Spring Boot Security et Spring Boot REST API - Authentification de Base

Démonstration 3 du module 7

L'objectif de cette démonstration est la mise en place d'une authentification de données pour déclarer les utilisateurs.

Contexte

Reprendre l'application de démonstration d'API REST

Déroulement

Configuration

• Création de la classe de configuration SecurityConfig:

```
package fr.eni.demo.security;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Configuration;
org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;
@Configuration
@EnableWebSecurity
public class SecurityConfig extends WebSecurityConfigurerAdapter {
    protected void configure(HttpSecurity http) throws Exception
         .csrf().disable()
         .authorizeRequests().anyRequest().authenticated()
         .and()
         .httpBasic();
    @Autowired
    public void configureGlobal(AuthenticationManagerBuilder auth)
            throws Exception
       //{noop} call Class NoOpPasswordEncoder
        auth.inMemoryAuthentication()
```

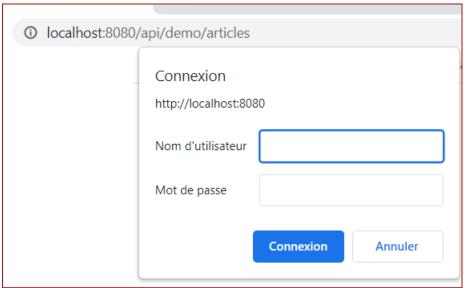


```
.withUser("admin")
    .password("{noop}test")
    .roles("USER");
}
```

Elle oblige pour toutes les requêtes à s'authentifier.

Exécution:

- Utiliser directement votre navigateur pour tester
 - Dès que vous voulez accéder à une URL (http://localhost:8080/api/demo/articles) de l'API; il va vous ouvrir la popup de connexion utilisateur:



o Et ainsi sauvegarder en session votre utilisateur reconnu:

