setAuthenticated(boolean)` : void

```
@Override
public Authentication attemptAuthentication(HttpServletRequest request,
    throws AuthenticationException {
    if (this.postOnly && !request.getMethod().equals("POST")) {
        throw new AuthenticationServiceException("Authentication method
    }
    String username = obtainUsername(request);
    username = (username != null) ? username.trim() : "";
    String password = obtainPassword(request);
    password = (password != null) ? password : "";
    UsernamePasswordAuthenticationToken authRequest = UsernamePasswordAu
        password);
    // Allow subclasses to set the "details" property
    setDetails(request, authRequest);
    return this.getAuthenticationManager().authenticate(authRequest);
}
```

AuthenticationManager

AuthenticationProvider

DaoAuthenticationProvider

InMemoryUserDetailsManager

LdapAuthenticationProvider

JDBC

Memory

LDAP

UserDetailsService

UserDetails loadUserByUsername(String username) throws UsernameNotFoundException;

UserDetailsService

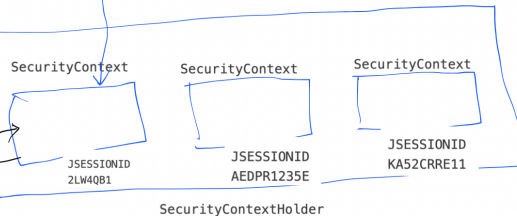
UserDetails loadUserByUsername(String username) throws UsernameNotFoundException;

Authentication

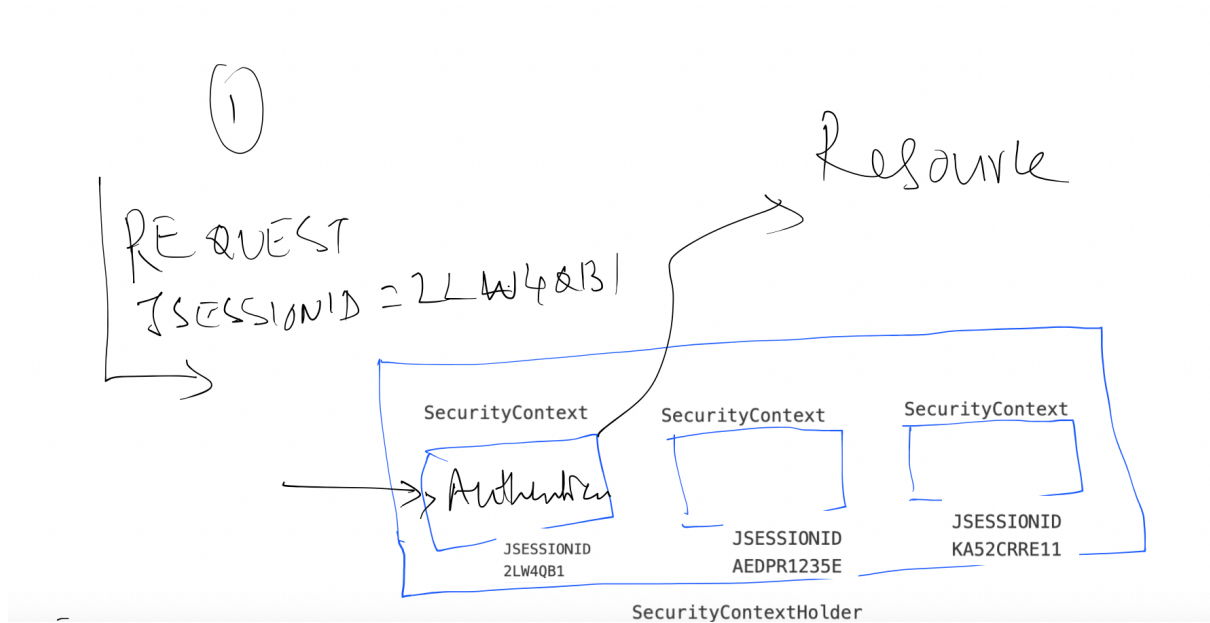
[MANAGER, ADMIN]

- ✓ `getAuthorities()` : Collection<? extend
- ✓ `getCredentials()` : Object *NULL*
- ✓ `getDetails()` : Object
- ✓ `getPrincipal()` : Object
- ✓ `isAuthenticated()` : boolean *TRUE*
- ✓ `setAuthenticated(boolean)` : void

```
protected void successfulAuthentication(HttpServletRequest request, HttpServletResponse response, FilterChain
    Authentication authResult) throws IOException, ServletException {
    SecurityContext context = this.securityContextHolderStrategy.createEmptyContext();
    context.setAuthentication(authResult);
    this.securityContextHolderStrategy.setContext(context, request, response);
    this.securityContextRepository.saveContext(context, request, response);
    if (this.logger.isDebugEnabled()) {
        this.logger.debug(LogMessage.format("Set SecurityContextHolder to %s", authResult));
    }
    this.rememberMeServices.loginSuccess(request, response, authResult);
    if (this.eventPublisher != null) {
        this.eventPublisher.publishEvent(new InteractiveAuthenticationSuccessEvent(authResult, this.getClass()));
    }
    this.successHandler.onAuthenticationSuccess(request, response, authResult);
}
```



RESPONSE
← JSESSIONID 2LW4QB1 → COOKIE



Logout → `SecurityContext` created for the given user is destroyed