Reflektion kring säkerhet och design

I detta projekt har säkerheten varit i stort fokus, jag har använt mig av Keycloak som identitets och åtkomsthanterare. Keycloak har gett stöd för OAuth2 och OpenID Connect, vilket ser till att jag kan arbeta med JWT som bär med sig både information om användaren ex: sub och preferred_username och användarens roller.

För att översätta rollerna från tokenet till Spring Securitys format så byggde jag en JwtAuthConverter. Den tar roller från både realm och klient, så att jag kan använda .requestMatchers och .hasRole i configen.
Jag valde att använda SecurityConfig istället för @PreAuthorize eftersom det gör att all auktorisering ligger samlad på ett ställe och inte blandas in i controllerklassen. På så sätt blir auktoriseringen enkel att läsa av i SecurityConfigen och säkerhetsreglerna blir tydliga och modulära.
Jag har även lagt till felhantering i SecurityConfigen där det skickas 401 om Bearer Token saknas eller är fel, samt 403 om man inte har tillåtelse till endpointen.

Jag la även till felhantering i servicelagret genom en metod som validerar att den som vill uppdatera eller ta bort en bloggpost är ägaren, detta görs genom att jämföra ägaren och den som vill göra ändringen i en hjälpmetod, denna metod kastar ett eget skapat exception (ForbiddenOperationException) om du inte är ägare till posten. Admin kan ta bort allas bloggpost så det finns logik för det också.
Det finns också felhantering om en bloggpost inte finns då kastas ett NotFoundException.

På designsidan av projektet valde jag att arbeta lagerindelat det gör koden mer modulär, lättare att testa och underhålla.
Jag valde också att vid skapande av en ny bloggpost att använda logger istället för system.out då jag läste någonstans att det var industri standard, det ser snyggare ut också.

Det som varit mest utmanande i projektet var Keycloak och säkerhetsdelen, men också det jag har lärt mig mest utav. Det har blivit mycket tydligare nu när jag har jobbat med det i praktiken.

Sammanfattningsvis har projektet gett mig en bättre förståelse för hur man kan kombinera säkerhet med tydlig arkitektur, Designvalen tillsammans med säkerhetslösningen gör applikationen både robust och flexibel.