T4 : SpringSecurity

Cet TP permet de voir différentes implémentations de la sécurité

4.1 Application Web

4.1.1 Authentification mémoire

Ajouter Spring Security dans les dépendances du projet Web précédent

Tester l'accès à l'application

Ajouter une classe de Configuration de type `WebSecurityConfigurerAdapter qui` :

- Autorise l'accès à la page home
- Oblige une authentification pour toutes les autres pages
- Sur un logout réussi, retourner à la page home

Modifier les crédentiels par défaut (*user* et mot de passe généré)

Tester l'authentification via SpringBoot

Quels sont les filtres activés dans la chaîne de filtre ?

Activer le debug et effectuer des requêtes

4.1.2 Authentification custom

Sous le profil ***domainSecurity***, mettre en place une classe implémentant *UserDetailService*, configurer l'authentification afin qu'elle utilise cette classe.

La classe s'appuiera sur le bean *MemberRepository* développé dans les Tps précédents

4.1.3 Authentification Jdbc (Optionnel)

Sous le profil ***jdbcSecurity***, configurer l'authentification en indiquant à Spring la requête à effectuer

4.1.1 Visualisation des chaînes de filtre

Après l'initialisation du contexte Spring, afficher sur la console le contenu du bean ***springSecurityFilterChain***

Passer en mode DEBUG

4.2 API Rest

4.2.1 JWT

Ajouter la dépendance suivante :

```
<dependency>
    <groupId>io.jsonwebtoken</groupId>
    <artifactId>jjwt</artifactId>
    <version>0.7.0</version>
</dependency>
```

Récupérer le code fourni, le regarder, le comprendre.

Configurer la sécurité afin d'intégrer le filtre *JWTFilter* dans la chaîne de sécurité

Après l'initialisation du contexte Spring, afficher sur la console le contenu du bean ***springSecurityFilterChain***

Passer la sécurité en mode DEBUG

Le test de fonctionnement peut s'effectuer via le script *jMeter* également fourni

4.2.2 Authentification via oAuth2

Voir les sources complets fournis