

```
1 HOL : Spring Security
2 -----
3 Task1. 간단한 인증처리하기
4 1. Spring Legacy Project 생성
5   1)In Package Explorer > right-click > New > Other > Spring > Spring Legacy Project
6   2)Project name : SpringSecurityDemo
7   3)Select Spring MVC Project > Next
8   4)com.example.biz > Finish
9
10 2. server.xml에 project 추가
11   1)Tomcat v9.0 Server at localhost > right-click > Add and Remove
12   2)SpringSecurityDemo > Add > Finish > OK
13
14 3. Spring security library download and install
15   1)In pom.xml 아래와 같이 수정
16   <java-version>1.8</java-version>
17   <org.springframework-version>5.2.0.RELEASE</org.springframework-version>
18   <org.aspectj-version>1.9.4</org.aspectj-version>
19   <org.slf4j-version>1.7.28</org.slf4j-version>
20   ...
21   <!-- Servlet -->
22   <dependency>
23     <groupId>javax.servlet</groupId>
24     <artifactId>javax.servlet-api</artifactId>
25     <version>4.0.1</version>
26     <scope>provided</scope>
27   </dependency>
28   <dependency>
29     <groupId>javax.servlet.jsp</groupId>
30     <artifactId>javax.servlet.jsp-api</artifactId>
31     <version>2.3.3</version>
32     <scope>provided</scope>
33   </dependency>
34   ...
35   <!-- Test -->
36   <dependency>
37     <groupId>junit</groupId>
38     <artifactId>junit</artifactId>
39     <version>4.12</version>
40     <scope>test</scope>
41   </dependency>
42
43   2)In mvnrepository, search 'spring security'
44   3)Select 'Spring Security Core' 5.2.0
45   4)Select 'Spring Security Web' 5.2.0
46   5)Select 'Spring Security Config' 5.2.0
47   6)Select 'Spring Security Taglibs' 5.2.0
48   7)pom.xml > right-click > Run As > Maven clean and Maven install
49
50
51 4. DelegatingFilterProxy filter 구성
52   1)In web.xml 에 code 추가
```

```
53
54     <filter>
55         <filter-name>springSecurityFilterChain</filter-name>
56         <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>
57     </filter>
58
59     <filter-mapping>
60         <filter-name>springSecurityFilterChain</filter-name>
61         <url-pattern>/*</url-pattern>
62     </filter-mapping>
```

### 5. 보안 관련 설정 File 만들기

- 1)/src/main/webapp/WEB-INF/config folder 생성
- 2)project > right-click > Build Path > Config Build Path
- 3)Source tab > Add Folder > select src/main/webapp/WEB-INF/
- 4)Click [Create New Folder] > Folder name : config > OK > Apply and Close
- 5)src/main/webapp/WEB-INF/config > right-click > New > Other > Springt > Spring Bean Configuration File
- 6)File name : securityContext.xml > Next
- 7)security check할 때 반드시 <https://www.springframework.org/schema/security/spring-security.xsd> 즉 xsd의 version이 없는 것을 선택해야 함.
- 8)Finish
- 9)In /WEB-INF/web.xml에 보안 관련 설정 file 등록

```
76     <context-param>
77         <param-name>contextConfigLocation</param-name>
78         <param-value>classpath:securityContext.xml</param-value>
79     </context-param>
```

### 6. Web 요청에 대한 보안 구성

- 1)In securityContext.xml code 추가

```
85     <security:http auto-config="true">
86         <security:intercept-url pattern="/login.html" access="hasRole('ROLE_USER')" />
87         <security:intercept-url pattern="/welcome.html" access="hasRole('ROLE_ADMIN')" />
88     </security:http>
89
90     <security:authentication-manager>
91         <security:authentication-provider>
92             <security:user-service>
93                 <security:user name="javaexpert" password="{noop}12345678" authorities="ROLE_USER"/>
94                 <security:user name="admin" password="{noop}12345678"
95                     authorities="ROLE_ADMIN,ROLE_USER"/>
96             </security:user-service>
97         </security:authentication-provider>
98     </security:authentication-manager>
```

### 7. web.xml 에서 코드 수정

```
102 <servlet-mapping>
103   <servlet-name>appServlet</servlet-name>
104   <url-pattern>*.html</url-pattern> <--여기 수정할 것
105 </servlet-mapping>
106
107 8. Test
108 1)/WEB-INF/views/security folder 생성
109 2)/WEB-INF/views/security/login.jsp
110
111   <body>
112     <h1>login.jsp</h1>
113   </body>
114
115 3)/WEB-INF/views/security/welcome.jsp
116
117   <body>
118     <h1>welcome.jsp</h1>
119   </body>
120
121 4)/WEB-INF/views/home.jsp
122
123   <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
124   <%@ page session="false" %>
125   <html>
126   <head>
127     <title>Home</title>
128   </head>
129   <body>
130     <h1>
131       ${greeting}
132     </h1>
133
134   </body>
135   </html>
136
137 5)HomeController.java 코드 수정
138
139   @RequestMapping(value = "/index.html", method = RequestMethod.GET)
140   public String home(Model model) {
141     model.addAttribute("greeting", "Hello! Spring Security");
142     return "home";
143   }
144
145   @RequestMapping("/login.html")
146   public String login(Model model) {
147     return "security/login";
148   }
149
150   @RequestMapping("/welcome.html")
151   public String welcome(Model model) {
152     return "security/welcome";
153   }
```

154  
155 6)project > right-click > Run As > Run on Server  
156  
157 7)/login.html로 들어가면 정상적으로 login.jsp 로 이동한다.  
158 8)username과 password를 미리 저장된 값을 넣지 않으면 Bad credentials로 로그인에러 발생한다.  
159 9)/welcome.html로 들어가면 User와 Password를 입력할 수 있는 창으로 이동한다.  
160 10)여기서 user와 password를 잘못 입력하면 'Bad credentials' 에러 메시지가 나온다.  
161 11)User에 미리 설정한 javaexpert, Password에 12345678을 입력하면 403 Forbidden 오류가 나온다.  
162 12)즉 권한이 없다는 뜻이다.  
163 13)만일 미리 설정한 admin/12345678을 입력하면 정상적으로 welcome.jsp로 이동한다.  
164 14)왜냐하면 이 계정은 ROLE\_ADMIN 권한을 갖고 있기 때문이다.

165  
166  
167 -----

## 168 Task2. Login page 만들기

### 169 1. securityContext.xml code 변경

170  
171 <security:http auto-config="true">  
172 <security:form-login login-page="/loginForm.html"  
173 username-parameter="j\_username"  
174 password-parameter="j\_password"  
175 login-processing-url="/j\_spring\_security\_check" />  
176 <security:intercept-url pattern="/login.html"  
177 access="hasRole('ROLE\_USER')" />  
178 <security:intercept-url  
179 pattern="/welcome.html" access="hasRole('ROLE\_ADMIN')" />  
180 <security:csrf disabled="true"/>  
181 </security:http>

### 182 183 184 2. /views/security/loginForm.jsp

185  
186 <%@ page language="java" contentType="text/html; charset=UTF-8"  
187 pageEncoding="UTF-8"%>  
188 <%@ taglib uri="<http://java.sun.com/jsp/jstl/core>" prefix="c" %>  
189 <!DOCTYPE html>  
190 <html>  
191 <head>  
192 <meta charset="UTF-8">  
193 <title>로그인 창</title>  
194 </head>  
195 <body>  
196 <h1>loginForm.jsp</h1>  
197  
198 <form action="<c:url value="j\_spring\_security\_check" />" method="post">  
199 ID : <input type="text" name="j\_username"> <br />  
200 PW : <input type="text" name="j\_password"> <br />  
201 <input type="submit" value="LOGIN"> <br />  
202 </form>  
203 </body>  
204 </html>  
205

```

206
207 3. HomeController.java code 추가
208   @RequestMapping("/loginForm.html")
209   public String loginForm(Model model) {
210       return "security/loginForm";
211   }
212
213
214 4. Test
215 1)project > right-click > Run As > Run on Server
216 2)http://localhost:8080/biz/login.html 을 요청하면 http://localhost:8080/biz/loginForm.html 로 redirect
217 3)ID와 Password를 미리 설정한 javaexpert/12345678을 넣으면 정상적으로 login.jsp로 이동한다.
218 4)http://localhost:8080/biz/welcome.html 을 요청하면 http://localhost:8080/biz/loginForm.html 로
  redirect
219 5)만일 ID와 Password를 미리 설정한 javaexpert/12345678을 넣으면 403 forbidden error가 발생한다.
220 6)만일 ID와 Password를 미리 설정한 admin/12345678을 넣으면 정상적으로 welcome.jsp로 이동한다.
221 7)만일 잘못된 ID와 Password를 입력하면 계속 loginForm.html?error가 나타난다.
222
223 8)만일 Could not verify the provided CSRF token because your session was not found. 오류가 발생하면
  아래의 조치를 취한다.
224   -securityContext.xml 에서 코드 추가한다.
225       <security:http auto-config="true">
226           <security:form-login login-page="/loginForm.html" />
227           <security:intercept-url pattern="/login.html"
228               access="hasRole('ROLE_USER')" />
229           <security:intercept-url
230               pattern="/welcome.html" access="hasRole('ROLE_ADMIN')" />
231           <security:csrf disabled="true"/>          <---cpde 추가
232       </security:http>
233
234
235 5. login 실패시 실패 message 출력하기
236 1)securityContext.xml code 추가
237
238   <security:form-login login-page="/loginForm.html"
239       username-parameter="j_username"
240       password-parameter="j_password"
241       authentication-failure-url="/loginForm.html?ng"   <---code 추가
242       login-processing-url="/j_spring_security_check" />
243
244 2)loginForm.jsp 코드 추가
245
246   <h1>loginForm.jsp</h1>
247   <c:url value="j_spring_security_check" var="loginUrl" />  <--추가
248   <form action="{loginUrl}" method="post">                  <-- 수정
249       <c:if test="{param.ng != null}">
250           <p>
251               Login Failure<br />
252               <c:if test="{SPRING_SECURITY_LAST_EXCEPTION != null}">
253                   Message : <c:out value="{SPRING_SECURITY_LAST_EXCEPTION.message}" />
254               </c:if>
255           </p>

```

```
256     </c:if>
257     ID : <input type="text" name="j_username"> <br />
258     PW : <input type="text" name="j_password"> <br />
259     <input type="submit" value="LOGIN"> <br />
260 </form>
261
262 3)Test
263 -http://localhost:8080/biz/login.html 을 요청하면 http://localhost:8080/biz/loginForm.html 로 redirect
264 -ID와 Password를 미리 설정한 javaexpert/12345678을 넣으면 정상적으로 login.jsp로 이동한다.
265 -만일 잘못된 ID나 Password를 입력하면 아래와 같은 오류 message가 나온다.
266
267     Login Failure
268     Message : Bad credentials
269
270 -http://localhost:8080/biz/welcome.html 을 요청하면 http://localhost:8080/biz/loginForm.html 로
271 redirect
272 -만일 ID와 Password를 미리 설정한 javaexpert/12345678을 넣으면 403 forbidden error가 발생한다.
273 -만일 ID와 Password를 미리 설정한 admin/12345678을 넣으면 정상적으로 welcome.jsp로 이동한다.
274 -만일 잘못된 ID와 Password를 입력하면 계속 loginForm.html?ng가 나오면서 아래의 message가 나타
275 난다.
276
277     Login Failure
278     Message : Bad credentials
279 -----
280 Task3. login, logout 상태 표시하기
281 1. security/login.jsp page code 추가
282 <%@ page language="java" contentType="text/html; charset=UTF-8"
283     pageEncoding="UTF-8"%>
284 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
285 <body>
286     <h1>login.jsp</h1>
287
288     <c:if test="${not empty pageContext.request.userPrincipal}">
289         <p>현재 로그인 상태</p>
290     </c:if>
291     <c:if test="${empty pageContext.request.userPrincipal}">
292         <p>현재 로그아웃 상태</p>
293     </c:if>
294
295     User ID : ${pageContext.request.userPrincipal}<br />
296     <a href="${pageContext.request.contextPath}/">Log Out</a>
297 </body>
298
299 2. Test
300 1)http://localhost:8080/biz/login.html 을 요청하면 http://localhost:8080/biz/loginForm.html 로 redirect
301 2)아이디와 패스워드를 미리 설정한 javaexpert/12345678을 넣으면 정상적으로 login.jsp로 이동한다.
302 3)이 때 화면에는 아래의 message가 나온다.
303
304     현재 로그인 상태
305     User ID :
```

```

306 org.springframework.security.authentication.UsernamePasswordAuthenticationToken@b09a9234:
307 Principal: org.springframework.security.core.userdetails.User@f486f6cc:
308 Username: javaexpert;
309 Password: [PROTECTED];
310 Enabled: true;
311 AccountNonExpired: true;
312 credentialsNonExpired: true;
313 AccountNonLocked: true;
314 Granted Authorities: ROLE_USER;
315 Credentials: [PROTECTED];
316 Authenticated: true;
317 Details: org.springframework.security.web.authentication.WebAuthenticationDetails@166c8:
318 RemoteIpAddress: 0:0:0:0:0:0:1;
319 SessionId: 949A2731FFC91E3C350818E0772C8E45;
320 Granted Authorities: ROLE_USER

```

```

321 Log Out

```

```

322
323 4)여기서 username만 출력하려면

```

```

324
325 User ID : ${pageContext.request.userPrincipal.name} 으로 변경한다.

```

```

326
327 3. securityContext.xml code 추가

```

```

328
329 <security:form-login login-page="/loginForm.html"
330 username-parameter="j_username"
331 password-parameter="j_password"
332 authentication-failure-url="/loginForm.html?ng"
333 login-processing-url="/j_spring_security_check" />
334 <security:logout logout-url="/j_spring_security_logout"/> <!--code 추가
335

```

```

336 4. Test

```

```

337 1)로그인 창에서 Log Out link를 클릭하면 index.html로 이동한다.

```

```

338
339
340 -----

```

```

341 Task4. UserDetailsService 이용하기

```

```

342 1. Spring Legacy Project 생성

```

```

343 1)In Package Explorer > right-click > New > Other > Spring > Spring Legacy Project

```

```

344 2)Project name : SpringSecurityDemo1

```

```

345 3)Select Spring MVC Project > Next

```

```

346 4)com.example.biz > Finish

```

```

347

```

```

348 2. server.xml에 project 추가

```

```

349 1)Tomcat v9.0 Server at localhost > right-click > Add and Remove

```

```

350 2)SpringSecurityDemo1 > Add > Finish > OK

```

```

351

```

```

352 3. Spring security library download and install

```

```

353 1)In pom.xml 아래와 같이 수정

```

```

354 <java-version>1.8</java-version>

```

```

355 <org.springframework-version>5.2.0.RELEASE</org.springframework-version>

```

```

356 <org.aspectj-version>1.9.4</org.aspectj-version>

```

```
357     <org.slf4j-version>1.7.28</org.slf4j-version>
358
359 2)version 수정
360     <dependency>
361         <groupId>javax.servlet</groupId>
362         <artifactId>javax.servlet-api</artifactId>
363         <version>4.0.1</version>
364         <scope>provided</scope>
365     </dependency>
366     <dependency>
367         <groupId>javax.servlet.jsp</groupId>
368         <artifactId>javax.servlet.jsp-api</artifactId>
369         <version>2.3.3</version>
370         <scope>provided</scope>
371     </dependency>
372     <dependency>
373         <groupId>junit</groupId>
374         <artifactId>junit</artifactId>
375         <version>4.12</version>
376         <scope>test</scope>
377     </dependency>
378
379 3)In mvnrepository, search 'spring security'
380 -Select 'Spring Security Core' 5.2.0.RELEASE
381 -Select 'Spring Security Web' 5.2.0.RELEASE
382 -Select 'Spring Security Config' 5.2.0.RELEASE
383 -Select 'Spring Security Taglibs' 5.2.0.RELEASE
384
385     <dependency>
386         <groupId>org.springframework.security</groupId>
387         <artifactId>spring-security-core</artifactId>
388         <version>5.2.0.RELEASE</version>
389     </dependency>
390     <dependency>
391         <groupId>org.springframework.security</groupId>
392         <artifactId>spring-security-web</artifactId>
393         <version>5.2.0.RELEASE</version>
394     </dependency>
395     <dependency>
396         <groupId>org.springframework.security</groupId>
397         <artifactId>spring-security-config</artifactId>
398         <version>5.2.0.RELEASE</version>
399     </dependency>
400     <dependency>
401         <groupId>org.springframework.security</groupId>
402         <artifactId>spring-security-taglibs</artifactId>
403         <version>5.2.0.RELEASE</version>
404     </dependency>
405
406 4)pom.xml > right-click > Run As > Maven clean and Maven install
407 5)project > right-click > Properties > Project Facets > Java > Select 1.8
408 6)Runtimes Tab > Check Apache Tomcat v9.0
```



409 7)Click Apply and Close

410

411 4. HomeController.java 코드 수정

412

413 package com.example.biz;

414

415 import java.util.Date;

416

417 import org.springframework.stereotype.Controller;

418 import org.springframework.ui.Model;

419 import org.springframework.web.bind.annotation.RequestMapping;

420 import org.springframework.web.bind.annotation.RequestMethod;

421

422 @Controller

423 public class HomeController {

424

425 @RequestMapping(value = "/", method = RequestMethod.GET)

426 public String home(Model model) {

427 model.addAttribute("serverTime", new Date());

428 return "home";

429 }

430 }

431

432 5. project > right-click > Run As > Run on Server

433

434 6. src/main/webapp/WEB-INF/spring/Spring Bean Configuration File > spring-security.xml 파일 생성

435 1)check 'security'

436 2)security check할 때 반드시 <https://www.springframework.org/schema/security/spring-security.xsd> 즉  
xsd의 version이 없는 것을 선택해야 함.

437

438 <security:http auto-config="true" use-expressions="true">

439 </security:http>

440

441 <security:authentication-manager>

442 </security:authentication-manager>

443

444 7. web.xml 코드 수정

445

446 <context-param>

447 <param-name>contextConfigLocation</param-name>

448 <param-value>/WEB-INF/spring/spring-security.xml</param-value>

449 </context-param>

450 ...

451 ...

452 <filter>

453 <filter-name>springSecurityFilterChain</filter-name>

454 <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>

455 </filter>

456

457 <filter-mapping>

458 <filter-name>springSecurityFilterChain</filter-name>

459 <url-pattern>/\*</url-pattern>

```
460 </filter-mapping>
461
462
463 8. WEB-INF/spring/appServlet/servlet-context.xml 수정
464
465 <context:component-scan base-package="com.example" />
466
467 9. project > right-click > Run As > Run on Server
468
469 10. spring-security.xml
470
471 <?xml version="1.0" encoding="UTF-8"?>
472 <beans xmlns="http://www.springframework.org/schema/beans"
473     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
474     xmlns:security="http://www.springframework.org/schema/security"
475     xsi:schemaLocation="http://www.springframework.org/schema/security
476         http://www.springframework.org/schema/security/spring-security.xsd
477         http://www.springframework.org/schema/beans
478         http://www.springframework.org/schema/beans/spring-beans.xsd">
479
480 <security:http auto-config="true" use-expressions="true">
481     <security:intercept-url pattern="/login"
482         access="permitAll" />
483     <security:intercept-url pattern="/resources/**"
484         access="permitAll" />
485     <security:intercept-url pattern="/**"
486         access="hasRole('ROLE_USER')" />
487     <security:form-login login-page="/login"
488         default-target-url="/welcome" username-parameter="j_username"
489         password-parameter="j_password"
490         authentication-failure-url="/login?error"
491         login-processing-url="/j_spring_security_check"
492         always-use-default-target="true" />
493     <security:logout invalidate-session="true"
494         logout-url="/logout" logout-success-url="/login?logout" />
495     <security:csrf />
496 </security:http>
497
498 <security:authentication-manager>
499     <security:authentication-provider user-service-ref="memberService" />
500 </security:authentication-manager>
501
502 <bean id="memberService"
503     class="com.example.service.MemberService">
504     <property name="userService" ref="userService" />
505 </bean>
506 <bean id="userService"
507     class="com.example.service.UserService">
508     <property name="userDao" ref="userDao" />
509 </bean>
510 <bean id="userDao" class="com.example.dao.UserDao">
511     <property name="username" value="{noop}javaexpert" />
512 </bean>
```

```
510     <property name="password" value="{noop}12345678" />
511   </bean>
512 </beans>
513
514
515 11. HomeController.java
516
517 package com.example.biz;
518
519 import java.util.Date;
520
521 import org.springframework.stereotype.Controller;
522 import org.springframework.ui.Model;
523 import org.springframework.web.bind.annotation.RequestMapping;
524 import org.springframework.web.bind.annotation.RequestMethod;
525 import org.springframework.web.bind.annotation.RequestParam;
526 import org.springframework.web.servlet.ModelAndView;
527
528 @Controller
529 public class HomeController {
530
531     @RequestMapping(value = "/", method = RequestMethod.GET)
532     public String home(Model model) {
533         model.addAttribute("serverTime", new Date());
534         return "home";
535     }
536
537     @RequestMapping(value = "/login", method = RequestMethod.GET)
538     public ModelAndView login(
539         @RequestParam(value = "error", required = false) String error,
540         @RequestParam(value = "logout", required = false) String logout) {
541         ModelAndView model = new ModelAndView();
542         if (error != null) {
543             model.addObject("error", "Invalid username and password!");
544         }
545         if (logout != null) {
546             model.addObject("msg", "You've been logged out successfully.");
547         }
548         model.setViewName("login");
549         return model;
550     }
551
552     @RequestMapping("/welcome")
553     public String welcome(Model model) {
554         return "welcome";
555     }
556
557     @RequestMapping(value = "/logout", method = RequestMethod.GET)
558     public String logout(Model model) {
559         return "logout";
560     }
561 }
```

```
562
563
564 12. com.example.vo.UserVO.java
565
566     package com.example.vo;
567
568     public class UserVO {
569         private String username;
570         private String password;
571
572         public UserVO() {}
573
574         public UserVO(String username, String password) {
575             this.username = username;
576             this.password = password;
577         }
578
579         public String getUsername() {
580             return username;
581         }
582
583         public void setUsername(String username) {
584             this.username = username;
585         }
586
587         public String getPassword() {
588             return password;
589         }
590
591         public void setPassword(String password) {
592             this.password = password;
593         }
594
595         @Override
596         public String toString() {
597             return "UserVO [username=" + username + ", password=" + password + "]";
598         }
599     }
600
601
602 13. com.example.dao.UserDao.java
603
604     package com.example.dao;
605
606     import org.springframework.stereotype.Repository;
607
608     import com.example.vo.UserVO;
609
610     @Repository("userDao")
611     public class UserDao {
612         private String username;
613         private String password;
```

```
614
615     public void setUsername(String username) {
616         this.username = username;
617     }
618
619     public void setPassword(String password) {
620         this.password = password;
621     }
622
623     public UserVO getUsersByID(String username) {
624         return new UserVO(this.username, this.password);
625     }
626 }
```

629 14. com.example.service.UserService.java

```
630
631     package com.example.service;
632
633     import org.springframework.beans.factory.annotation.Autowired;
634     import org.springframework.stereotype.Service;
635
636     import com.example.dao.UserDao;
637     import com.example.vo.UserVO;
638
639     @Service("userService")
640     public class UserService {
641         private UserDao userDao;
642
643         public void setUserDao(UserDao userDao) {
644             this.userDao = userDao;
645         }
646
647         public UserVO getUsersByID(String username) {
648             return this.userDao.getUsersByID(username);
649         }
650     }
```

653 15. com.example.service.MemberService.java

```
654
655     package com.example.service;
656
657     import java.util.ArrayList;
658     import java.util.Collection;
659
660     import org.springframework.beans.factory.annotation.Autowired;
661     import org.springframework.security.core.authority.SimpleGrantedAuthority;
662     import org.springframework.security.core.userdetails.User;
663     import org.springframework.security.core.userdetails.UserDetails;
664     import org.springframework.security.core.userdetails.UserDetailsService;
665     import org.springframework.security.core.userdetails.UsernameNotFoundException;
```

```
666 import org.springframework.stereotype.Service;
667
668 import com.example.vo.UserVO;
669
670 @Service("memberService")
671 public class MemberService implements UserDetailsService{
672     private UserService userService;
673
674     /*
675     * 사용자가 입력한 ID/PWD 를 검증하기 위해 저장소에 그 ID/PWD 가 있는지 확인하는 source이다. 저
676     장소는 in-memory가 될 수도 있고,
677     * DB가 될 수도 있고 어떠한 것(Facebook,Naver등등)도 될 수있다. userService는 DB 에 접근하기 위한
678     service이다.
679     */
680
681     public void setUserService(UserService userService) {
682         this.userService = userService;
683     }
684
685     @Override
686     public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
687         UserVO userVO = this.userService getUsersByID(username);
688         if(userVO == null)
689             throw new UsernameNotFoundException("No user found with username " +
690                 userVO.getUsername());
691         Collection<SimpleGrantedAuthority> roles = new ArrayList<SimpleGrantedAuthority>();
692         roles.add(new SimpleGrantedAuthority("ROLE_USER"));
693         UserDetails user = new User(username, userVO.getPassword(), roles);
694         return user;
695     }
696 }
697
698 16. views/login.jsp
699
700 <%@ page language="java" contentType="text/html; charset=UTF-8"
701     pageEncoding="UTF-8"%>
702 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
703 <!DOCTYPE html>
704 <html>
705 <head>
706     <title>로그인 페이지</title>
707     <meta charset="UTF-8">
708     <script>
709         function doLogin() {
710             if(frm.j_username.value == "") {
711                 alert("아이디를 입력해주세요.");
712                 return;
713             }
714             if(frm.j_password.value == "") {
715                 alert("패스워드를 입력해주세요.");
716                 return;
717             }
718         }
```

```

715     }
716     frm.submit();
717 }
718 </script>
719 </head>
720 <body>
721     <form name="frm" action="j_spring_security_check" method="post">
722         <ul>
723             <li>
724                 <label for="userID">ID</label>
725                 <input id = "userID" type="userID" name="j_username" placeholder="ID" required>
726             </li>
727             <li>
728                 <label for="password">Password</label>
729                 <input id = "password" type="password" name="j_password" placeholder="PASSWORD"
730                     required> </li>
731             <li>
732                 <input type="button" value="로그인" onclick="doLogin()"/>
733             </li>
734         </ul>
735         <input type="hidden" name="${_csrf.parameterName}"
736             value="${_csrf.token}" />
737     </form>
738     <c:if test="${not empty error}">
739         <div class="error">${error}</div>
740     </c:if>
741     <c:if test="${not empty msg}">
742         <div class="msg">${msg}</div>
743     </c:if>
744 </body>
745 </html>

```

746  
747  
748 17. views/welcome.jsp

```

749
750 <%@ page language="java" contentType="text/html; charset=UTF-8"
751     pageEncoding="UTF-8"%>
752 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
753 <!DOCTYPE html>
754 <html>
755 <head>
756 <meta charset="UTF-8">
757 <title>Member Page</title>
758 </head>
759 <body>
760     <c:if test="${pageContext.request.userPrincipal.name != null}">
761         <h2>
762             Welcome : ${pageContext.request.userPrincipal.name} |
763             <a href="/logout"> Logout</a>
764         </h2>
765     </c:if>

```

```
766 </body>
767 </html>
768
769
770 18. views/logout.jsp
771
772 <%@ page language="java" contentType="text/html; charset=UTF-8"
773     pageEncoding="UTF-8"%>
774 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
775 <c:set var="username" value="${pageContext.request.userPrincipal.name}" />
776 <!DOCTYPE html>
777 <html>
778 <head>
779 <meta charset="UTF-8">
780 <title>Logout page</title>
781 </head>
782 <body>
783     <c:url value="/logout" var="logoutUrl" />
784     <!-- csrt for log out-->
785     <form action="${logoutUrl}" method="post" id="logoutForm">
786         <input type="hidden" name="${_csrf.parameterName}"
787             value="${_csrf.token}" />
788     </form>
789     <script>
790         function formSubmit() {
791             document.getElementById("logoutForm").submit();
792         }
793     </script>
794     <form>
795         <span style="color: gray;" ><h5> ${username} 님 반갑습니다.
796         <a href = "javascript:formSubmit()"> 로그아웃 </a> </h5> </span>
797     </form>
798 </body>
799 </html>
```

```
800
801
802 19. Test
803 1)SpringSecurityDemo1 > right-click > Run As > Run on Server
804 2)http://localhost:8080/biz/login 으로 이동되며
805 3)이때 ID와 Password를 잘 못 입력하면 Invalid username and password! 를 보며
806 4)javaexpert/12345678을 입력하면 welcome으로 이동한다.
807 5)welcome에서 Logout link click하면
808 6)logout으로 이동하고 [로그아웃] link를 click하면 다시 첫 page로 이동하지만, You've been logged out
    successfully. message를 만난다.
```

```
809
810
811 -----
812 Task5. 보안관련 taglibs 사용하기
813 1. SpringSecurityDemo project에서
814 2. pom.xml 에서 spring-security-taglibs : 5.2.0.RELEASE 확인
815 3. security/login.jsp에서
816 1)아래 taglib 추가
```



```
817 <%@ taglib uri="http://www.springframework.org/security/tags" prefix="security" %>
```

```
818
```

```
819 2)코드 변경
```

```
820
```

```
821 <c:if test="${not empty pageContext.request.userPrincipal}">
```

```
822 <p>현재 로그인 상태</p>
```

```
823 </c:if>
```

```
824 <c:if test="${empty pageContext.request.userPrincipal}">
```

```
825 <p>현재 로그아웃 상태</p>
```

```
826 </c:if>
```

```
827
```

```
828 3)위의 코드를 아래의 코드로 변경
```

```
829
```

```
830 <security:authorize access="hasRole('ROLE_USER')">
```

```
831 <p>현재 로그인 상태</p>
```

```
832 </security:authorize>
```

```
833 <security:authorize access="hasRole('ROLE_ANONYMOUS')">
```

```
834 <p>현재 로그아웃 상태</p>
```

```
835 </security:authorize>
```

```
836
```

```
837 4)아래의 코드도 변경
```

```
838
```

```
839 User ID : ${pageContext.request.userPrincipal.name}<br />
```

```
840
```

```
841 5)아래의 코드로 변경
```

```
842
```

```
843 User ID : <security:authentication property="name" /><br />
```

```
844
```

```
845 4. Test
```

```
846 1)http://localhost:8080/biz/login.html 을 요청하면 http://localhost:8080/biz/loginForm.html 로 redirect
```

```
847 2)아이디와 패스워드를 미리 설정한 javaexpert/12345678을 넣으면 정상적으로 login.jsp로 이동한다.
```

```
848 3)이 때 화면에는 아래의 message가 나온다.
```

```
849
```

```
850 login.jsp
```

```
851 현재 로그인 상태
```

```
852 User ID : javaexpert
```

```
853 Log Out
```

```
854
```

```
855 4)Logout link를 click하면 다시 첫 page로 이동한다.
```

```
856
```

```
857
```

```
858 -----
```

```
859 Task6. Database-based Login Authentication
```

```
860 1. Spring Legacy Project 생성
```

```
861 1)In Package Explorer > right-click > New > Other > Spring > Spring Legacy Project
```

```
862 2)Project name : SpringSecurityDatabaseDemo
```

```
863 3)Select Spring MVC Project > Next
```

```
864 4)com.example.biz > Finish
```

```
865
```

```
866 2. server.xml에 project 추가
```

```
867 1)Tomcat v9.0 Server at localhost > right-click > Add and Remove
```

```
868 2)SpringSecurityDatabaseDemo > Add > Finish > OK
```

```
869
870 3. Spring security library download and install
871 1)In pom.xml 아래와 같이 수정
872     <java-version>1.8</java-version>
873     <org.springframework-version>5.2.0.RELEASE</org.springframework-version>
874     <org.aspectj-version>1.9.4</org.aspectj-version>
875     <org.slf4j-version>1.7.28</org.slf4j-version>
876
877 2)version 수정
878     <dependency>
879         <groupId>javax.servlet</groupId>
880         <artifactId>javax.servlet-api</artifactId>
881         <version>4.0.1</version>
882         <scope>provided</scope>
883     </dependency>
884     <dependency>
885         <groupId>javax.servlet.jsp</groupId>
886         <artifactId>javax.servlet.jsp-api</artifactId>
887         <version>2.3.3</version>
888         <scope>provided</scope>
889     </dependency>
890     <dependency>
891         <groupId>junit</groupId>
892         <artifactId>junit</artifactId>
893         <version>4.12</version>
894         <scope>test</scope>
895     </dependency>
896
897 3)In mvnrepository, search 'spring security'
898 -Select 'Spring Security Core' 5.2.0.RELEASE
899 -Select 'Spring Security Web' 5.2.0.RELEASE
900 -Select 'Spring Security Config' 5.2.0.RELEASE
901 -Select 'Spring Security Taglibs' 5.2.0.RELEASE
902
903     <dependency>
904         <groupId>org.springframework.security</groupId>
905         <artifactId>spring-security-core</artifactId>
906         <version>5.2.0.RELEASE</version>
907     </dependency>
908     <dependency>
909         <groupId>org.springframework.security</groupId>
910         <artifactId>spring-security-web</artifactId>
911         <version>5.2.0.RELEASE</version>
912     </dependency>
913     <dependency>
914         <groupId>org.springframework.security</groupId>
915         <artifactId>spring-security-config</artifactId>
916         <version>5.2.0.RELEASE</version>
917     </dependency>
918     <dependency>
919         <groupId>org.springframework.security</groupId>
920         <artifactId>spring-security-taglibs</artifactId>
```

```
921     <version>5.2.0.RELEASE</version>
922   </dependency>
923
924 4)pom.xml > right-click > Run As > Maven clean and Maven install
925 5)project > right-click > Properties > Project Facets > Java > Select 1.8
926 6)Runtimes Tab > Check Apache Tomcat v9.0
927 7)Click Apply and Close
```

```
928
929
930 4. HomeController.java
```

```
931
932 package com.example.biz;
933
934 import org.springframework.stereotype.Controller;
935 import org.springframework.ui.Model;
936 import org.springframework.web.bind.annotation.RequestMapping;
937 import org.springframework.web.bind.annotation.RequestMethod;
938
939 @Controller
940 public class HomeController {
941     @RequestMapping(value = "/", method = RequestMethod.GET)
942     public String home(Model model) {
943         model.addAttribute("greeting", "Hello Spring Security");
944         return "home";
945     }
946 }
```

```
947
948
949 5. home.jsp
```

```
950
951 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
952 <%@ page session="false" %>
953 <html>
954 <head>
955     <title>Home</title>
956 </head>
957 <body>
958 <h1>${greeting} </h1>
959
960 </body>
961 </html>
```

```
962
963
964 6. project > right-click > Run As > Run on Server
```

```
965
966 7. Database table & sequence creation
```

```
967
968 CREATE SEQUENCE seq_user_role_id
969 MAXVALUE 9999;
970 /
971
972 CREATE TABLE Users (
```

```
973     username VARCHAR2(20),
974     password VARCHAR2(20) NOT NULL,
975     enabled NUMBER(1) DEFAULT 1 ,
976     CONSTRAINT users_username_pk PRIMARY KEY (username)
977 );
978
979 CREATE TABLE User_roles (
980     user_role_id  NUMBER(4),
981     username VARCHAR2(45),
982     role VARCHAR2(45) NOT NULL,
983     CONSTRAINT user_roles_id PRIMARY KEY (user_role_id),
984     CONSTRAINT user_roles_uk UNIQUE (role,username),
985     CONSTRAINT user_roles_fk FOREIGN KEY (username) REFERENCES users (username)
986 );
987
988
```

989 8. Inserts some records for testing.

```
990
991 INSERT INTO Users(username,password,enabled) VALUES ('javaexpert','12345678', 1);
992 INSERT INTO users(username,password,enabled) VALUES ('alex','12345678', 1);
993
994 INSERT INTO user_roles (user_role_id, username, role)
995 VALUES (seq_user_role_id.NEXTVAL, 'javaexpert', 'ROLE_USER');
996
997 INSERT INTO user_roles (user_role_id, username, role)
998 VALUES (seq_user_role_id.NEXTVAL, 'javaexpert', 'ROLE_ADMIN');
999
1000 COMMIT;
1001
1002
```

1003 9. pom.xml에 oracle driver와 MyBatis 관련 dependency 추가

```
1004
1005 <dependency>
1006     <groupId>com.oracle</groupId>
1007     <artifactId>ojdbc6</artifactId>
1008     <version>11.2</version>
1009 </dependency>
1010 <!-- https://mvnrepository.com/artifact/org.mybatis/mybatis -->
1011 <dependency>
1012     <groupId>org.mybatis</groupId>
1013     <artifactId>mybatis</artifactId>
1014     <version>3.5.3</version>
1015 </dependency>
1016 <!-- https://mvnrepository.com/artifact/org.mybatis/mybatis-spring -->
1017 <dependency>
1018     <groupId>org.mybatis</groupId>
1019     <artifactId>mybatis-spring</artifactId>
1020     <version>2.0.3</version>
1021 </dependency>
1022
1023
```

1024 10. Database 연결을 위한 환경설정

```
1025 1)Build path에 config folder 추가
1026 -project right-click > Build Path > Configure Build Path > Select [Source] tab
1027 -Click [Add Folder] > Select 현재 project > Click [Create New Folder...]
1028 -Folder name : config > Finish > OK > Apply and Close
1029 -Java Resources > config folder 확인
1030
1031 2)config/dbinfo.properties file 생성
1032
1033 db.driver=oracle.jdbc.driver.OracleDriver
1034 db.url=jdbc:oracle:thin:@localhost:1521:XE
1035 db.username=hr
1036 db.password=hr
1037
1038 3)pom.xml에 spring-jdbc 추가
1039
1040 <!-- https://mvnrepository.com/artifact/org.springframework/spring-jdbc -->
1041 <dependency>
1042   <groupId>org.springframework</groupId>
1043   <artifactId>spring-jdbc</artifactId>
1044   <version>5.2.0.RELEASE</version>
1045 </dependency>
1046
1047 11. web.xml 코드 추가
1048
1049 <filter>
1050   <filter-name>springSecurityFilterChain</filter-name>
1051   <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>
1052 </filter>
1053
1054 <filter-mapping>
1055   <filter-name>springSecurityFilterChain</filter-name>
1056   <url-pattern>/*</url-pattern>
1057 </filter-mapping>
1058
1059 12. web.xml 수정
1060
1061 <context-param>
1062   <param-name>contextConfigLocation</param-name>
1063   <param-value>/WEB-INF/spring/spring-security.xml</param-value>
1064 </context-param>
1065
1066 13. src/main/webapp/WEB-INF/spring/spring-security.xml 생성
1067 1)src/main/webapp/WEB-INF/spring > right-click > New > Spring Bean Configuration File
1068 2)File name : spring-security.xml > Next
1069 3)Check beans, security, context > Finish
1070 4)security check할 때 반드시 https://www.springframework.org/schema/security/spring-security.xsd 즉
1071 xsd의 version이 없는 것을 선택해야 함.
1072
1073 <?xml version="1.0" encoding="UTF-8"?>
1074 <beans xmlns="http://www.springframework.org/schema/beans"
1075   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1076   xmlns:security="http://www.springframework.org/schema/security"
```

```

1076 xmlns:context="http://www.springframework.org/schema/context"
1077 xsi:schemaLocation="http://www.springframework.org/schema/security
http://www.springframework.org/schema/security/spring-security.xsd
1078 http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
1079
1080 <context:property-placeholder location="classpath:dbinfo.properties"/>
1081 <bean id="dataSource"
1082     class="org.springframework.jdbc.datasource.DriverManagerDataSource">
1083     <property name="driverClassName"
1084         value="${db.driver}" />
1085     <property name="url"
1086         value="${db.url}" />
1087     <property name="username" value="${db.username}" />
1088     <property name="password" value="${db.password}" />
1089 </bean>
1090 <bean id="customPasswordEncoder"
1091     class="com.example.security.MyPasswordEncoder" />
1092 <!-- enable use-expressions -->
1093 <security:http>
1094
1095     <security:intercept-url pattern="/admin**" access="hasRole('ROLE_ADMIN')" />
1096
1097     <!-- access denied page -->
1098     <security:access-denied-handler error-page="/403" />
1099
1100     <security:form-login
1101         login-page="/login"
1102         default-target-url="/welcome"
1103         authentication-failure-url="/login?error"
1104         login-processing-url="/j_spring_security_check"
1105         username-parameter="username"
1106         password-parameter="password" />
1107     <security:logout invalidate-session="true"
1108         logout-url="/j_spring_security_logout" logout-success-url="/login?logout" />
1109     <!-- enable csrf protection -->
1110     <security:csrf disabled="true"/>
1111 </security:http>
1112
1113 <!-- Select users and user_roles from database -->
1114 <security:authentication-manager>
1115     <security:authentication-provider>
1116         <security:jdbc-user-service data-source-ref="dataSource"
1117             users-by-username-query=
1118                 "select username,password, enabled from users where username=?"
1119             authorities-by-username-query=
1120                 "select username, role from user_roles where username =? " />
1121         <security:password-encoder ref="customPasswordEncoder" />
1122     </security:authentication-provider>
1123 </security:authentication-manager>
1124 </beans>
1125

```

```
1126
1127 14. views/hello.jsp
1128
1129 <%@ page language="java" contentType="text/html; charset=UTF-8"
1130     pageEncoding="UTF-8"%>
1131 <%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
1132 <%@taglib prefix="sec"
1133     uri="http://www.springframework.org/security/tags"%>
1134 <html>
1135 <body>
1136     <h1>Title : ${title}</h1>
1137     <h1>Message : ${message}</h1>
1138
1139     <sec:authorize access="hasRole('ROLE_USER')">
1140         <!-- For login user -->
1141         <c:url value="/j_spring_security_logout" var="logoutUrl" />
1142         <form action="${logoutUrl}" method="post" id="logoutForm">
1143             <input type="hidden" name="${_csrf.parameterName}"
1144                 value="${_csrf.token}" />
1145         </form>
1146         <script>
1147             function formSubmit() {
1148                 document.getElementById("logoutForm").submit();
1149             }
1150         </script>
1151
1152         <c:if test="${pageContext.request.userPrincipal.name != null}">
1153             <h2>
1154                 User : <sec:authentication property="name" /> | <a
1155                     href="javascript:formSubmit()"> Logout</a>
1156             </h2>
1157         </c:if>
1158
1159     </sec:authorize>
1160 </body>
1161 </html>
1162
1163
1164
1165 15. views/login.jsp
1166
1167 <%@ page language="java" contentType="text/html; charset=UTF-8"
1168     pageEncoding="UTF-8"%>
1169 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
1170 <html>
1171 <head>
1172 <title>Login Page</title>
1173 <style>
1174     .error {
1175         padding: 15px;
1176         margin-bottom: 20px;
1177         border: 1px solid transparent;
```

```
1178     border-radius: 4px;
1179     color: #a94442;
1180     background-color: #f2dede;
1181     border-color: #ebccd1;
1182 }
1183
1184 .msg {
1185     padding: 15px;
1186     margin-bottom: 20px;
1187     border: 1px solid transparent;
1188     border-radius: 4px;
1189     color: #31708f;
1190     background-color: #d9edf7;
1191     border-color: #bce8f1;
1192 }
1193
1194 #login-box {
1195     width: 300px;
1196     padding: 20px;
1197     margin: 100px auto;
1198     background: #fff;
1199     -webkit-border-radius: 2px;
1200     -moz-border-radius: 2px;
1201     border: 1px solid #000;
1202 }
1203 </style>
1204 </head>
1205 <body onload='document.loginForm.username.focus();'>
1206
1207     <h1>Spring Security Login Form (Database Authentication)</h1>
1208
1209     <div id="login-box">
1210
1211         <h2>Login with Username and Password</h2>
1212
1213         <c:if test="${not empty error}">
1214             <div class="error">${error}</div>
1215         </c:if>
1216         <c:if test="${not empty msg}">
1217             <div class="msg">${msg}</div>
1218         </c:if>
1219
1220         <form name='loginForm'
1221             action="<c:url value='/j_spring_security_check' />" method='POST'>
1222
1223             <table>
1224                 <tr>
1225                     <td>User:</td>
1226                     <td><input type='text' name='username'> </td>
1227                 </tr>
1228                 <tr>
1229                     <td>Password:</td>
```



```

1230         <td><input type='password' name='password' /> </td>
1231     </tr>
1232     <tr>
1233         <td colspan='2'><input name="submit" type="submit"
1234             value="submit" /> </td>
1235     </tr>
1236 </table>
1237
1238     <input type="hidden" name="{_csrf.parameterName}"
1239         value="{_csrf.token}" />
1240
1241 </form>
1242 </div>
1243 </body>
1244 </html>
1245
1246
1247 16. views/admin.jsp
1248
1249 <%@ page language="java" contentType="text/html; charset=UTF-8"
1250     pageEncoding="UTF-8"%>
1251 <%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
1252 <html>
1253 <body>
1254     <h1>Title : ${title}</h1>
1255     <h1>Message : ${message}</h1>
1256
1257     <c:url value="/j_spring_security_logout" var="logoutUrl" />
1258     <form action="{logoutUrl}" method="post" id="logoutForm">
1259         <input type="hidden" name="{_csrf.parameterName}"
1260             value="{_csrf.token}" />
1261     </form>
1262     <script>
1263         function formSubmit() {
1264             document.getElementById("logoutForm").submit();
1265         }
1266     </script>
1267
1268     <c:if test="{pageContext.request.userPrincipal.name != null}">
1269         <h2>
1270             Welcome : ${pageContext.request.userPrincipal.name} | <a
1271                 href="javascript:formSubmit()"> Logout </a>
1272         </h2>
1273     </c:if>
1274
1275 </body>
1276 </html>
1277
1278
1279 17. views/403.jsp
1280
1281 <%@ page language="java" contentType="text/html; charset=UTF-8"

```

```
1282     pageEncoding="UTF-8"%>
1283     <%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
1284     <html>
1285     <body>
1286         <h1>HTTP Status 403 - Access is denied</h1>
1287
1288         <c:choose>
1289             <c:when test="${empty username}">
1290                 <h2>You do not have permission to access this page!</h2>
1291             </c:when>
1292             <c:otherwise>
1293                 <h2>Username : ${username} <br/>
1294                     You do not have permission to access this page!</h2>
1295             </c:otherwise>
1296         </c:choose>
1297
1298     </body>
1299     </html>
1300
1301
1302 18. HomeController.java
1303
1304     @RequestMapping(value = { "/", "/welcome**" }, method = RequestMethod.GET)
1305     public ModelAndView defaultPage() {
1306
1307         ModelAndView model = new ModelAndView();
1308         model.addObject("title", "Spring Security Login Form - Database Authentication");
1309         model.addObject("message", "This is default page!");
1310         model.setViewName("hello");
1311         return model;
1312
1313     }
1314
1315     @RequestMapping(value = "/admin**", method = RequestMethod.GET)
1316     public ModelAndView adminPage() {
1317
1318         ModelAndView model = new ModelAndView();
1319         model.addObject("title", "Spring Security Login Form - Database Authentication");
1320         model.addObject("message", "This page is for ROLE_ADMIN only!");
1321         model.setViewName("admin");
1322         return model;
1323
1324     }
1325
1326     @RequestMapping(value = "/login", method = RequestMethod.GET)
1327     public ModelAndView login(@RequestParam(value = "error", required = false) String error,
1328         @RequestParam(value = "logout", required = false) String logout) {
1329
1330         ModelAndView model = new ModelAndView();
1331         if (error != null) {
1332             model.addObject("error", "Invalid username and password!");
1333         }
```

```
1334
1335     if (logout != null) {
1336         model.addObject("msg", "You've been logged out successfully.");
1337     }
1338     model.setViewName("login");
1339
1340     return model;
1341 }
1342
1343 //for 403 access denied page
1344 @RequestMapping(value = "/403", method = RequestMethod.GET)
1345 public ModelAndView accesssDenied() {
1346
1347     ModelAndView model = new ModelAndView();
1348
1349     //check if user is login
1350     Authentication auth = SecurityContextHolder.getContext().getAuthentication();
1351     if (!(auth instanceof AnonymousAuthenticationToken)) {
1352         UserDetails userDetail = (UserDetails) auth.getPrincipal();
1353         model.addObject("username", userDetail.getUsername());
1354     }
1355
1356     model.setViewName("403");
1357     return model;
1358 }
1359
1360 @RequestMapping(value = "/logout", method = RequestMethod.GET)
1361 public String logout(Model model) {
1362     return "logout";
1363 }
1364
1365
1366
1367 19. com.example.security.MyPasswordEncoder Class 생성
1368     1)com.example.security package 생성
1369     2)com.example.security.MyPasswordEncoder Class
1370
1371     package com.example.security;
1372
1373     import org.springframework.security.crypto.password.PasswordEncoder;
1374
1375     public class MyPasswordEncoder implements PasswordEncoder {
1376
1377         @Override
1378         public String encode(CharSequence rawPassword) {
1379             return rawPassword.toString();
1380         }
1381
1382         @Override
1383         public boolean matches(CharSequence rawPassword, String encodedPassword) {
1384             return rawPassword.toString().equals(encodedPassword);
1385         }
1386     }
```

```
1386     }
1387
1388 20. Test
1389 1)javaexpert/12345678은 admin과 일반 login모두 성공
1390 2)하지만 alex/12345678은 login 실패, 왜냐하면 role이 없기 때문에
1391 3)login하려면 아래 query처럼 user_roles에 등록해야 ROLE_USER 권한을 부여 받음.
1392     INSERT INTO user_roles (user_role_id, username, role)
1393     VALUES (seq_user_role_id.NEXTVAL, 'alex', 'ROLE_USER');
1394 4)그렇다면 alex가 admin page에 접근할 수 있을까?
1395 5)현재는 403 error page로 이동한다. 왜냐하면 그는 ROLE_USER 권한만 있기 때문이다.
1396     HTTP Status 403 - Access is denied
1397     Username : alex
1398     You do not have permission to access this page!
```