

TP 08

SPRING SECURITY (INTRODUCTION)

Objectif

S'initier à la pratique de Spring Security

Etape 1 – Créer le filtre Spring Security

- Dans le projet « pizzeria-stock-app », ajouter la dépendance vers « spring-security-web » et « spring-security-config ».
- Créer la classe « SecurityConfig » :

```
@EnableWebSecurity
public class SecurityConfig extends WebSecurityConfigurerAdapter {

    @Override
    protected void configure(AuthenticationManagerBuilder auth) throws Exception {
        auth.inMemoryAuthentication()
            .withUser("bob")
            .password("secret")
            .roles("USER");
    }
}
```

Etape 2 – Initialiser l'application Web avec le support de Spring Security

- Créer la classe « SecurityWebApplicationInitializer »

```
public class SecurityWebApplicationInitializer extends
AbstractSecurityWebApplicationInitializer {

    public SecurityWebApplicationInitializer() {
        super(SecurityConfig.class);
    }
}
```

Etape 3 – Supprimer la configuration XML

Dans l'initialiseur Spring MVC écrit précédemment, supprimer la ligne :

```
servletContext.addListener(new ContextLoaderListener(webContext));
```

Cette opération est déjà effectuée par l'initialiseur Spring Security.

Etape 4 – Tester

- Tester l'application
 - L'application est désormais sécurisée
 - Elle devrait fonctionner à l'aide du formulaire fournit par Spring Security

Etape 5 – Distinguer les rôles

- Créer plusieurs utilisateurs avec deux rôles (USER et ADMIN) :

```
public void configure(AuthenticationManagerBuilder auth) throws Exception {  
    auth.inMemoryAuthentication().withUser("user").password("user123").roles("USER");  
    auth.inMemoryAuthentication().withUser("admin").password("admin123").roles("ADMIN");  
}
```

- Activer l'annotation @Secured

```
@EnableGlobalMethodSecurity(securedEnabled = true)
```

A positionner sur la classe Java Config de Spring Security.

- A l'aide de l'annotation @Secured
 - Autoriser l'accès à « /mvc/ingredients » aux utilisateurs aux rôles « ADMIN » ou « USER »
 - Autoriser l'accès à « /mvc/performances » aux utilisateurs au rôle « ADMIN » uniquement.

Etape 6 – Tester

- Tester l'application
 - L'application est désormais sécurisée
 - Elle devrait fonctionner à l'aide du formulaire fourni par Spring Security