Spring Security 5 Login Form Example

Dans cette partie Spring Security 5, apprenez à ajouter une sécurité basée sur un formulaire de connexion personnalisé à notre application Spring WebMVC. J'utilise Spring security 5 pour créer cet exemple. Ce didacticiel traite également de la déconnexion de la session.

1. Inclure les dépendances Spring Security 5

Incluez des pots de sécurité à spring-boot. J'utilise maven, j'ai donc ajouté des dépendances respectives pour la version de sécurité du spring 5.0.7.RELEASE.

```
<properties>
        <failOnMissingWebXml>false</failOnMissingWebXml>
        <spring.version>5.2.0.RELEASE</spring.version>
</properties>

<!-- Spring MVC Dependency -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>${spring.version}</version>
</dependency>

<!-- Spring Security Core -->
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-core</artifactId>
    <version>${spring.version}</version>
</dependency>

<!-- Spring Security Config -->
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-config</artifactId>
    <version>${spring.version}</version>
```

```
</dependency>

<!-- Spring Security Web -->
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-web</artifactId>
    <version>${spring.version}</version>
</dependency>
```

2. Créer une configuration de sécurité Spring

2.1. Configurer l'authentification et la sécurité des URL

- J'ai créé cette configuration de sécurité simple et ajouté deux utilisateurs de démonstration ' `user`' et ' `admin`'.
- `@EnableWebSecurity` active la prise en charge de la sécurité Web de Spring Security et fournit également l'intégration Spring MVC.
- `WebSecurityConfigurerAdapter` fournit un ensemble de méthodes utilisées pour activer une configuration de sécurité Web spécifique.
- `configure(HttpSecurity http)` est utilisé pour sécuriser différentes URL nécessitant une sécurité.
- Il configure la page de connexion personnalisée à l'URL `/login`. Si le nom d'utilisateur/mot de passe correspond, la demande est redirigée vers `/home`, sinon la page de connexion est actualisée avec le message d'erreur correspondant.
- Il configure également la déconnexion de la session . Après la déconnexion, l'utilisateur est à nouveau redirigé vers la page de connexion.

J'ai utilisé l'authentification en mémoire à l'aide de `auth.inMemoryAuthentication()`. Vous pouvez configurer l'authentification JDBC à l'aide `auth.jdbcAuthentication()` ou l'authentification LDAP à l'aide de `auth.ldapAuthentication()`.

2.2. Lier la sécurité Spring à l'application Web

Dans les applications Web Spring, la sécurité est implémentée à l'aide de `DelegatingFilterProxy`. Pour l'enregistrer, avec le conteneur Spring en configuration Java, vous devrez utiliser `AbstractSecurityWebApplicationInitializer`.

Le spring-boot détectera l'instance de cette classe lors du démarrage de l'application et enregistrera le `DelegatingFilterProxy` pour utiliser le `springSecurityFilterChain` avant tout autre filtre enregistré. Il enregistre également un `ContextLoaderListener`

Incluez également `SecurityConfig` à `AppInitializer`.

3. Ajouter un formulaire de connexion personnalisé

3.1. Contrôleur de connexion

Ajoutez la méthode du contrôleur de connexion pour gérer les demandes de connexion et de déconnexion.

3.2. login.jsp

Créer un `login.jsp` fichier qui acceptera `username` et `password`; et publiez-les sur URL `/login`.

login.jsp :

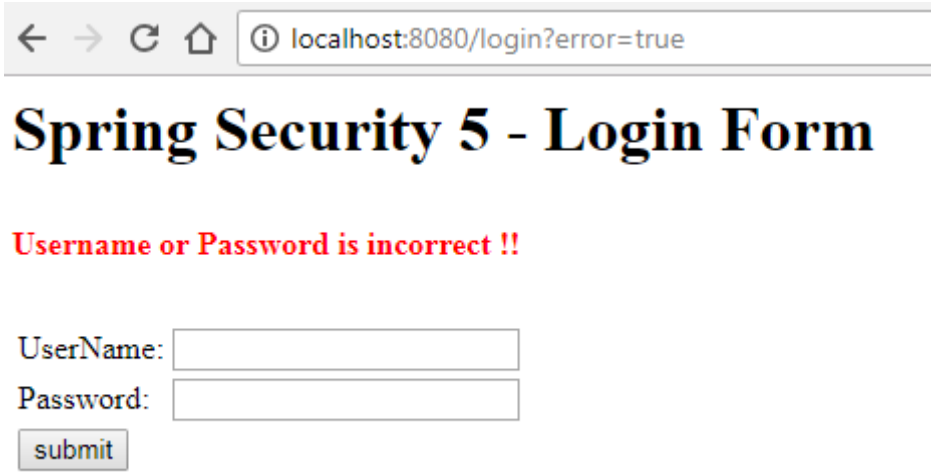
```
%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<html>
<body onload='document.loginForm.username.focus();'>
    <h1>Spring Security 5 - Login Form</h1>

    <c:if test="${not empty errorMessge}"><div style="color:red;
font-weight: bold; margin: 30px 0px;">${errorMessge}</div></c:if>

    <form name='login' action="/login" method='POST'>
        <table>
            <tr>
                <td>UserName:</td>
                <td><input type='text' name='username' value=''></td>
            </tr>
            <tr>
                <td>Password:</td>
                <td><input type='password' name='password' /></td>
            </tr>
            <tr>
                <td colspan='2'><input name="submit" type="submit"
value="submit" /></td>
            </tr>
        </table>
        <input type="hidden" name="${_csrf.parameterName}"
value="${_csrf.token}" />
    </form>
</body>
</html>
```

Démo du formulaire de connexion Spring Security 5

1. Démarrez l'application avec la commande maven run `tomcat7:run`. Lancer la page d'accueil `http://localhost:8080/home`. Il sera redirigé vers la page de connexion `http://localhost:8080/login`.

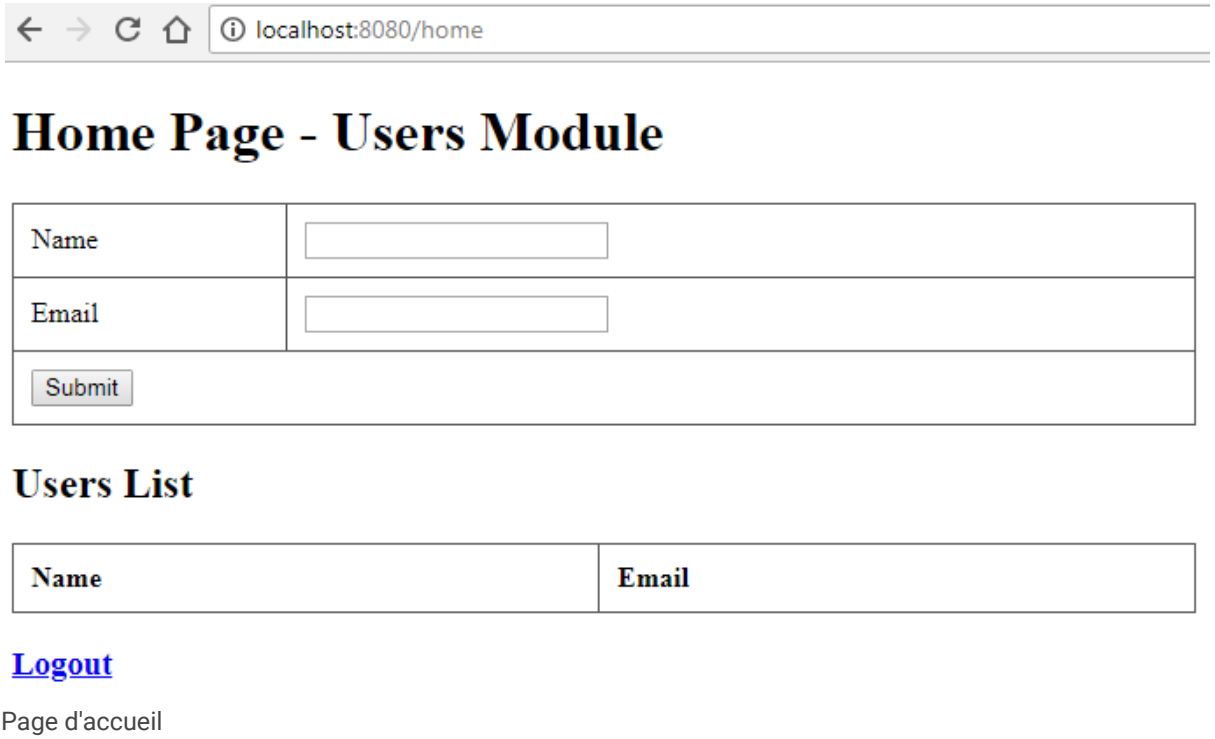
Formulaire de connexion

2. Saisissez un nom d'utilisateur ou un mot de passe INCORRECT. Un message d'erreur de connexion s'affiche.



identifiant ou mot de passe incorrect

3. Entrez la combinaison CORRECTE du nom d'utilisateur et du mot de passe. `user` et `123456`. La page d'accueil s'affichera.



Page d'accueil

4. Cliquez sur le bouton de déconnexion. L'utilisateur sera déconnecté et redirigé vers la page de connexion.

Déconnecté